

CONTRACT NUMBER 508830

**DEISA**  
DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR  
SUPERCOMPUTING APPLICATIONS

**European Community Sixth Framework Programme**

RESEARCH INFRASTRUCTURES  
Integrated Infrastructure Initiative

Grid-enabled simulation codes: status report of first release

Deliverable ID: DEISA-D-JRA6-1.1

**Due date: October 30<sup>th</sup>, 2004**

**Actual delivery date: November 18, 2004**

**Lead contractor for this deliverable: IDRIS - CNRS**

**Project start date: May 1<sup>st</sup>, 2004**

**Duration: 5 years**

<b>Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)</b>		
<b>Dissemination Level</b>		
<b>PU</b>	Public	X
<b>PP</b>	Restricted to other programme participants (including the Commission Services)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission Services)	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)	

## Table of Content

Table of Content .....	1
1. Introduction .....	2
1.1 Executive Summary .....	2
2. Coupled applications description .....	3
2.1 Terminology .....	3
2.2 Environmental Sciences .....	3
2.3 Combustion project .....	4
2.4 Astrophysical project .....	5
3. Installing CORBA .....	7
3.1 CORBA overview .....	7
3.2 CORBA Name Server .....	7
3.3 Installing CORBA .....	8
4. Migrating and deploying coupled applications .....	9
4.1 Interactive tests .....	9
4.2 Deploying applications with the Unicore coupled application plug-in .....	10
4.3 Future tests .....	10
5. Conclusion .....	11
6. References and Applicable Documents .....	12
7. Document Amendment Procedure .....	13
8. List of Acronyms and Abbreviations .....	14

## **1. Introduction**

### **1.1 Executive Summary**

The main objective of JRA6 is to facilitate scientific research projects based on coupling methodology using distributed objects software technology. Coupling codes deal with multi-physic or/and multi-scale simulations and are more especially well-suited for interdisciplinary projects in which several legacy codes have to cooperate.

In this document we present the status report of the first three coupled applications to be migrated to the DEISA infrastructure. First we describe briefly the three scientific projects, then we introduce the basic CORBA knowledge and, to conclude, we describe the migration status on the DEISA infrastructure.

This document is public.

## 2. Coupled applications description

Before we describe the three initial coupling projects, we are going to define terms which could be ambiguous in this document.

### 2.1 Terminology

A *module* (*CORBA module* or *component*) is one executable code which belongs to the whole coupled application and interacts with other modules, typically on remote systems.

An *application* (or *coupled application*) is the set of all modules needed to run the whole application. For example in the astrophysical project, *DCosmo3D* is the name of the coupled application which is composed of 3 modules: Gravitation, Chemistry and Hydrodynamic.

CORBA applications are client-server applications. A software *server* (or *CORBA server*) is a module which offers one or more CORBA computing service(s). Other CORBA modules which use this service are *clients*. Modules can assume the role of a *client* or/and a *server*. In the text below we will use the terms *server* or *client* to specify the primary role of a module.

An *invocation* designates when a CORBA-based application calls a remote service incarnated in another CORBA-based application. We will say also that an application *invokes* a CORBA service.

### 2.2 Environmental Sciences

<i>Title</i>	Hydrological cycles over West African continent (HYCYMAC)
<i>Scientific leader</i>	Michel Vauclin and Christophe Messenger, Laboratory LTHE (Laboratoire d'études des transferts en hydrologie et environnement) Grenoble, France.
<i>Partner Laboratories</i>	LGGE (Laboratoire de glaciologie et géophysique de l'environnement), Grenoble, France. HSM (Hydro sciences Montpellier), Montpellier, France. IRM (Institut Royal de Météorologie), Bruxelles, Belgium. LEGOS (Laboratoire d'étude en géophysique et océanographie spatiale), Toulouse, France. LEGI (Laboratoire des Ecoulements Géophysiques et Industriels), Grenoble, France. Lancaster University, Bailrigg, United Kingdom.
<i>Links with other scientific projects</i>	AMMA (African Monsoon Multidisciplinary Analysis) International project on West African monsoon

