

From: turba@s5000.RSVL.UNISYS.COM

To: WG11

Subject: Comments on CLIDT

Date: 3 Jul 91 12:28:18 CDT (Wed)

Comments on Common Language-Independent Datatypes
Working Draft #5
X3T2/91-109
JTC1/SC22/WG11 N233

T. N. Turba
3 July 1991

1. Introduction

At the June 1991 meeting of X3J9 the committee did a quick review of Working Draft #5. From this quick review it was determined that a thorough review of the document would be impossible at the meeting because of the size of the document and the number of issues that were being discovered.

It was agreed that a committee response could not be prepared at the meeting, and that members of X3J9 should send in their comments individually after the meeting and to inform other possible commentors on the committee of their findings.

The following are the things that I noted in a cursory review of the document. It is by no means a complete review of the document because other commitments did not permit an in-depth review.

The comments are somewhat ordered by the perceived severity of the underlying problem.

2. General Comment

None of the datatype concepts of object-oriented programming, e.g., subclassing and inheritance, are addressed in this document. This is extremely unfortunate, especially when common terms such as "subtype" are used in this area but have a definition that is more inclusive.

3. Undefined Datatype 7.1.14

Undefined is not a datatype!!!

The notes under Outstanding Issues suggests that there is an underlying confusion here. Maybe the comments here can add light to the issue.

By your own definition of a datatype in 6.1, "a datatype is a collection of distinct values, a collection of properties on those values and a collection of characterizing operations on those values", undefined is not a datatype. It does not have any values; it does not have any properties on those values (because there are no values); and it does not have any operations on those values.

Either the definition of a datatype must be warped into something I cannot even imagine so that it can accommodate undefined, or more correctly, undefined should not be classified as a datatype.

The correct classification for undefined is that it is a state of a variable or field that can contain a value of a datatype. It is the state when the variable or field has not been attributed a value of its datatype.

4. *Null Datatype 7.1.13*

Null should not be a datatype!

From the comments under Outstanding Issues it is clear that there is considerable confusion here. This is probably, in part, due to the fact that there are several distinct semantic concepts that are attempting to be rolled into one.

A null record or null variant is not a null datatype. It is simply a record or variant that has no content. Its datatype is that of a record, array, etc.

Representation of the absence of a value in a position where something may or not be found in the search of a database does not require the invention of a null datatype. What is normally returned in such positions is a set of objects, where that set may be the null set of those objects.

5. *Procedure Datatype 7.1.16*

There is considerably more syntax than semantics and I can only guess at what some things might be. For example, what is an exception-argument- list in exception? Considerable more description is needed.

The rules for subtypes disagree with some current languages such as C++.

I would hope that the "call" operation would be allowed for values of type procedure.

6. *Complex Datatype 7.1.12*

This datatype assumes an implementation strategy based on Cartesian coordinates, which is not the only means by which complex numbers can be implemented. This is unnecessary and conflicts with the definition of the complex datatype in languages such as Extended Pascal.

7. *Annex A*

Requiring that Choice be a required datatype generator seems a bit strong, especially for languages like FORTRAN that only indirectly support the concept. What is the rationale for requiring it?

Also, why are Lists a required datatype generator? They are not nearly as essential as records or arrays, and many languages do not directly support them.