Date: Fri, 8 Mar 91 17:38 GMT
From: Brian <UDAA000@HAZEL.CC.KCL.AC.UK>
To: SC22WG11@AWITUW01
Subject: CLID subsetting etc

First I apologise that I have been pretty silent since Monterey on WG11 issues. The usual reasons, compounded by travel problems last month since British Rail are unable to admit that subzero (Celsius) temperatures and solidifed water, as precipitation or otherwise, are things they should allow for.

Particularly have I failed to provide views on the Treat/Hamilton and Yellin papers.

At this time, not being in my office but having a few minutes, I can do no more than make a few remarks on Ed's recent mailing about subsetting:

ALL datatypes recognised as having a separate identity should appear in the TEXT of CLID though - as I said from very early on - some could be there in an informative and not a normative context, for completeness. A good example from Ed's proposals is (or would be, since I don't agree with him!) STATE and ENUMERATED - you have the one, but an informative note related the other to it. ORDINAL is a better example perhaps because there I in fact DO agree with Ed!

I believe that this all comes down to the conformity rules. All options in standards are evil and I am against them on principle, but even options have different degrees (levels?!) of depravity.

The WORST are things that are left out altogether, because you don't ever see them and if people want to bung them in they make it up the way that suits them, leading to incompatibilities.

Next are things MENTIONED but only to say they aren't covered in the document. This at least means that one is warned to look out for them lurking in some vicious implementor's product.

Next are things DESCRIBED but left IMPLEMENTATION-DEPENDENT; marginally better than the last because it may well be clearer what they are. These may or may not be normative; in some ways it's better if they are, in others if they're not.

Next are the IMPLEMENTATION-DEFINED things, where they ARE normative but at least the conformity rules require the implementor to specify what has been done. You then at least know where you are without having to find out the hard way, even if where you are is in a pool of nasty stuff.

Then (and orthoganal to much of the above) is the matter of OPTIONS: the thing is completely defined but the implementor chooses whether to provide it or not. In conformity rule terms, "you don't have to provide this but if you do you have to do it this way". Such options are evil just as all the others are evil but they may be lesser evils. My back-of-neck antennae inform me that in the CLID case they are indeed the lesser evils and they may have to be swallowed, like nasty medicine (which won't do us much good but...)

PROPOSAL:

Since drafting conformity rules which tie the ******s down enough in situations like that I propose that one aspect of the Arles meetings should be to review the WG12 conformity rules TR guidelines, I haven't the number here because I'm not in my office. (I suspect the RPC folk might find them useful as well, suitably adapted to their environment.)

Points of detail:

I've mentioned STATE and ORDINAL, and my assertion (demonstration, I believe!) that BOOLEAN and BIT are truly distinct is on the record and won't be repeated, you'll be glad to hear. With gritted teeth I have to admit that Ed has a point on DATE-AND-TIME and will be sweating until I find a way out of it. RATIONAL needed be normative but it must be mentioned. Mostly I agree with Ed's detailed analysis EXCEPT that I think the operational aspects of LIST and ARRAY are sufficiently different to require them to be distinguished (though I have no objection whatever to an informative discussion of their relationship and the way that one can stand in for the other.) Without access to my papers I can't be sure but I thought Ed Greengrass had definitively made the distinction in a paper some time ago.

As for BITSTRING, array of BIT doesn't cope unless you add clumsy procedural definitions of the operations of MASK, SHIFTLEFT and SHIFTRIGHT. I am uneasily aware that this seems to be getting dangerously close to the underlying representation level; however, the importance of this for things like pattern-matching (maybe if we called it PATTERN we'd magic away the problem!) makes be believe that this is a misconception - it IS fundamental (and different from ARRAY or LIST of BIT), but just happens (like BIT itself) to be trivial to translate down to the representational level. Hence I disagree with Ed that BITSTRING is there "precisely for the purpose of giving implementations a hook on which to hang optimisation of ARRAY of BIT/BOOLEAN". No fear, mate! One of the less attractive, idol-has-feet-of-clay aspects of Pascal is the unlovely PACK/UNPACK duo. The justification for BITSTRING, or BITMAP, is that it can show pictures, things of beauty in themselves that need no sordid utilitarian justification!

Brian