Mise en forme: Puces et numéros

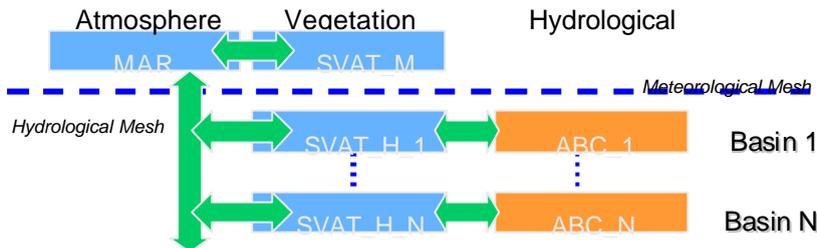
Mise en forme: Puces et numéros

The scientific goal of this project is to have a better understanding of the West Africa climate and more especially to study the water exchanges between water cycles from the atmosphere, the hydrological watersheds and the vegetation at thin scale. This is

typical an inter-disciplinary project in which coupling methodology has been adopted to avoid mixing several legacy codes.

Three kinds of codes had been coupled with CORBA:

- ? the atmospheric model RCM-MAR (Regional Climate Model),
- ? the vegetation/soil model called SISVAT (Sea Ice Soil Vegetation Atmosphere Transfer),
- ? and the hydrological model ABC.



**Figure 2: Description of the Atmosphere-Vegetation-Hydrology coupling.**

Two scales are considered in this application:

- ? the atmosphere grid on which the Atmosphere model and the SISVAT model compute,
- ? and the hydrological mesh on which a SISVAT describes at a smaller scale the basin studied and the hydrological model itself (see figure 2).

This application is running in production mode at IDRIS between a vector machine and a scalar one (the RCM-MAR and SISVAT codes are notably faster on vector platform whereas ABC code is faster on scalar machine like an IBM-SP4).

This application can run with different configurations and it is quite easy to add a new basin by adding a SISVAT and ABC modules. In coupling software layers point of view this means that the application is *extensible*.

### 2.3 Combustion project

<i>Title</i>	Large eddy simulations (LES) of turbulent combustion including pollutant species prediction and radiative heat transfers.
<i>Scientific leader</i>	Denis Veynante, Laboratory EM2C (Laboratoire d'énergie moléculaire et macroscopique), Chatenay Malabry, France.
<i>Partner Laboratories</i>	- CERFACS/IMFT (Institut de mécanique des fluides de Toulouse), Toulouse, France. - LMFN-CORIA (Laboratoire de mécanique des fluides numérique), Rouen, France.

The scientific aims of this project is to build a new numerical simulation code in order to develop practical systems like aeronautic engines, automotive engines, industrial furnaces, etc. to optimize them in terms of efficiency (fuel consumption and power) and to reduce pollutant emissions.

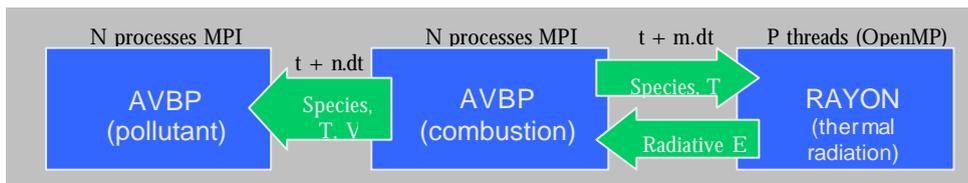
Three physical phenomena are computed in three different modules:

- ? the combustion (AVBP),
- ? the pollutant formation (AVBP),
- ? and the computation of the radiative source terms (RAYON).

AVBP is a CFD tool for numerical simulation of unsteady turbulence for reaction flows. AVBP, developed by the CERFACS, is parallelized with MPI. Two instances of AVBP are used: one to compute the combustion process and another one to compute the evolution of pollutant species. The code RAYON, specially developed by EM2C laboratory, determines the thermal radiations (see figure 1). RAYON is parallelized with OpenMP.

Because the physical processes involved in the 2 modules AVBP/Pollutant and RAYON are slower than the combustion process, the AVBP/Combustion doesn't need to invoke AVBP/Pollutant or Rayon at each iteration.

Various experiments showed that the pollutant module can be invoked from the combustion one every 50 iterations (this rate depend on physical case studied).



**Figure 1: Coupled application with the three modules pollutant, combustion and thermal radiation and their main data exchanges.**

Depending of the case studied, there are several possibilities to couple these three modules:

- ? pollutant/combustion,
- ? combustion/thermal radiations,
- ? pollutant/combustion/thermal radiations.

