

From: Ed Greengrass

To: WG11

Subject: Comments on Brian Meek's Comment on Turba's Comment on Undefined!  
(WG11/N265)

Date: Wed, 17 Jul 91 14:57:03 -0400

I have already had my say (in 3 parts) on Turba's comments. Having just read Brian's comments, I want to add one more remark on the contentious issue of Undefined.

Brian says that "There are essentially 3 approaches to the vexed NULL/UNDEFINED problem." He gives them as follows:

(1) One is to say that NULL and UNDEFINED are states of an entity which would normally have a value of a given (possibly Choice) datatype, i.e. aren't values at all.

(2) Another is to say that every datatype has NULL and UNDEFINED in its value set.

(3) The third is that used in the CLID CD, of just having Null and Undefined datatypes.

Brian says that he "can actually live with any of the three, if they are carried through consistently." However, he says that his "PERSONAL preference is for the second possibility rather than what we have now [i.e., alternative 3]. My own view, expressed in my comments on Turba, is a PERSONAL preference for alternative 3 combined with the acknowledgment that I could live with alternative 2 if I had to.

But the point I want to make here (as I have made it in several papers including my comments on Turba) is that there is an essential error in alternative 1. There is an essential difference between a "status" of a variable and an Undefined datatype. The former has meaning only within a program while the latter has "real world" semantics. (I won't repeat the details of this argument yet again.) I have tried to express this difference by arguing for a change in the name of the datatype to Unknown. The value space of this "state" datatype is indeed open-ended as Barkmeyer and Yellin have pointed out. (I suggested such a value space but made no claim that my list of Unknown states was exhaustive.) However, this is not an argument for excluding Unknown from CLIDT. As a matter of fact, the confusion on this point is itself one of the best reasons for including an Unknown type in CLIDT. Programmers (like me, for example) are very familiar with undefined and uninitialized variables. They are usually NOT familiar with the semantics of an Unknown type (Null and Unknown seem to be more familiar to database research types). Hence, there is a very strong tendency to confuse the Undefined status with the Unknown datatype. Including an Unknown datatype in CLIDT would go a long way toward alleviating this confusion - which is one of the legitimate functions of a CLIDT standard!