

## Report on WG11 meeting Tampere 21-23 August 1992

After identification and copying of papers needed, and assignment of temporary (Tampere) numbers T1, T2 etc. to them, the meeting began at about 10.45 on Friday 21 August 1992. (This report uses the WG11 numbers later allocated to these)

Present were Ed Barkmeyer (USA, CLID editor), Ken Edwards (USA, CLIP editor), Brian Meek (UK), and as observer and invited participant Roger Scowen (UK and WG17 Prolog convenor).

It was noted that the convenor, Willem Wakker, would not be arriving until late Saturday, and Brian Meek was appointed chairman in his absence. Since no formal agenda was available, this being primarily an editing meeting to review the latest drafts of the CLID and CLIP documents, the draft agenda for the October meeting was used as a guide. Administrative items 1-4 and 14-15 were left until the convenor arrived. Under item 6 (bindings TR) it was noted that the letter ballot comments should be reviewed though their resolution would have to be left to the October meeting, but this item too would be deferred until the convenor arrived. There were documents relating to items 8 (LIA-1) and 11 (RPC) that might need some agenda time, as also was the case with 12a (Posix) and 12b (PCTE). Nothing was known that would need discussion under the other agenda items. It was therefore agreed to deal with item 7 (CLID) on the first day and item 6 (CLIP) on the second day (Saturday).

## 7. Common language-independent datatypes

### 7.1 Review of Roger Scowen's comments (N309)

The UK panel had reviewed and discussed N309 with Roger Scowen and had produced a response document (N324). In the meantime Roger had incorporated the substance of N309 in a UK contribution to SC22 (N1201) which as explained in a later electronic mail message was an individual submission, not an endorsed UK position. He had also produced for this meeting documents his reactions to Brian Meek's draft of the response document now designated N324.

Brian Meek began by going through N309 and N324, explaining where the UK panel agreed with N309 and expanding on the reasons given in T1 where disagreement was expressed. Much of the disagreement arose from different perceptions of the scope, purpose, and level of abstraction of the CLID document; the panel agreed with many of the statements of need but did not believe that the CLID standard was the appropriate place to address them. Many of them needed to be addressed in mapping standards (bindings) or cross-language service standards such as CLIP, but did not belong in CLID. The UK panel agreed that definitions could be improved and tightened but felt that the necessary precision and rigour could be attained without introducing mathematical formalism which would render the draft standard unacceptable to much of the readership in the language communities.

Roger Scowen then introduced his reply document and initiated a discussion on the issues he had raised and still caused him concern, as an individual or as a language WG convenor. The most important of these was his view that a datatype is not just a set of values but also the set of operations on those values. The discussion was not intended to resolve the issues but rather to clarify them, though some had already been addressed in the latest version (version 6) of CLID (N319).

On the question of using LaTeX, the project editor said that this is not a practical possibility for him but he intended to use double column for future drafts. He also accepted the suggestion that issues in the issues list should each have a unique and constant identifier for ease of comparison between drafts.

WG11 did not feel able to accept Roger Scowen's position on all of the issues raised. In particular members argued that, since many potential applications of CLID did not need the operations, as Roger had pointed out in his conformity recommendations, it made sense to keep those separate. Because of the many

different uses of datatypes in different contexts, it would be very difficult to achieve agreement on and acceptance of a normative set of operations for all datatypes. The place for normative operations was in standards based on CLID that needed them; the role of CLID was as enabling standard, to establish a common set of definitions based at the static level of value sets, on which such operational standards could build.

However, numerous points were raised during the discussion which later proved helpful in resolving outstanding issues in N319.

## 7.2 Review of version 6, N319, and editor's notes, N320

Ed Barkmeyer introduced N319 and went through the notes in N320 explaining the changes to the document and the resulting changes to the outstanding issues list. The remaining outstanding issues on pages (i) - (iii) of N319 were then discussed, with the outcome as follows. (Roger Scowen participated in the discussions but not in the decisions on these issues.)

