--

SC22/WG11/N316

To: SC22/WG15 - POSIX
From: SC22/WG11 - Language Bindings
Subject: Review of LIS and C binding, comments on WG15/N244

During its meeting in April 1992, WG11 has reviewed document WG15/N244: IEEE 1003.1LIS and IEEE 1003.16 - Open Issues.

It occurred to WG11 that the LIS POSIX documents are developed in accordance with the IEEE/TCOS document on Programming Language Independent Specification Methods, but that this IEEE document has no international status. WG11 has reviewed this document, and endorses the document in general.
Since it can be expected that more US domestic language independent standards (which in a later stage can become international standards) will be developed based on this document, and since there is a perceived need to have a Guidelines document on the production of language independent specifications, WG15 is advised to seek means to promote this document to a ISO technical report.
As the foreword of the TCOS-LIS document already states, sections of the document that (re)define notions that are also defined by standards currently under preparation by WG11 (Common Language Independent Datatypes and Common Language Independent Procedure Calling Mechanism) should be updated to refer to the WG11 standards. When then the POSIX LIS 9945 standards are updated according to this new technical report, full alignment with the WG11 standards is ensured.
The scope of the TR should however be broader than only POSIX: it should serve as a generic guide for all language bindings projects within JTC1. It this way the document may be seen as an companion document to TR 10182 (and as the current TCOS document makes use of TR 10182, the relationship between the 2 documents should be clearly documented).

For alignment of the LIS POSIX standards and the work currently under development in WG11, it is advised to make use as much as possible of WG11 documents (refer to them), even if they are not yet approved standards. The critical part in this is the IDN; WG11 hopes to have a stable IDN by mid 1993; depending on the progression of the 9945-1 LIS document that IDN could be used in a later draft of the IDN.

WG11 has the following comments on the open issues (WG15/N244):

Issue 1

It is not clear why conformance with the Language Standards is needed for implementations: what one might expect is that the implementation of the POSIX interface is done on a development system with a standard compilation system. However, from that moment on, the implementation should bind the functionality, defined in LIS standard, to the language interface defined in the language binding. Hence, conformity with the language standard is not necessary. It is the language binding itself which must conform to the language standard, as well as the intent of the LIS specification and the language binding guidelines, but these are requirements on the binding standard and not on the implementations.

Issue 2

As there is currently no standard for a storage model a new standard should be prepared. However, as the WG11 standards are explicitly independent of any storage model, WG11 cannot be of any help at this point.

Issue 3

If a model is needed, we would advise you to consider the work of X3H7 and the equivalent work of OMG.

Issue 4

Issue outside of scope of WG11.

Issue 5

It is clear that 9945-1 intends that some of the datatypes may be extended by implementors, while other datatypes may not be, in order to insure portability of conforming software. Similarly, it is clearly intended that an implementation may supply additional procedures, but may not modify the argument types of any of the standardized ones. It appears, therefore, that extension is a meaningful notion at the LIS level, and in fact, extension at the language-binding level should only be permitted to the extent that it is a mapping of the conceptual extension permitted by the LIS version.

Extension is not an issue which WG11 has studied. It is not clear that it is an issue for Language-Independent Datatypes per se, but it is clearly an issue in "language-independent specification". It appears that in the case of 9945-1, the permissible extensions are intimately bound up with the semantics of the abstract types, rather than the formal datatypes, and thus it would be surprising if there were a general rule or rules. However, it would still be desirable to describe the permissible extensions to a particular L-I datatype in a standard way. E.g. State/Enumerated types are extended by additional values, Record types by additional fields, Choice types by additional alternatives, and List/Array types, such as BitString, by additional values at the end. And each of these would need some appropriate "extension syntax".

However, it is not clear whether extensions to the datatypes can be permitted without impairing the portability of conforming programs in the general case. In general, the implementation must be able to distinguish a conforming value provided by the application from an extended value provided by the application, in order to avoid references to components whose values are not intentionally defined by the application, and may in fact be bad physical references. Similarly, the implementation may not return an extended value to an application variable which was declared to accept a conforming value. The assumption that the local description of the standard-defined datatypes will always be obtained from a locally provided header file, and that programs will use only these declared datatypes in the actual interfaces, is not valid unless the standard requires the application to do that, and that may not be possible in all ISO programming languages.

A technique used in certain ISO standards is to specify some additional fields, cells, or alternatives as to datatype, with the provision that the semantics of those fields/cells/alternatives is implementation-defined. This technique can even be extended to procedure arguments, so long as there is some standardized "vanilla" value which means "this extension is not present". This presents interesting problems for the object model, but resolves the datatype extension problems, if consensus on the (representation) types of these implementation-defined values can be reached. This is admittedly an "ugly" means of solving the problem, but it may be the only means which insures portability of conforming application software.

Issue 6

Issue outside of scope of WG11.

Issue 7

Issue outside of scope of WG11.

Issue 8

Issue outside of scope of WG11.