From: Willem Wakker, Convener SC22/WG11
To: Mark Woodman, Convener SC22/WG13
Subject: Liaison statement on Binding of Modula-2 to LI Arithmetic

WG11 applauds the effort made by the WG13 committee in attempting in the Modula-2 standard (CD 10514) to provide a binding to the Language-Independent Arithmetic Part 1 (CD 10967-1). Indeed, part of the Modula-2 approach is now being incorporated in the DIS version of 10967-1. We are, however, concerned that the current Modula-2 draft does not guarantee that arithmetic will be "implementation-defined".

Specifically, we suggest:

1. The IEEE flag in LowReal and LowLong should be defined as:

   TRUE if the implementation of the datatype conforms to IEC 559:1989 (IEEE 754:1987) or IEEE 854:1987 in all regards: representation, arithmetic, rounding modes, return of NaN values, etc.
   FALSE in any other case.
   That is, reference should be made to the ISO/IEC version of the IEEE standard, and "almost IEEE" (non-conforming) implementations may NOT return TRUE. Requirement for a similar flag (conformance to IEC 559 only; conformance to IEEE 854 is not considered to be of enough value to merrit a special treatement in this respect) is being added to LIA.

2. It should be made clear to which IEEE format from which IEEE standard the datatype conforms when LowReal.IEEE or LowLong.IEEE are true.

3. The "ISO" flag in LowReal and LowLong should be renamed "LIA1", or something the like, since reserving the name "ISO" is presumptuous.

4. The LIA1 flag ("ISO") should be defined as:

   TRUE if the implementation of the datatype conforms to ISO 10967-1:199x in all regards: parameters, arithmetic, exceptions, notification;
   FALSE in any other case.
   and include the following additional requirement:

   "For implementations not conforming to ISO 10967-1:199x, a statement of the differences between the arithmetic performed by the Modula-2 implementation and the requirements of ISO 10967-1:199x shall be supplied. That is, the implementation shall document which of the properties required by ISO 10967-1:199x hold for this datatype, and which do not."

   The objective is to make the arithmetic implementation-defined, even when it does not conform to any standard. LIA-1 identifies a list of properties on which the user can base arithmetic-sensitive algorithms and even in most non-conforming environments, only one or two of them will not hold.

5. CD 10514 expressly says that the parameters in LowReal and LowLong: r, p, emin, emax, etc. are "those of the representation". This is not particularly useful. When the implementation conforms to LIA, their meaning, and the expectations a user can have of them, is much more carefully defined by LIA. When the implementation does not conform to LIA, the "parameters of the representation" are not necessarily well-defined. For example, does "p" (precision) depend on the sign of the number? Does it depend on the implicit location of the binary point? Is "fmax" the largest positive number, the largest absolute value, or the largest number whose negative is also representable? (On at least one existing machine, those are three different numbers.) On the other hand, non-conforming implementations such as Cray may conform in the meaning of the parameters, but not in some other aspect, such as rounding error or notification.

   We suggest:

> "The parameters shall have the meanings defined in the LIA model if LIA1 ("ISO") is TRUE. If LIA1 is FALSE, the implementation shall document whether each parameter has the meaning defined in the LIA model, or what (nearest equivalent) meaning is assigned to it by the implementation."
>
> LIA is itself being modified to provide some further guidance in this area.

The following changes are being made in the DIS version of LIA-1 and will affect the Modula-2 binding. We believe these changes will further facilitate the binding:

(a) alignment of IEEE and LIA definitions of emin and emax.

(b) requirement for, and definition of, the IEEE flag in LIA.

(c) guideline for language-bindings vis-a-vis non-conforming implementations, including requirement for an LIA-conformant flag and documentation of individual differences.

(d) introduction of "modular" integer arithmetic, in which, unlike "bounded" integer arithmetic, overflow detection is not required.