Issue 1 (move datatypes from clause 7 to clause 9):

No moves are necessary. This is not a major issue: what is important is that datatypes that are similar but can be distinguished are distinguished. Other taxonomies of datatypes are possible which might entail such changes but the current one is viable for these datatypes and these changes would not bring about noticeable improvements.

Issue 2 (array, etc subtypes of a more general Map datatype?)

It was agreed that the term "mapping" must be avoided except in the context of mappings between CLID datatypes and language datatypes. Brian Meek pointed out that it was an established UK requirement that a generic definition of "aggregate" datatypes be included, together with explanation of the derivation of the various kinds of aggregate in terms of their structural and conceptual properties - homogeneity, dimensionality, indexability etc. This would have been in formal UK technical comment accompanying its NO vote had it chosen to vote in that way on the 1st CD, and had only received little attention so far because WG11 had concentrated on the formal comments from other NBs.

After discussion it was agreed to adopt this taxonomy, with the current aggregates 7.3.2 - 7.3.7 being move to a new subsection "Aggregate datatypes" and 7.3.1 (Choice) together with 7.1.14 (Pointer) and 7.1.15 (Procedure) and probably 7.3.8 (Declared- generator types) remaining in a (possibly renamed) "Generated datatypes" subsection.

Issue 3 (model for Set and Bag?)

After discussion it was agreed to adopt the mathematical model to provide the relationship and distinction between Set and Bag but to provide explanatory text and Venn diagrams as illustration.

Issue 4 (description of Table generator)

This could be rolled up into the generic taxonomy for aggregates; multiple keys would be included (i.e. Table can be multidimensional).

Issue 5 (user-defined datatypes and generators)

It was agreed that CLID must have IDN definitions of everything it needs regardless of whether or not these are needed for CLIP or RPC, the only constraint being that they must not introduce conflicts or inconsistencies. It was noted that RPC would include IDN definitions not needed for CLIP or CLID and CLIP would include IDN definitions not needed for CLID and perhaps some not needed for RPC. A

solution to the principle of "the same IDN (definitions)" was that all three standards could contain an informative annex containing all IDN definitions distinguishing those used in that standard from those used only in one or both of the others. That was a matter which could be resolved with the RPC group at the October meeting.

Issue 6 (is direct compliance meaningful?)

The answer is yes: mapping (binding) standards would conform directly, and the possibility must be allowed for interface definitions without their own standards or relevant defined mappings to be able to use CLID datatypes and claim conformity.

Issue 7 (use of outward mappings?)

Outward mapping had to be defined in order to identify formally which CLID datatypes correspond to the internal language datatypes and where necessary to document constraints or minimal requirements which may be specified in the case of some datatypes.

Issue 8 (radix other than 10?)

For the purposes of the definitions in CLID itself more than one radix is not necessary so this is not at CLID issue at all. It may be important for languages with literal denotations in their syntax for different radices, but that is a matter for the mapping standards.

Issue 9 (informative annex with example mapping?)

It had been agreed to include this in response to a French comment and in principle it would be a useful addition. However, it would appear on reflection that it might cause objections from the language WG or community concerned if WG11 itself provided this. It was agreed that this could be raised at SC22 plenary during the discussions on cross-language issues - when it was possible that a WG might volunteer to provide the annex - and also perhaps to ascertain if France would object to the 2nd CD if it were not possible to provide such an annex at that stage.

Issue 10 (compliance of various kinds of entity?)

Though this had not yet been done the project editor thought that there was now enough material in existence on possibilities or actual attempts at compliance to enable this to be done in time for the 2nd CD. It was therefore agreed that this would be included in the final 2nd CD draft for the October meeting.

Issue 11 (reference model only? compliance rules?)

It was agreed that CLID had characteristics of a reference model but went beyond that. As stated under issue 6, direct compliance is needed so that products such as cross-language or cross-entity utilities can reference, use, and claim conformity to CLID, especially where no relevant standards exist which would allow indirect compliance. In addition, the possibility of direct compliance will encourage future products, including new kinds of products, to use standard CLID datatypes directly rather than defining their own and then needing mappings. No changes to the compliance rules were proposed; some of the subsidiary issues under this heading were not addressed.

