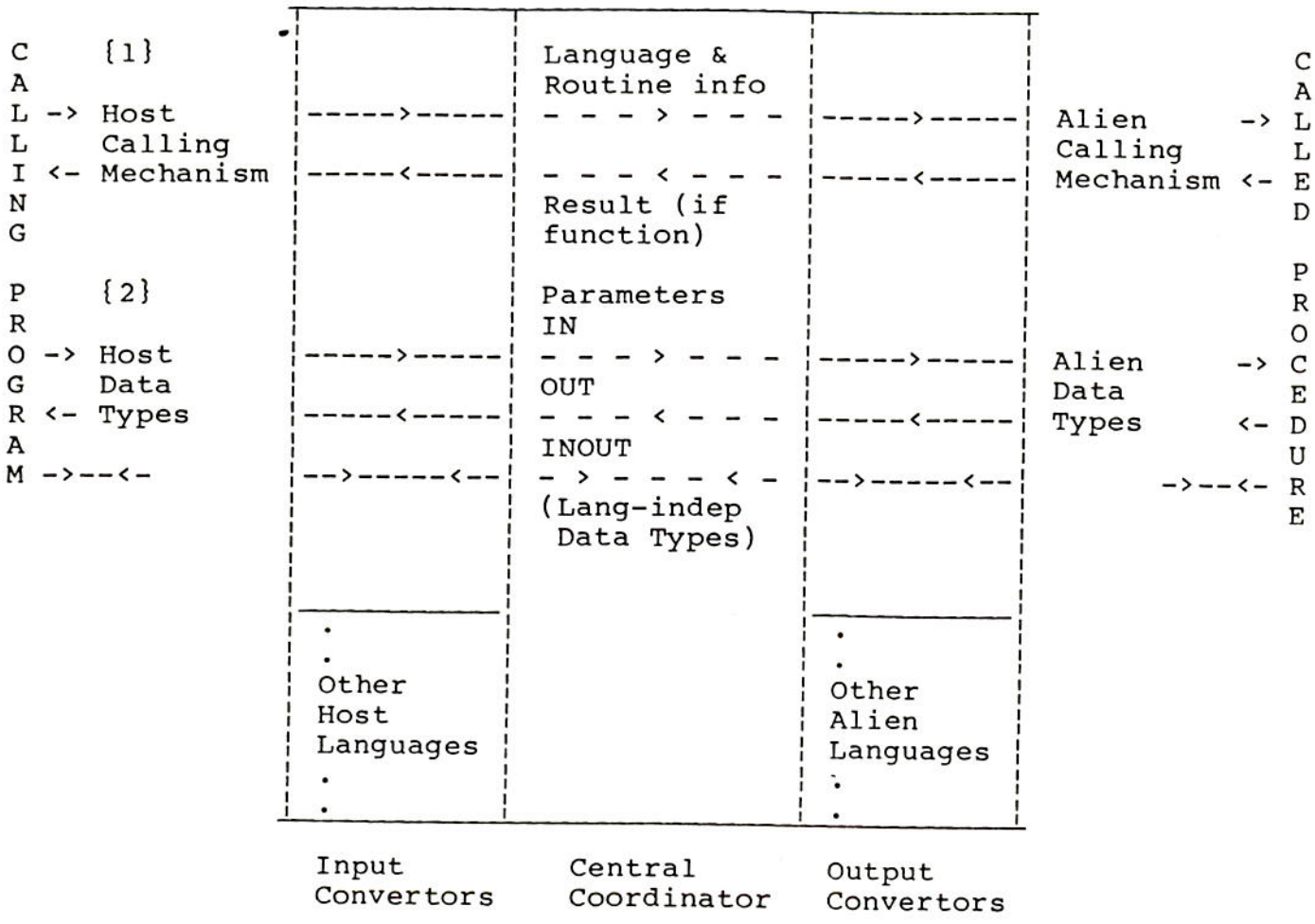
ISO TC97/SC22/WG11 N57

Common Language-independent Procedure Call - General Description



The general call of an alien routine would be

```
ALIEN (language, routine, routine-descriptor,          {1}
                 param-1, param-1-descriptor,          {2}
                 ...      ...                          ...
                 param-N, param-N-descriptor)          {2}
```

which would invoke the INPUT CONVERTOR appropriate to the host programming language.

The Input Convertor would convert any param-K, specified (in param-K-descriptor) to be of parameter class IN or INOUT, to the appropriate language-independent data type; and pass all such parameters, together with the information in 'language', 'routine' and 'routine-descriptor', to the CENTRAL COORDINATOR.

The Central Coordinator would invoke the OUTPUT CONVERTOR appropriate to the alien programming language.

The Output Convertor would convert the IN & INOUT param-K from language-independent data types to the parameter types of the specified alien routine, then invoke that routine. Subsequently it would convert any param-K specified to be of parameter class INOUT or OUT, and the alien function result (if any), to the appropriate language-independent data type, and return them to the Central Coordinator. This would pass them to the Input Convertor, which would convert them to the appropriate data types of the host programming language.

This process has the effect (from the host language's point of view) that param-K is treated as a VALUE parameter if it is of parameter class IN, as a RESULT parameter if of class OUT, and as a VALUE/RESULT parameter if of class INOUT.

Note that the process could be optimised - data conversions are conceptual, and might not actually be performed if the system knows the host & alien data types to be compatible.

There could be alternative entry points to the Input Convertor if the host language supports the concept of FUNCTION routines, e.g CHAR-ALIEN, REAL-ALIEN, INTEGER-ALIEN. If not, results of alien functions would have to be returned through an extra parameter (of class OUT).

The information contained in the descriptors would include:

ROUTINE-DESCRIPTOR

(C) routine-class : PROCEDURE or FUNCTION
(C) number-of-parameters (N)
(U) result-type*

PARAM-DESCRIPTOR

(U) parameter-class : IN, OUT or INOUT
(U) type* of parameter (or of elements, if an array)
(C) parameter-structure : SIMPLE or ARRAY
(C) host number-of-dimensions
(C) size of each host dimension
(U) alien number-of-dimensions
(U) size of each alien dimension

Notes: (C) denotes information which could be supplied by the host language compiler,
(U) denotes information which must be supplied by the user;

* denotes common language-independent data types;

There is no requirement for the 'shape' of an array to be identical in the host program and the alien procedure (e.g a 2-D host array may be mapped onto a 1-D alien array) - the defined order of occurrence of elements of an array of common language-independent data types determines the mapping. It would be an error if the numbers of host and alien elements were not compatible.

If the called routine is remote (i.e executed on different hardware from the calling program) this information must also be passed to the system, e.g in 'language' or 'routine-descriptor'.