This is the summary of how to use the CORBA facilities. We assumethat you have some IDL modules/interfaces defined and resident in the running Interface Repository. Also, let's assume that there is a server, registered under the name "StatServer" in the naming service, that implements an IDL interface *Foo* and that this has an operation *bar*. With this in place, in *S*/*R*, we can invoke the method in the server with the call

```
x <- .Corba("StatServer", "bar", arg1, arg2, .., argn)
```

and store the result in the variable *x()*. The first argument is the name of the server in the naming service. The second argument is the name of the operation. The other arguments are prescribed by the IDL definition of the operation. The key to the CORBA interface is how these arguments are converted from user-level objects to their CORBA counterparts. The rules are very simple.

Primitives Objects that correspond to CORBA primitives are converted to the corresponding primitive type. This handles integer, character, numeric, etc. vectors in *S*/*R*. Vectors of length 1 can map to arrays or simple CORBA scalars. Coercion from one primitive type to the expected CORBA argument type is done automatically.

Non-primitives More complex data structures that do not map to CORBA primitives are converted to CORBA objects of the type that the operation expects for that parameter. These are proxy objects whose requests are passed up to the user-level environment and dispatched, by default, to a function call of the same name as the CORBA operation name. Thus, they can be implemented by creating user-level functions with the appropriate name and arguments. The return value of these functions is converted to the type expected by the caller based on the IDL operation specification.

These are CORBA servers. We can create them outside of the context of an argument to a CORBA call. Instead, we can create them as top-level objects that run as pure servers. We have to be careful about asynchronous calls to the evaluator.

There are several other ways to use the *.Corba()* function. One is to use grouped-calls.

Ensure that the `LD_LIBRARY_PATH` is set to locate the different libraries. This includes the **libCommonCorba** in this directory, **libOmegaCorba** in **"Env –OMEGA˙HOME˝/Interfaces/CORBA/IDLTypes/**

Copyright (c) 1998, 1999 The Omega Project for Statistical Computing. All rights reserved.