Issue 12 (integer-modulo, modulo)

It was agreed to include modulo as Cyclic-enumerated and integer-modulo renamed as Modulo, with Enumerated as a generator from State and Cyclic-enumerated as a generator from Enumerated. This was thought to be a noticeable taxonomic improvement.

## Issue 13 (annex B)

This is a minor issue, to be decided when views of French and German members of WG11 have been obtained.

## Issue 14 (Pointer to Procedure)

This is an issue in CLIP rather than CLID, relating to invocation and parameter passing and not to the conceptual notion of datatype; it is a part of the procedure datatype characterising operation "Apply". For CLID it is simply the application of Pointer to the Procedure datatype and can remain as it is.

## Issue 15 (Nonnegative and Inorder)

The project editor felt that he could now resolve this in the light of the discussions on the relationship between CLID and LIA and the use of mathematical formalism under agenda item 7.1.

## Issue 16 (atomicity of values)

Values of primitive datatypes are necessarily atomic, values of others may or may not be. This will be clearer with the introduction of the separate category of aggregate datatypes. This is a minor issue of intent and understanding only, not a major technical issue.

## Issue 17 (dense/discrete, approximate/exact)

This issue was resolved by the removal of the concept of dense v. discrete from the document, which was not necessary to it. The concept of approximate v. exact was important and would be retained. Index datatypes should be finite and exact, not discrete. Selecting, excluding, and Set datatype should be limited to exact rather than discrete. Scaled is not an approximate datatype, it is an exact datatype which through operations on its values is used in some applications to provide approximations to values which it cannot represent directly. The concept of "numeric datatype" was not required in the CLID definitions though it could appear informally for explanatory informative text if this were felt to be useful.

In summary, all issues had been resolved except issue 5 which needed discussion with the RPC group, issue 9, to be raised at SC22, some remaining parts of issue 11 still to be addressed by WG11, and issue 13, to be decided after the views of the French and German members of WG11 had been obtained.

## 6. CLIP

### 6.1 Review of N317

On Saturday morning the task list was reviewed and the group then turned to CLIP. Ken Edwards introduced N317 and remaining issues discussed. In response to a query by Roger Scowen it was confirmed that CLIP covers no representational issues and in particular did not prescribe a required serial ordering of the individual elements of aggregate datatypes; however, it was required that mappings and hence marshalling and unmarshalling must preserve the integrity of the individual elements. How this was achieved was immaterial to CLID.

In response to queries about the scope statement it was agreed that in clause 1.2 the term "processor" would be better replaced by "language processor" in the sense of TR 10176, that in the exclusions reference to "data processing system" would be better replaced by "configuration" in the sense used in the conformity TRs, and that a separate explicit exclusion about ordering of actual parameters would make clear the point just raised about aggregated values.

Changes would entail corresponding changes in clause 3, in which also the definition of "procedure" as the

Procedure datatype was not what was required, and an explanation was needed that "client procedure" in the sense used in the CLIP document represented a generalisation of the usual language concept of "procedure".

## 6.2 Review of N331

The comments by Dan Yellin (N331) were reviewed. The meeting disagreed with the view of global data expressed in point 1: it must at least conceptually involve marshalling and unmarshalling and this in turn can be regarded as a form of parameter passing. However, for global data this marshalling and unmarshalling can be at any time before the call, at the time of call, during execution of call before access to the global data is needed, or at the time access is required. In that sense the "implied parameter passing" and corresponding marshalling is not constrained in the same way as explicit parameter passing by the specific types listed in clause 6.4.

On Yellin's comment 2, 6.4.1 and 6.4.2 distinguishes whether the parameter is first passed at the time of invocation or at some later time. Call-by-value-sent-on-request (6.4.2) is not "weird", it is an optimisation aid so that where an actual parameter may not in fact be used during a particular call, it is not passed, since it is not requested by the server. Languages may not "implement" this specifically but many standards do not preclude it. As for repeated calls, these can be handled by making the parameter a pointer which can be repeatedly referenced by the server, or a procedure which can be repeatedly called by the server.

