

Subject: Types as type parameters
 From: Van Snyder
 Reference: 03-258r1, section 2.2.4

1 Number

2 TBD

3 Title

4 Types as type parameters.

5 Submitted By

6 J3

7 Status

8 For consideration.

9 Basic Functionality

10 Expand the type parameter system to include types.

11 Rationale

12 The arguments for doing this are similar to the arguments for a parameterized module facility: Many
 13 algorithms can be applied to more than one type. This proposal is an anemic attempt to provide support
 14 for that requirement. It is a backup to the proposal to provide a parameterized module facility (or one
 15 of equivalent functionality but different foundation). Although this would be desirable in the event no
 16 parameterized module facility is provided, it has the same fundamental problems of integration with
 17 type-bound procedures that arise from kind type parameters. If a parameterized module facility is
 18 provided, this proposal should not be pursued.

19 Estimated Impact

20 Small to moderate. Most of the changes will be in Section 4, but changes will also be needed in Section
 21 12, and maybe Section 16.

22 Detailed Specification

23 Expand the type parameter system to include types. This will require a declaration that a type parameter
 24 is a type, and changes in the discussion of argument association to account for a type parameter that is
 25 a type.

26 Type parameters that are types can themselves have parameters, including parameters that are types,
 27 etc.

28 8.1 Example

```
29 TYPE :: T ( U, V, K, L )
30     TYPE :: U, V ! Doesn't introduce a new type because type definitions don't nest
31     INTEGER, KIND :: K, L
32     TYPE(U(V(K),L)) :: COMPONENT
33 END TYPE T
```

- 1 Type T has a parameter U, which is a type that has a parameter that is a type, specified by V, which
- 2 has a kind parameter specified by K. U also has a kind parameter specified by L.

3 **9 History**