Overview of the Math Megamodule

Dorothea Beringer, 1/2/98

1 Services of the math megamodule:

The math megamodule is a CORBA server that provides the following services:

- **SETUP**: Input is a string that contains the desired precision and format, outputs are the precision and format set by the service (the output is not really relevant). The service does not create any objects because the object *MServer* which offers these services is already created when the math megamodule server is started and registered at the ORB.
- **INVOKE**: Input is a string that contains the desired operation and two doubles containing the numbers on which the operation is to be performed. Output is again the desired operation (not really needed).
- **EXTRACT**: Input is a string (not used by the service), output is the result of the operation performed in the INVOKE service.
- **TERMINATE**: Input and output are strings that are not relevant. The object *MServer* is not deleted by this service, it gets deleted when the server terminates.

In contrast to the CHAIMS-syntax, neither the SETUP nor the INVOKE command return any handles. Therefore the math megamodule is not reentrant (no invocation handle), i.e. after the INVOKE command the result must first be read by the EXTRACT command before any further INVOKEs are called.

2 Overview of the files

We can distinguish between files on the client side, files on the server side and CORBA-specific files.

- **client side**: *client.cc*
- **server side**: *server.cc, MM.cc, MM.h, MM_i.h, MM_i.cc*
- **CORBA-specific files**: *MM.idl, MM.hh, MMC.cc, MMS.cc*

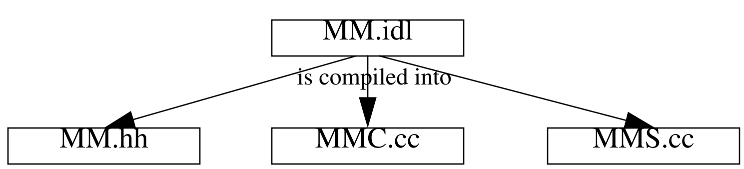
The file *MM.idl* contains the interface definition of the math megamodule in IDL (math megamodule is abreviated by *MM)*.

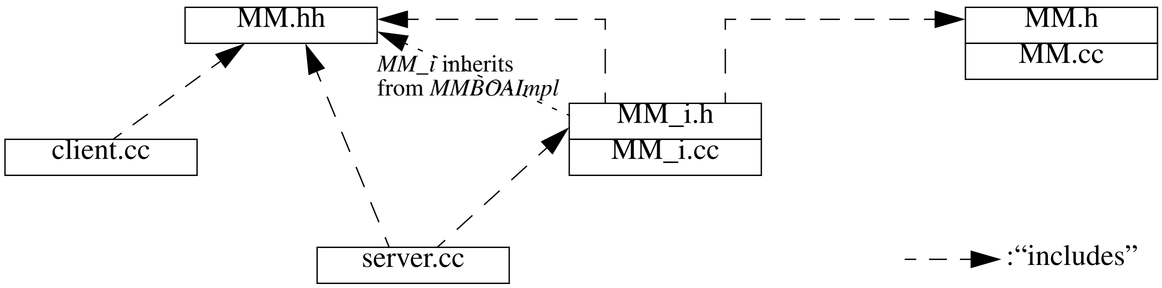
The files *MM.hh, MMC.cc, MMS.cc* are generated by the IDL-compiler of CORBA, the compiler takes the file *MM.idl* as input. All the other files are coded by the programmer. The names of the two files *MM_i.h, MM_i.cc* are given by CORBA, the names for the files *client.cc, server.cc, MM.cc, MM.h* are chosen by the programmer. The file *MM.hh* has no direct relation to the files *MM.h* and *MM.cc*.

MM.h and *MM.cc* contain the effective math megamodule. *MM_i.h* and *MM_i.cc* contain a wrapper that wraps the effective math megamodule into the form required by the ORB.

Server.cc and *client.cc* each contain a main for starting the CORBA-server and CORBA-client.

3 Relationships between the files





The file ***MM.hh*** contains the abstract class *MMBOAImpl* (created by the IDL-compiler). This class declares the C++ mapping of all the services defined in the file *MM.idl*. The arguments are either native C++ types (e.g. char) or CORBA-types (e.g. CORBA::double).

The file ***MM_i.h*** defines the class *MM_i*, a subclass of *MMBOAImpl*. *MM_i* implements the abstract services of *MMBOAImpl*. Yet *MM_i* is only a wrapper. It creates an object of type *MServer* (defined in file *MM.h*) and dispatches all function calls to the functions of *MServer*. When dispatching, *MM_i* also performs the data-conversion from CORBA-types to native C++ types (e.g. from CORBA::double to double), because the class *MServer* is independent of CORBA and does not offer any CORBA-types in its interface.

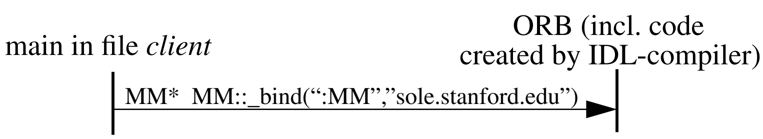
The file ***MM.h*** defines the class *MServer*. *MServer* provides the services of the math megamodule and is thus the core of the math megamodule. *MServer* could be used independant of the other files.

The file ***client.cc*** contains a main. The main uses the services provided by the math megamodule. The syntax is the one given by the class *MMBOAImpl* in file *MM.hh.* Before using the services, the main asks the ORB for the object-handle of the math megamodule (function *MM::_bind*, *MM* is also provided by *MM.hh)*.

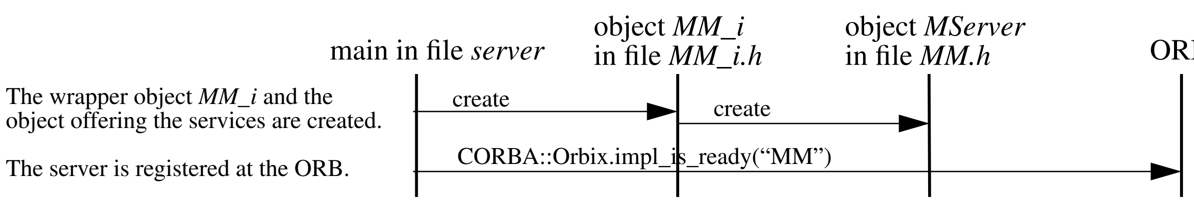
The file ***server.cc*** contains a main that creates the wrapper object *MM_i* and registers it at the ORB (function *CORBA::Orbix.impl_is_ready(“MM”))*.

4 Interactions

Starting the client:



Starting the server:



Calling a service (e.g. INVOKE):