As for 6.4.4, the document states explicitly that it is "primarily here for completeness", so that CLIP can cover all possibilities should the need arise. In fact 6.4.3 and 6.4.4 can both be improved since they should relate to the timing of the server's return of the value rather than what the client does with that value when returned, something which is a matter for the client language standard or CLIP binding standard. Hence 6.4.3 should be changed to call-by-value-returned-on-termination. The project editor was asked to do this and the corresponding changes in the body of 6.4.3 and 6.4.4.

The third of Dan Yellin's points was editorial.

## 6.3 Further discussion on N317

The conformity rules in clause 5 were considered and it was decided that the rules for language processors and for other information processing entities should be provided separately.

## 7.3 Review of N299

Len Moss's had submitted comments from X3J3 which had been deferred from the Baltimore meeting. Of the four sets, nos. 1 and 4 were already resolved in version 6 (N319).

On the array queries in no. 2, several of the varieties appeared to arise from needs for different kinds of parameter passing for different purposes, and effectively all array datatypes with particular constraints or characteristics that could be included in a mapping standard. However, they also appeared in some cases to be based upon an implementation model which was more akin to a sequence or flat list (without substructure) with an indexing capability overlaid upon it. After extensive discussion it was agreed that the Fortran community, WG5 and/or X3J3, could be asked to review this again once the aggregate datatypes have been recast. Brian Meek undertook to try to contact WG5 people on these matters during the SC22 meeting and to find out if a copy of the Fortran 90 standard was available for reference.

As far as inability to denote a null pointer was concerned, the outward mapping (or inward reverse mapping) of Fortran POINTER would be to Pointer and no null value would ever be passed out. The inward mapping could be defined with the constraint that if used with a CLID value of null an error would be generated.

## 7.4 Prolog datatypes

Roger Scowen circulated some excerpts from the draft Prolog standard and gave a short tutorial on the datatypes of Prolog. There followed a discussion on possible ways of mapping these. Terms had to be one of five types. Of these, integer and floating point were straightforward. Atom seemed to be best mapped as New CharacterString and Variable as Pointer to the union of all types. Compound was much trickier and would need considerable work. It was noted that CLID was never intended to be universal and there was a guideline used on whether a datatype was included depended on how many languages it appeared in. Roger Scowen said that he would not be concerned if the current CLID types could not accommodate Atom, Variable or Compound. The group thought that it was always valuable to shed new light on CLID by comparing it with actual languages, and if possible adjusting or adding to definitions to accommodate new datatypes. However, this did not seem easy in the case of Prolog Compound.

Issues raised in discussion with Scowen:

1. Unresolved: Should the CLID model of type Real be the same as the LIA model of type Real? They are different, because the CLID model purports to be the model of user requirements while the LIA model is a model of an implementation. How does this affect inward and outward mapping?
2. Many users of datatypes are solely concerned with communication of values and therefore only with representations which do not lose information. Other users are concerned with the operations at different levels of abstraction (characterizing operations vs. LIA). This should be reflected in the wording of 5.1 ii), or the addition of a Note thereto, and in the 3rd paragraph of 12.2.
3. Consensus that every definition in Clause 3 shall be consistent with the definition of the same term in ISO 2382, or shall indicate that it is not and why not.

Review of N333: Alignment with 11578-2.

