

The PRISM coupling and I/O System : Oasis4



Damien Déclat¹, René Redler², Hubert Ritzdorf²,
Thomas Schoenemeyer³, Sophie Valcke¹, Reiner Vogelsang⁴

Empowered by Innovation

1 CERFACS, France
2 NEC CCRLE, Germany

3 NEC HPCE, Germany
4 SGI Deutschland, Germany



Provide open source software package and model technical interface to manage: **Coupling of climate component models**
I/O of each individual component

- ✗ Programming language: **Fortran90** and **C**
- ✗ Use of freely available open source softwares, hidden from the model developer (**MPICH, LAM-MPI, NetCDF, libXML, mpp_io, SCRIP, ...**)
- ✗ Use of **optimized versions** for specific independent software components (e.g. MPI, NetCDF)

Oasis4 three main parts:

the Driver:

- Static launching of component models: all at beginning of simulation (MPMD/SPMD)
- Distribution of global information: initial date, calendar, coupling parameters
- Component model **monitoring**

the Transformer:

- Spatial interpolation:** nearest-neighbour, bilinear, bicubic, trilinear, *tricubic, conservative remapping, ...*
- Spatial transformations:** flux correction, merging
- General **algebraic operations**
- Combination of coupling data**
- On different types of fields: **1D, 2D, 3D, scalar or vector**
- On different types of grids: **regular, gaussian, stretched, reduced, unstructured**

the model interface library PSMILE:

- Insures **coupling data flow with MPI** between any component models, including **parallel repartitioning**
- Insures **data input/output** from/to files (**mpp_io**)
- Parallel 3D neighborhood search** based on multigrid algorithm for data interpolation
- time integration**
- Switch between coupling exchange and input/output is:
 - indicated by the user in a coupling configuration file (**XML**)
 - transparent for the component model
 - managed automatically by the PSMILE

A simple example: How to implement, define, compose and deploy a PRISM system ?

The component model interface PSMILE

The coupled run configuration

```

Program example
use PRISM

! Initialisation phase
call PRISM_init_comp ( component name, comp_id, ierror )
call PRISM_get_localcomm ( mpi_comm_local, comp_id, ierror )

! Grid definition
call PRISM_def_grid ( grid_id, grid_name, comp_id, shape, type, ierror )
call PRISM_set_offset ( grid_id, # blocks, offset, ierror )
call PRISM_set_corners ( grid_id, # corners, shape, "data", ierror )
call PRISM_set_scalefactor ( grid_id, shape, data, ierror )

! Grid information related to transient variables
call PRISM_set_points ( point_id, name, grid_id, shape, "data", ierror )
call PRISM_set_angle ( point_id, shape, data, ierror )
call PRISM_set_mask ( point_id, mask_id, shape, data, ierror )

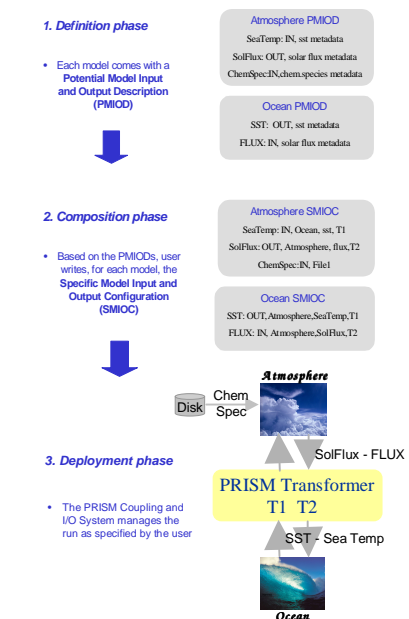
! Definition of transient variables
call PRISM_def_var ( var_id, var_name, point_id, mask_id, dimension, shape, type, ierror )

! Time step loop
do i = 1, no_timesteps
[...]
call PRISM_put ( var_id, date, date bounds, data, ierror )
call PRISM_get ( var_id, date, date bounds, data, ierror )
enddo

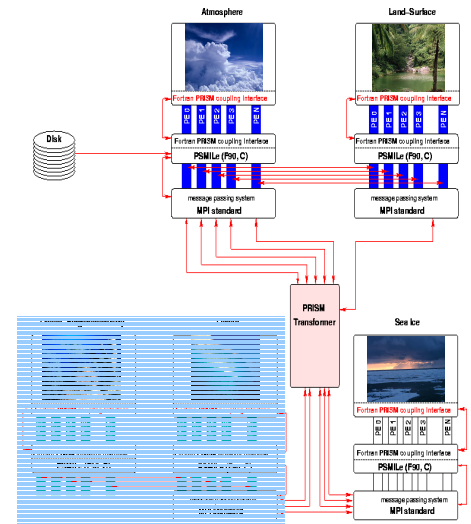
! Finalisation phase
call PRISM_terminate ( ierror )

end program example

```



The data flow



- directly between component models within the same executable
- directly between component models in separate executables
- through the transformer processes
- from/to a file from/to a component model

Current Status and Perspectives

Today: Oasis4 prototype version is available.

Perspectives:

- Oasis4 final PRISM version will be delivered in December 2004.
- Oasis4 developments will go on after December 2004 with PRISM sustained support.

<http://prism.enes.org>
PRISM Work Package 3A
Contact: valcke@cerfacs.fr