Subject: Miscellaneous items
From: Van Snyder
References: 00-192r1 00-261

The current revision of this paper is shorter than the previous revision, because much of it has been put into 00-261, a paper arising from the WG5 meeting at Oulu.

Here are several things that may or may not need attention. I don't even offer edits (well, sometimes I offer crappy ones). If they need attention, we can develop edits at the meeting, if we have time, or insert unresolved issue notes.

Page and line numbers refer to 00-007r2.

Do we want to allow to specify default values for type parameters? 41:32

[Editor: The Frame-ism (724) here probably ought to be (7.5.1.5).] 45:38

It's not obvious why the BIND attribute necessarily implies the SAVE attribute. It wouldn't hurt to have an explanation. 79:32-33

Add "of type integer" at the end of the constraint. (This one apparently has the wrong page and line number. To where does it apply? Look at 00-192r1?) 84:17

The description of the result characteristics in the case of *proc-binding* does not include the case when an abstract interface is explicitly specified in the PROCEDURE statement. 115:28-43

I suggest: 132:22+

Constraint: In the case of intrinsic assignment, the types of *variable* and *expr* shall conform according to the rules in table 7.9.

Put table 7.9 here, but with "type parameters" for a derived type replaced by "kind type parameters."

Constraint: In the case of intrinsic assignment, the *variable* and *expr* shall have the same rank or the *expr* shall be a scalar.

Replace by "The *variable* and *expr* shall conform in shape. If *variable* is of derived type, corresponding type parameters of *variable* and *expr* shall have the same values." 133:19-21

If this and the change suggested for [132:20+] above are not accepted, at least move table 7.9 to [133:21+].

There was a discussion in /data subgroup concerning whether the associate name ought to have the POINTER or ALLOCATABLE attribute if and only if the selector does. I thought the outcome was that it ought to, but those attributes are absent here. If the outcome was that they ought not to, it wouldn't hurt to have a note here explaining why not – it's not obvious. On the other hand, if it's possible for the associate name to have the POINTER or ALLOCATABLE attributes, it would be useful: It would make it easier to allocate, deallocate and pointer-assign components that have complicated antecedents. If this needs to be done, should it be done in §14? 156:34

The term **effective item** is set in bold-face type, but is not in the glossary. 183:42

Do we need to add "accessible" after "entity," or was the intent to restrict IMPORT to work only for entities declared within the scoping unit containing the interface body? 247:3

This paragraph also needs to mention dummy procedures and procedure pointers. 247:4-6

It would be convenient to be able to use any accessible explicit interface to declare the interface 252:12+

for a procedure pointer. Could we add "**or** *procedure-name*" as an additional right-hand side for R1211? We would also need to replace "consists ... pointers" by "and specifies an explicit specific interface, the declared procedures or procedure pointers have the same explicit specific interface" at [253:7].

It is not generally necessary for actual arguments associated with INTENT(OUT) dummy arguments to be defined when the procedure reference takes place. In order to have a dynamic type, however, the *data-ref* has to be defined. Do we need a note to the effect that *data-ref* shall be defined even if the passed-object dummy argument has INTENT(OUT)? Or how about a constraint that the passed-object dummy argument shall not have INTENT(OUT)? 254:26+

"If ... polymorphic" could be constraints, because allocatable and pointer dummy arguments require explicit interface. (This one apparently has the wrong page and line number. To where does it refer? Look at 00-192r1?) 257:24-28

The phrase "an elemental intrinsic actual procedure may be associated with a dummy argument that is not elemental" leads one to believe that dummy arguments can be elemental. The part "that is not elemental" should be removed. Three possibilities for what to do next are (1) nothing, (2) add a parenthetic remark "(which cannot be elemental)", or (3) put in a note $12.27\frac{1}{2}$ to the effect that dummy arguments cannot be elemental. 260:3-5

We could get rid of "other than as the argument of the PRESENT intrinsic function" by making the argument of the PRESENT intrinsic function optional. 261:6-17

Subclause 14.1.2.3 has nothing to do with generic procedure references. The title ought to be "Unambiguous generic procedure definitions." 343:10

The sentence "If a generic..." conflicts with, or at least belongs in [344:25-26]. 343:42-44

These paragraphs do not explicitly apply to defined operations, defined assignment, or user-defined derived-type input/output. They apply indirectly to defined operations by way of the phrase "it is generic in exact analogy to generic procedure names" at [250:15], but there is no parallel statement for defined assignment or user-defined derived-type input/output. 345:4-14

Either "procedure" should be "interface" at [345:8], or vice-versa at [345:14]. 345:8,14

Consider the following: 389:42+

```
TYPE(C_PTR) :: Y
REAL(C_FLOAT), ALLOCATABLE, TARGET :: Z(:)
ALLOCATE ( Z(10) )
Y = C_LOC(Z)
DEALLOCATE ( Z )
! Is Y still defined here?  Do we need to say something in 14.7.6?
```

---

generic interface needs to be in the glossary. 402:10+