At present time, this coupled application is running in production mode at IDRIS.

## 2.4 Astrophysical project

<i>Title</i>	From origin of galaxies to stars.
<i>Scientific leader</i>	Jean-Michel Alimi, LUTH (Laboratoire de l'Univers et de ses Théories), Meudon, France
<i>Partner Laboratories</i>	Miguel Hernandez University, Elche, Spain. Astrophysical department in CEA (Commissariat à l'Energie Atomique), Saclay, France.
<i>Links with other scientific projects</i>	Horizon project: French astrophysical project which federates the research efforts of several laboratories in numerical simulations to understand the complex physical mechanisms

Mise en forme: Puces et numéros

Mise en forme: Puces et numéros

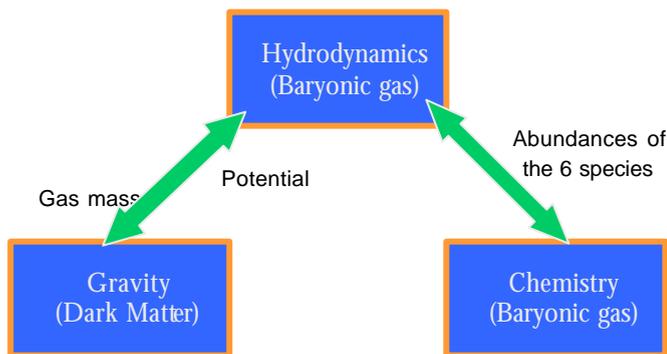
which organize the galaxy distribution.

The scientific aim of this project is to provide an efficient numeric code to study the galaxy and star formation. Because many physical processes are involved in this numeric simulation, the coupling approach had been adopted allowing us to choose the best platform for a given algorithm among heterogeneous grid. In future works, this will allow to change a module by another one (obviously with a different algorithm to simulate the same physical phenomena) and/or coupling a new physical process with the initial coupled application.

The coupled application is constituted of 3 modules:

- ? Gravity module computes the evolution of dark matter.
- ? Hydrodynamics module computes the baryonic gas evolution.
- ? Chemistry module treats of chemical reactions inside the baryonic gas (six main species are considerate).

A visualisation module can be added to the three other ones for demonstrations.



**Figure 3: Description of the 3 modules involved in astrophysical application and their main data exchanges.**

Gravitation and Hydrodynamics modules run notably faster on a vector platform whereas Chemistry runs faster on a scalar machine like an IBM-SP4. This implies that this coupled application will really take advantage of an heterogeneous Grid.

A new module describing the dark matter evolution is being implemented by the LUTH laboratory.

### 3. Installing CORBA

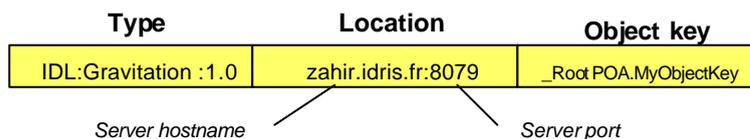
In this section we are going to describe briefly the CORBA framework, one useful CORBA utility which help us to start coupled applications and finally the CORBA installation.

#### 3.1 CORBA overview

CORBA (Common Object Request Broker Architecture) is an OMG's specification defined to allow computer applications to work together over networks. The CORBA middleware allows a CORBA-based program to interoperate with another remote one, what-ever are the hardware's, the operating systems, programming languages and the net-works.

CORBA provides all mechanisms to call a remote service from the caller (client) to the callee application (server). In CORBA terminology, a service can be an object method, a function or a subroutine (written in different languages) which can be remotely invoked by another CORBA-based application. This service is described (name, argument types and in/out/inout specifiers) in OMG IDL language (Interface Definition Language) in a file. Then a *projection* of this description is made by an IDL compiler to a particular language to be bound with user program. One part of this *projection* is for the caller program (*stubs*) and the other one for the callee program (*skeleton*). It is possible to project in languages such as C, C++, Ada, COBOL, Java, Lisp, Smalltalk or script languages as Python, but Fortran projection have never been specified by the OMG. Thus we chose a widespread projection language in CORBA implementations: the C++ language and Fortran legacy codes are encapsulated with the C++ *projections*.

