Title: Document types in Language Independent work

Source: Danish Standards Association

Date: 1995-05-28

Distribution: JTC1/SC22/WG11 (SC22WG11.510)

Status: Member body contribution

Action: for discussion at WG11 meeting in May/June 1995
Document types in Language Independent work

1. Introduction

Following discussion in WG11 we hereby give an overview of document types in language independent specification work.

The purpose of language independent specifications is to make it economical and consistent to have the same semantic set of specifications for a specific set of functionality available in a number of programming languages.

2. Language Independent standards

The main document is thus the language independent specification (LIS) of the subject in question, for example a set of procedures for handling graphics.

The language independent specification is done according to standards for specifying language independent semantics. These standards specifies which data types are available, how they are named and their properties (LIDT), and also how procedure calls etc are handled (LIPC).

3. Bindings

There are then a number of ways to bind the LIS to a programming language:

3.1 Binding a language to LIDT and LIPC

If a programming language specifies how its data types and procedure calls are bound to the LIDT and LIPC standards, then all LIS standards are automatically specified for use from the language, and no specific language bindings for the LIS need to be produced.

3.2 Thick binding to a LIS

A binding to a LIS can be done specifying all of the semantics once again in the binding; this is called a "thick" binding. This may be appropriate when a programming language has some specific properties, which makes the specification much simpler by rewriting the semantics, as some specifications in the LIS are dealt with automatically in the language. It might also be caused by binding problems, for example return values may be of different data type depending on if an error has occurred, or the function returns normally. A problem with "thick" bindings is that as the semantics are rewritten, there may be introduced semantic differences from the original LIS.

3.3 Thin binding to a LIS

The binding, where the semantics of the specifications are not copied, but where only syntax is described for a programming language, is called a "thin" binding. In some cases the "thin" binding is still quite thick, say more than half of the size of the LIS. The "thin" binding has the advantage that because semantics are not repeated, consistent semantic behaviour across language bindings are guaranteed for the LIS.

3.4 Reference card binding to a LIS.

A popular documentation type for commercial products is the "reference card": a short listing of the syntax of the specification in question in the appropriate programming language syntax. No semantics are given. This can be seen as a "thin" binding to a LIS, and is very usable to programmers as a quick guide to the standard.

4. Guidelines for writing LIS standards

When writing LIS standards, one should take care that all of the binding methods described above can be employed. For the "reference card" binding method, that would include that all functions have section numbers.

As there may be several styles of procedure calls stemming from different programming languages, a LIS writer should consult the prospective binding writers to detect such special issues, to ensure good binding adaptability. This could for example lead to specification of alternate LIS APIs for the same semantics, but adapted to the different styles of procedure calling for the different languages, when one language can return an error value and a normal result in the same variable - where other programming languages cannot. Another example may be different versions of the same procedure where one version operates on a global system variable, and other versions have all parameters explicitly specified.

To ensure that the LIS standard be usable, it is recommended that the LIS writer cooperate with at least 2 prospective or actual language binding writers.

5. Conclusion

We think that the above guidelines will enhance the usability of LIS standards considerably, especially producing "reference cards" as thin programming language bindings is seen as a way to economically and quickly produce bindings, and also to produce documents that can be sold well to programmers, as it is a proven commercial document type.