1. Add restricted and unaliased modifiers to Pointer type. CLID style would make the form "pointer to (X): unaliased" rather than "unaliased pointer to (X)". Discuss syntax alignment in Paris.
2. Agree that attributes/extensions should be allowed anywhere and treated as comment in the general case. Documentation for a given extension must indicate where the extension is permitted to occur and with what syntactic/semantic object it is to be associated. Remove all "attributes" from CLID; remove clause 10; add extensions to clause 4.  
Disagree with the syntactic replacement of "[" and "]" by "%(" and ")%". Request rationale from RPC group. Discuss syntax alignment in Paris.  
<extension-source-OID> is extremely cumbersome. Suggest some alternative, possibly value-identifier.
3. If the type is called choice-type, the keyword should be "choice" and if the type is called select-type, the keyword should be "select". WG11 does not care which is chosen.  
The syntax of the choice-type should be changed to force the "default" alternative to occur last and at most once. The keyword for the default alternative should be "default".
4. The limitation on placement of arrays with non-constant bounds is exclusive to RPC, although it may be appropriate to LIPC.
5. The proposed parametrized type syntax, in which the datatype of the parameter is given, is neither necessary nor sufficient. Any value which is NOT an identifier is immediately identifiable as to its (base) type, while the type of an identifier may be defined by the declaration itself, e.g. State. What is important is that the parameter is known to represent a value, and thus the actual parameter must have the syntax of value-expression. If absolutely necessary, RPC may require a qualified-value.

6. WG11 understands the motivation for finer granularity of "in" and "out" procedure parameters, for records and pointers, but notes that in and out notions are specific to the use of the datatype in a particular argument (role) and not to the datatype in general. Thus the mechanism for declaring the finer granularity should not invade the type-specifier itself. This should be discussed jointly in Paris.

Examination of T14, a paper from Mike Sykes:

1. The ordering on type Character may be derived from the character-set standard (somehow) or may be implementation-defined. The ordering on type Character-String is ALWAYS user-defined, not implementation-defined, unless the user accepts some default implementation-defined ordering. In either case, it is not "implementation-dependent".
2. The notion "primitive datatype" is a taxonomic category primarily for syntactic purposes within the standard. It does imply that all values of a primitive type are intrinsically atomic, but types in other categories may also have atomic values.

Revisitation of Compliance (U.S. Comment 1):

1. Remove the definitions of Total and Partial compliance. Make all conformance "partial" in spirit, by requiring documentation of all types which are actually supported. Make no distinction between types in Clause 7 and Clause 9 in this regard.
2. Retain Direct and Indirect compliance, but revise the wording: A standard "complies" with the requirements of CLID for types and mappings. A product or implementation "conforms" to a standard.
3. An IPE which conforms directly to the CLID shall not use the CLID datatype names or syntax to designate conceptually DIFFERENT datatypes, and must document which, if any, characterizing operations are supported. 5.1 ii) should read "... any operations provided by the entity shall not be in conflict with the characterizing operations ..."
4. Changes to the Scope Clause.
  - a. Do not insert the words "reference model" into the Scope.
  - b. Move paragraphs 3 and 4 to Clause 6.
  - c. Strike "primitive and non-primitive" in paragraph 5.
5. Revise 12.3 to define the "reverse inward mapping". Editor is, however, to send suggested text to the active members via email before actually modifying the text.

Miscellaneous concerns of the Editor:

1. An open issue is whether the operation IsType on a choice-type is meaningful or even still appropriate. The proper characterizing operations and model are to be resolved by email discussion.
2. The comment from France which motivated the elimination of Undefined now applies to Void. Should Void have a single value and define Equal? Since the principal (perhaps sole) use of Void is in Choice-types, the resolution of this issue may be related to the issue of characterizing operations on Choice. It is argued that Void is simply the name of the "empty" Record, i.e. record(), which is not currently permitted by the syntax.
3. The elimination of attributes causes the distinction between "CLI-datatype" and "datatype-designator" to vanish. Both should be replaced by "type-specifier" for consistency with RPC.
4. The Range keyword should be retained (as an alternative), but range should be rolled up in Selecting.

5. "restricted to" should be changed to "subtype".
6. The RPC group should be asked to provide the value-syntax for Distinguished-Name.
7. The Editor and Roger Scowen should discuss the proper models for mathematically tractable datatypes. The issue of eliminating the "equivalence" definition for "Equal" depends on the resolution of the model for type Rational, and possibly others.
8. The explanation for the failure to include the "file" datatype should reference POSIX.