At run time, the program which incarnates a CORBA service creates its reference, a kind of "universal pointer" called IOR (Inter Operable Reference). In particular, the hostname and the port on which the service is listening, are coded in the IOR (see figure 4).



**Figure 4: Inter Operable Reference (IOR) content.**

This "universal pointer" can be written in a file and send to the client or can be stored in the Name Server (CORBA utility which works as a remote IOR's database). Once the client part read this IOR or query the Name Server to obtain the service's IOR, it can access remotely to it.

#### 3.2 CORBA Name Server

The CORBA Name Server (NS) is one of the CORBA utilities specified by OMG which helps to solve the problem of getting IOR's services when a distributed application is launched. A service can be launched everywhere, so an application which needs this service doesn't know where it is located. The Name Server is a remote service from which applications can register their available services – the name of the service and its IOR are stored in it database – thus other applications which need a particular service can inquire the NS to get the IOR. Generally the user services can run on various

locations (various IOR's); at the contrary the Name Service must have a fixed IOR (fixed host and port) to be accessed by any CORBA application.

A functionality in the Name Server called the *NameContext* can be used to classify the IOR's. It works as hierarchical directory. We use this functionality to avoid the different services having the same name and possible conflicts if we run the same coupled application twice or more.

In the framework of our coupled applications, we use also the Name Service to synchronize all modules involved. More precisely, if a module inquires the Name Server for a remote service IOR which have not started yet, then the module will wait and will inquire again it until the service is registered in.

### **3.3 Installing CORBA**

As the CORBA implementation we had usually used, became a commercial product we selected the omniORB 4.0.3 implementation before migrating applications. This public implementation is known as a robust and fast one, have an active mailing list and have up to now regular updates. First we installed and tested it on our coupled applications at IDRIS after making minor changes. Then this ORB implementation had been installed at CINECA, Jülich and RZG (in 64 bits). Furthermore information about omniORB can be found in its web site[3].

## 4. Migrating and deploying coupled applications

### 4.1 Interactive tests

All these three applications have been migrated to the DEISA core sites: CINECA, IDRIS, FZJ and RZG. We ran each application interactively at each site without any problem (all modules of the applications ran on the same site). We are going to test inter-sites runs when all features (as multi-site LoadLeveler) need to submit jobs on the DEISA Infrastructure will be available.

For instance, here are the main steps, to run the astrophysical application:

```
## First we set up the CORBA environnement
$ . /usr/local/pub/omniORB/config/corba.env

## Start the Name Server on chosen port (here 9993):
omniNames -start -ORBEndPoint giop:tcp:zahir.idris.fr:9993 &

## Run the Chemistry module. We have to define
## - the context name (option -c).
## - the NameService location (-ORBInitRef NameService=...)
chemistry -c Coupling/Astro -ORBInitRef NameService= \
  corbaname::zahir.idris.fr:9993 &

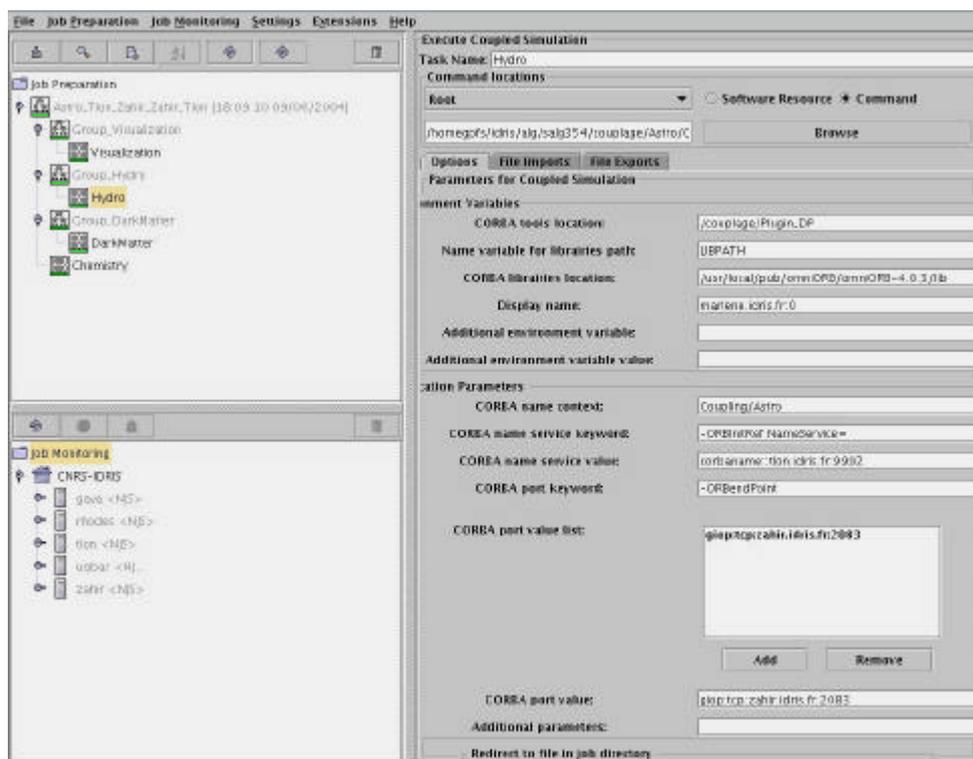
## Run the Gravitation module:
gravitation -c Coupling/Astro -ORBInitRef NameService= \
  corbaname::zahir.idris.fr:9993 &

## Run the Hydrodynamics module
hydrodynamics -c Coupling/Astro -ORBInitRef NameService= \
  corbaname::zahir.idris.fr:9993 &
```

