02-242r2

To: J3
From: Malcolm Cohen
Subject: Edits for sections 4 to 6
Date: 14 August 2002

[31:6] Change 'derived by composition of other types' to
'defined by a derived-type definition (4.5.1).'
{Previous was incorrect and almost incorrigible.}

[31:6-8] Delete "Object ... its components."
{This is said better, maybe even correctly, in 4.5.}

[41:7] Embolden "components".
{This is where the glossary entry points to.}

[33:Note 4.4+1] Change 'including' to 'and' and delete both commas.
{No, correct as it stands.}

[41:6] After "intrinsic types" add "and other derived types".
{Make this sentence less misleading.}

[41:16+2]
{No. Current text is ok.}

[41:19+] Add introductory witter.
"A derived type may have procedures bound to it. A type-bound procedure is accessible when an object of the type is accessible."

[42:7] Subsumed by 02-257r1.

[43:10] After 'shall' add 'either be an initialization expression or be a
specification expression that does'.
{The wording in Corrigendum 1 to F95 allows for an initialization expression here. The present wording would lead to an inconsistency with f95. See also C550 on p. 79.}

[43:14] Change 'colon or' to 'colon, be an initialization expression, or'.
{See previous edit.}

[43:31 to 44:3] Subsumed by 219r1.

[44:4] After "the" insert "same".
{Clarify ambiguity.}

44:5. Delete.
{219r1 already put this text into C463½.}

[44:25&27] (TECHNICAL CHANGE).
{No. This is too complicated to deal with in this paper. Another paper may revisit this issue.}

[45:11-19] Subsumed by 219r1.

[45:19] Insert "same" after "the".
{Remove ambiguity.}

[45:31] Delete.
{219r1 already put this text into C463½.}

[46:8]
[46:9-10]
[46:14]
[46:15+9]
No. All subsumed by 02-247.

[47:19-22] No. The suggested text is a circular definition.

[47:20] Before "." insert
", and the subcomponent is not a subobject of an object that is default-initialized or explicitly initialized."

[47:20-22] Delete "If ...)."
{Delete unnecessary explanation.}

[49:2] Change 'component' to 'data component' and '<component-def->' to
'<data-component-def->'.
{The text is correct but this makes it clearer.}

[50:4-5] (Technical Change as above). Deferred to another paper.

[51:2] (Technical Change as above). Deferred to another paper.

[51:3] Change "a procedure" to "an overriding procedure".
{Clearer.}

[51:6] No.

[51:23+6-8] No. Suggested text does not seem a significant improvement.

[52:15] No. Current text is needed to cater for extended types.

[52:18] No. Current text is correct, suggested replacement is wrong.

[52:19] No. It is not accessible in any other way.
(The implementation procedure might be accessible, but the type-bound procedure is not.)

[56:6] Change 'nongeneric' to 'specific'.
{This is what is used in the bnf.}

[56:20] Change 'binding' to 'specific binding'.

[57:1-3] No. Existing text is ok.

[57:14] Insert "nonparent" before "components".
[57:15] Add to end of paragraph
'Parent components are excluded from the component order of an extensible type."

[58:7] Delete 'and the type parameters'.

[This makes it correct if rather bland. Otherwise, we need to say something like 'The set of values of a particular derived type with a particular set of type parameter values is ...']

[58:37-39] Change to
"C471 At most one <component-spec> shall be provided for a component.
C471a If a <component-spec> is provided for a component, no <component-spec> shall be provided for any component with which it is inheritance associated.
C472 A <component-spec> shall be provided for a component unless it has default initialization or is inheritance associated with another component for which a <component-spec> is provided or that has default initialization."
{We need to allow for explicit and default initialization of parent components.)

[59:20-21] Delete.
{The restriction at 59:20-21 is now covered by the constraints.}

[60:4] Subsumed by 225.

[61:11-] Insert new sentence
"Only finalizable entities are <<finalized>>."
[61:11] Delete "finalizable" and disembolden "<<finalized>>".
{Make it clear the the entire rest of 4.5.10 and its subclauses apply only to finalizable entities.}

[61:16] No. Existing text is ok (clarifying).

[61:24] No. This would be too confusing.

[61:33] Delete 'whose result is finalizable'.
[61:35] Delete 'for a finalizable type'.
[61:38] Delete 'whose result is finalizable'.
{These qualifications are not necessary.}

[69:3] Embolden and index "attributes".
{This is the definition.}

[70:18] Before "in an <entity-decl>" insert "in a <char-length>".
{Give the path to the <type-param-value>.}

[72:26-28] Subsumed by 234r1.
[73:0] Ditto.

[74:33-34] These features are ***NOT***obsolescent!

[76:20]. After the first sentence add 'For a polymorphic entity, type compatibility is based on its declared type.'
[I completely misinterpreted this at first, thinking it was based on the dynamic type.]

[81:28] Subsumed by 238r1
[81:37] Subsumed by 238r1
[81:39] Subsumed by 238r1

[85:6] Change "object" to "entity" .
[85:8] Change "the object" to "it" and "The object" to "It".
[85:9-10] Delete "Such ... <<unsaved>> object."

[85:11] Change "object" to "entity" .
[85:13] Change "the object" to "it".
{Make this cover both objects and procedure pointers.}

[85:14-15] Delete.
{Now covered by first paragraph.}

[85:15+] Insert new paragraph
"A <<saved>> entity is an entity that has the SAVE attribute. An <<unsaved>> entity is an entity that does not have the SAVE attribute."
{"saved" and "unsaved" are not limited to objects in a subprogram.}

[106:7] Unnecessary. Paper 268 clarifies this.

[111:14] Change "values of the kind type parameters" to "kind type parameter values".
[111:15] Replace 'those' by 'the corresponding type parameter values'.
{The sets of kind type values can differ.}

[111:40] Add 'If it is an expression, subsequent redefinition or undefinition of any entity in the expression does not affect the type parameter value.'

[114:0+2] Change 'not saved' to 'unsaved'.
{Use our nicely defined term.}

[116:4] Add 'either' before 'allocatable'.
{At first, I read the sentence as if 'either' appeared before 'a function'.}

[116:8] Add 'either' before 'allocatable'.

[116:12-13] Delete "or a subobject of an actual argument".
[116:13] Change "." to
"; an allocated allocatable object that is a subobject of an actual argument associated with an INTENT(OUT) dummy argument is deallocated."
{Previous text was unclear and not quite right.}

[116:20] Change '<dealloc-opt>' to '<dealloc-opt-list>'.
{Clearer.}