Subject: Comments on many comments
Date: Thu, 14 Mar 91 16:52:56 -0500
From: craig@crl.dec.com (Craig Schaffert)

Dear Ed, Dan, Chi, and Paul,

I have read the recent exchanges with interest and I'd like to throw in my $2 (inflation, you know).

-- SECTION 1 --

Ed's comments on "RPC, CLIPC and subsetting" are overall right on the mark. A crude summary of Ed's in-favor/out-of-favor RPC list would seem to be

Exclude state, ordinal, rational, null, undefined, table, bitstring, switch, cardinal, integer modulo, currency, and stack.

Continue to include date-time, scaled, complex, set, list, and bag.

DECIDE SOON on array versus list semantics, modify array to multidimensional.

Add OSI-Object-Identifier.

-- SECTION 2 --

I would like to reiterate my strong opinion that there should be two classes of types in the CLID:

Core types: enough to be representationally complete, plus types that are very common. This set probably does not grow over time.

Extended types: any type that seems to have significant interest, and that is semantically distinct from other types in some non trivial way. This set probably expands over time.

Core types are given a semantics through a model or a list of generating or "characterizing" operations. Extended types are the same but each one of them also is given

A standard "external representation" in terms of (a combination of) core types.

A standard information-preserving translation to and from that external representation.

The external representation has no impact on computational representation, but it does provide a standard way of transmitting any value of an extended type.

Examples:

Date-time can be viewed as an extended type whose external representation is String of ISO-LATIN-1 characters, and whose to/from maps are specified in ISO 8601.

Scaled can be viewed as an extended type whose external representation is RECORD (multiplier, base,

exponent: integer), and whose "from" map is
from (<m,b,e>) = m * (b ^ e) and whose "to" map is any inverse of the "from" map producing records with base >= 2.

Even set and bag can be viewed as extended types if we want to.

The concept of extended types (coupled with the local_rep/encode/decode attributes in RPC) would (1) permit the RPC committee to just require the core types (if they are feeling conservative today), or (2) permit an RPC implementer to build a (simpler) two-level implementation consisting of a "core types only" RPC below, and a set of IDN definitions for the extended types on top.

-- SECTION 3 --

Now to specific points in Ed's memo.

(11) I have never found the Mathematical vs Computational philosophy arguments particularly helpful. Ed: what do you mean by a "scientific number"? And what's wrong with REAL(rel-err) as proposed with the IDN?

(13) We do not need any form of Undefined. If the RPC standard chooses not to specify the value in an OUT parameter, then that's a lack of information, but it doesn't need a CLI datatype to capture this fact.

(14) Both Private (or Opaque) and Octet-sequence preserve bit-level representation when returned to the originator. However, the bits of an Octet-sequence are visible (and constant) to intermediate recipients as well. We can use either, it's not true that they are identical.

(16) Ed, what is your current thinking on other CLID "subtype forms?" The IDN tried to support them

(*) The IDN provides potential notation for any kind of "constant" even tables. (Not pointers or unnamed procedures though.) Ed: are constants still needed for everything?

(19) The IDN does not restrict pointers at all! There are annotation for declaring that particular instances of pointers satisfy further properties.

(25) The IDN does not syntactically distinguish parameterized types and type generators. I don't think that this is bad.

-- SECTION 4 --

I would like to second Al Simons' point that it is really nice to have a single piece of text (a file?) that defines the semantics of an interface in a platform- and language-independent manner, and to have a separate piece of text that "customizes" that interface with language specific or client application specific information. Example customizations might be

> Accept this string argument as a pointer to a null-terminated C string.
>
> Accept this other string argument as a n integer count and a pointer to
> the first character.

We should try very hard to allow a textual separation between these sorts of information and the core interface. Annotations should be addable to definitions imported from other modules or files.

-- SECTION 5 --

On Ed and Dan’s second round:

(6a) Date-and-time does seem appropriate to include.

(6b) ISO-Object-Identifier yes. However, a general "name" datatype is inappropriate. Name is more of a role than a type.

(24) On table, we need to decide on an adequate external representation, not on the "single right one." It is the nature of external representations that there is rarely a single best one. Fortunately, there is also no need to pick a best, since the external rep is not the computational one.

-- SECTION 6 --

I need to talk to Chi about the meaning of some of his comments attached to the latest IDN. Comments on those later.

-- Craig