From: edbark@cme.nist.gov (Ed Barkmeyer)
Message-Id: <9103121720.AA00851@radio.cme.nist.gov>
To: chi@osf.org, dmy@ibm.com, rabin@osf.org, sc22wg11@awituw01.bitnet
Subject: Meek's 4-level categorization of datatype notions

All of the debate on datatypes has convinced me that there are many levels of abstraction and, even worse, more than one dimension to "abstraction". What we need is not so much a nomenclature, but a clear notion of "levels of concern". That is, we need to decide at exactly what level of concern the CLI Datatypes standard resides, and what notions of datatype are appropriate and inappropriate at that level. Then we need to decide whether the CLIPC and the RPC are (respectively) at the same level of concern or not, and if not, what we need to do about the difference.

I am reasonably certain that the ASN.1 concept of datatype is a level lower than that what is wanted at the language and interface level. What is not clear to me is whether any two of the "user" level (what I mean), the "language" level (what I can write), and the "interface" level (what I can exchange) are the same.

It does seem to me that at the Service level - the interface level perceived by the user (which may be different from the program or language perception) - the CLIPC and RPC should have the same concerns and have the same level of abstraction. But at the Protocol level, the RPC must have a lower-level of abstraction, in which the higher level of abstraction may be understood as "labels" to be carried on "more fundamental" "interchange" datatypes. In most existing OSI standards, this kind of distinction exists, but the "program" level is restricted to notions proper to the OSI service itself, so that the "labels" can be predefined, in the standard itself. RPC (and RDA, FTAM done right, and maybe TP) have the problem of having to carry user-defined information which goes beyond the notions of the OSI service, and cannot be standardized per se in either the service or protocol. This has all kinds of consequences for the architecture of Application and Presentation layer software, and specifically what can be precompiled, but the only aspect which is of concern to us is that there is not in the RPC a one-to-one onto mapping between the user/program interface information units and the information units in the protocol. Rather, a user-defined interface information unit must be mapped into a "linguistic" label and some ASN.1 "operational representation" datatype. As in ASN.1 itself, it is often possible to achieve the linguistic labelling by the position of the information unit in the interchange, although this technique breaks down on encounter with Choice datatypes.

I envision the number of "operational" datatypes to be smaller than the number of CLI datatypes, but there is no reason to make this many-to-one mapping visible (or annoying) to the user. He should have as rich a set of linguistic datatypes as may facilitate optimum communication and marshalling schemes. That is, the CLIPC/RPC should serve communication between programs written in the same language efficiently, if possible, AND permit programs which are written in two different languages to interchange common information units via marshalling schemes which produce different optimal representations in the two different languages. This argues for a relatively large set of CLI datatypes, and a relatively small set of "operational" types which appear in the protocol, a possibly optional labelling scheme, and a standard rendering of the CLI datatypes into the operational types.

-Ed