## 4.2 Deploying applications with the Unicore coupled application plug-in

With Unicore [4], preparing and running of coupled applications is simplified using the coupled application plug-in developed previously by IDRIS in the Eurogrid project [5].

In particular the different environments variables (as the library path), CORBA parameters (as the Name Server location or the port number), name context, etc., will be set -up using default values for each target system at the different sites.



## 4.3 Future tests

To complete our code migrations, we have mainly to test these applications with the dedicated network DEISA. In this sub-network we wouldn't have any security problems to deploy the coupled application over several DEISA sites. To anticipate possible issues, small inter-site tests were done by tunnelling invocations under `ssh`. This secure channel is well-suited for general CORBA services like the Name Service but nevertheless it would be less adapted (for efficiency and practical reasons) between application modules.

## 5. Conclusion

After installing the chosen CORBA implementation on each site, we migrated the three coupled applications to each core site. All applications have been tested successfully inside a given site. To finish the migration step, we have to execute them in distributed mode them between several sites through the dedicated network.

At the application level, the next main steps consist of having a better load-balancing between coupled modules to avoid wasting CPU cycles when modules wait (to be requested or the end of an invocation). This work can be done:

- ? By making an analysis of data dependency between modules to point out asynchronous treatments
- ? By adjusting the number of processors for each module to load-balance the whole application.
- ? By providing a redistributing mechanism to manage the different data distribution between parallel modules.

## 6. References and Applicable Documents

- [1] <http://www.deisa.org>
- [2] <http://www.omg.org>
- [3] <http://omniorb.sourceforge.net/>
- [4] <http://unicore.sourceforge.net/>
- [5] <http://www.eurogrid.org/>

## **7. Document Amendment Procedure**

## 8. List of Acronyms and Abbreviations

<b>CFD</b>	Computational Fluid Dynamics.
<b>CORBA</b>	Common Object Request Broker Architecture is the OMG's specification which defines the middleware to allow remote computer applications to work together over networks.
<b>OMG</b>	The Object Management Group is a consortium that produces and maintains computer industry specifications for interoperable enterprise applications. The other well-known specifications are UML (Unified Modelling Language) and MDA (Model Driven Architecture).
<b>NC</b>	(Name Context) locates where an IOR is located in the Name Server database. NameContext's have a hierarchical structure like UNIX directories. Example: "DEISA/Astro/Run1/Gravity" identifies the IOR corresponding at the service "Gravity" in the NameContext "DEISA/Astro/Run1"
<b>NS</b>	(Name Server) is a CORBA utility which works as an IOR's database and which can be remotely inquired.
<b>IOR</b>	(Inter Operable Reference) is a reference on a CORBA service. This reference is a kind of "universal pointer" in which the hostname and the port on which the service is running are coded. Once created by the application which delivers this service, it can be used by any CORBA application which invokes it.