From: Willem Wakker
To: SC22/WG11
Subject: Draft Pascal binding for LIPC
Date: March 11, 1994

In response to the UK request to supply a binding to LIPC, I have tried to produce a first skeleton of such a binding. It is by no means complete (or even correct I assume) but should serve as 'something on the table' for the next meeting. If we can update/improve this document by email discussions before the meeting, so much the better.
I took Pascal as the language to be bound to since DIS 11404 already has a sample Pascal binding.
I assume that the binding (cf N376R, sect 7) should define how the IDN syntax is mapped on the syntax of the language that is bound. The resulting LIPC binding is then remarkable small.
Am I missing something? I will follow the LIPC document section by section. Maybe a more 'natural' order will appear later.

I also appended a number of notes on N376R - LIPC: when you really READ the text you always find something.

- Willem.

==========

Draft Pascal binding for LIPC

==========

7.2 Interface Type Declarations

As Pascal does not allow separate compilation, the binding of the interface-type shall be implementation defined.

7.3 Import Declarations

As Pascal does not allow separate compilation, the binding of the import declaration shall be implementation defined.

7.5 Value Declarations

The IDN value-decl designates the Pascal constant-definition (IS 7185, sect. 6.3).

The IDN types integer-type, real-type and character-type designate the Pascal types integer-type, real-type and array of char.

7.6 Datatype Declarations

See DIS 11404, annex E.

7.6.1.7 The procedure datatype

The IDN procedure-type designates either a Pascal procedure or a Pascal function. When the procedure-type has "returns" "(" return-argument ")" in its declaration, the it designates a Pascal function, and the argument-type of the return-argument designates the result-type of the function- heading. When the

procedure-type has no "returns" "(" return- argument ")" in its declaration, the it designates a Pascal procedure.

When the direction in the IDN argument-declaration is "in", the corresponding argument designates a Pascal value- parameter-specification. The parameter in the actual call is passed as described in section 5.2.1.1: Call by Value Sent on Initiation.

When the direction in the IDN argument-declaration is "out" or "inout", the corresponding argument designates a Pascal variable-parameter-specification. The parameter in the actual call is passed as described in section 5.2.1.4: Call by Value Returned when Available.

The IDN argument-name in the argument designates the Pascal identifier of the identifier-list from the value-parameter- specification or the variable-parameter-specification, and the IDN argument-type designates to corresponding Pascal type-identifier from the value-parameter-specification or the variable-parameter-specification,

The optional argument-name in the IDN return-argument is ignored.

The meaning of the termination-list is implementation defined.

==========

Notes on N376R

==========

1. 7.2.1 1st sentence
   I assume that the identifier is used in the interface-body, and is not the identifier from the interface-identifier. Reword to "If an identifier in a type definition in an interface-body is used to refer to ..."
2. 7.2.2 1st sentence
   See remark at 7.2.1
3. 7.4 Declarations
   Does this one-liner merits a full section on its own? And if so, should not 7.5 be 7.4.1 And where are the procedure-decl and termination-decl defined? Should the procedure-dcl not be the procedure-type, which is already included in type-decl? Should the title of 7.6 not be Type Declarations?
4. 7.5 Value Declarations
   Move the rule of value-expr immediately after the rule for constant-type-spec. Why is the enumeration-literal missing from value-expr? The 3rd paragraph (An interface shall only define constants of ...) is a restriction on the use of the IDN. Is this explained anywhere?
5. 7.6.1.7 The procedure datatype
   As the difference between a function and a procedure is the presence of the "returns" "(" return-argument ")", it is more convenient in a binding to have this as one non-terminal.
6. 7.6.1.7.2 Procedure values, 7.6.1.7.3 Subtypes
   This text is also in 11404. Should it remain here, or just a reference? In any case, most of it has hardly anything to do with the IDN.