A review of the coupling software developed and used in the climate modelling community

S. Valcke - CERFACS

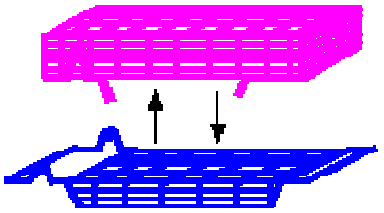
Outline

- Code coupling in climate modelling
- Different technical solutions to assemble model codes
- Some coupling software used in the climate community
  - OASIS coupler
  - Earth System Modeling Framework
  - Flexible Modeling System (GFDL)
  - Model Coupling Toolkit
  - Type DataTransfer library
  - Bespoke Framework Generator

Code coupling in climate modelling

Why couple ocean and atmosphere (and sea-ice and land and …) models?

- Of course, to treat the Earth System globally



What does "coupling of codes" imply?

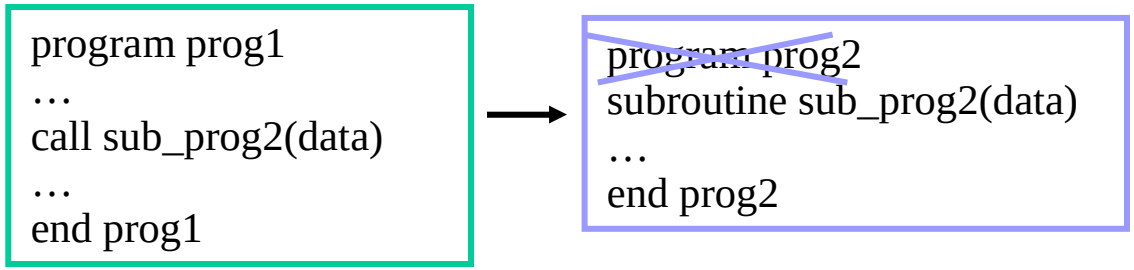
- Exchange and transform information at the code interface
- Manage the execution and synchronization of the codes

What are the constraints?

- ✓ Coupling should be **easy to implement**, **flexible**, **efficient**, **portable**
- ✓ **Coupling algorithm** dictated by **science** (sequential vs concurrent coupling
- ✓ Start from **existing** and **independently** developed codes
- ✓ Global **performance** and **load balancing** issues are crucial
- ✓ **Platform** characteristics (CPU size, message passing efficiency, …)
- ✓ Interaction with the **Operating System** must be considered

Different technical solutions to assemble model codes

1. merge the codes:



☺ efficient (memory exchange)

☺ portable

☹ not easy to implement with existing codes (splitting, conflicts in namespaces and I/O)

☹ not flexible (coupling algorithm hard coded)

☹ no use of generic transformations/interpolations

Different technical solutions to assemble model codes

2. use existing communication protocole (MPI, CORBA, UNIX pipe, files, …)

```
program prog1
…
call xxx_send (prog2, data, …)
end
```

```
program prog2
…
call xxx_recv (prog1, data)
end
```

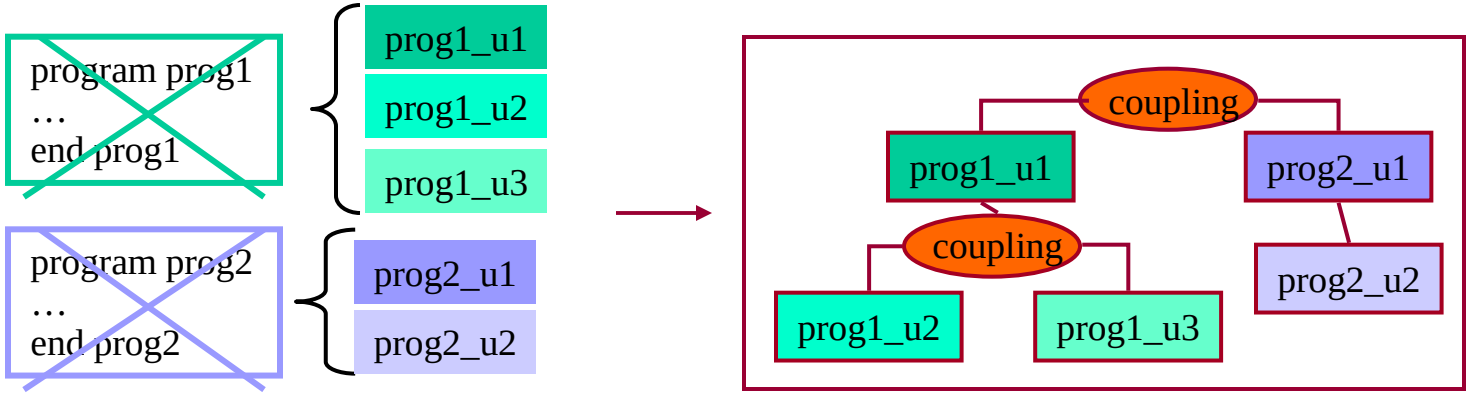
☺ existing codes
☺ concurrent components

☹ not easy to implement (need to be protocol expert)
☹ not flexible
☹ less adapted to sequential coupling (waste of resources)
☹ no use of generic transformations/interpolations
☹ efficient
😐 (portable)

Different technical solutions to assemble model codes

3. use coupling framework/library

- Split code into elemental units
- Write or use coupling units
- Use the library to build a hierarchical merged code
- Adapt code data structure and calling interface



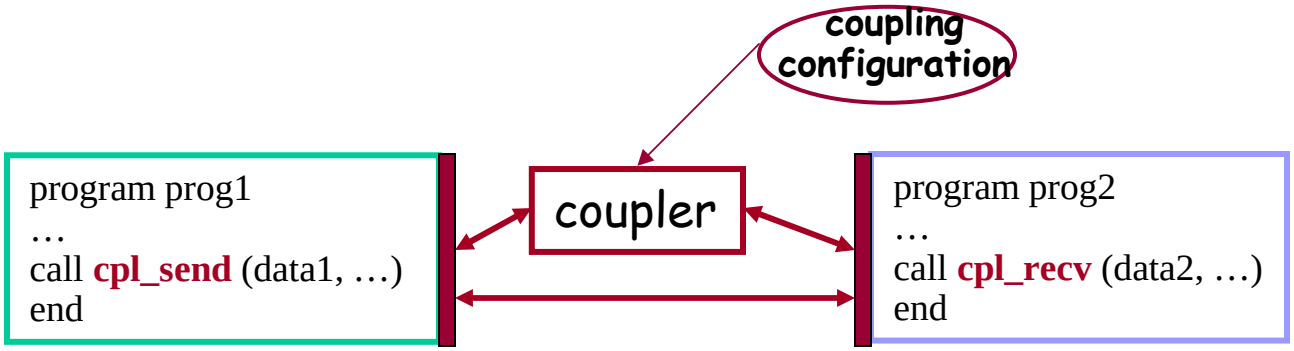
☺ flexible
☺ efficient
☺ portable
☺ use of generic utilities (parallelisation, regridding, time management, etc.)
☺ sequential and concurrent components

☹ existing codes
😐 (easy)

→ probably best solution in a controlled development environment

Different technical solutions to assemble model codes

4. use a coupler (OASIS, PALM, MPCCI …)



- ☺ existing codes
- ☺ flexible
- ☺ portable
- ☺ use of generic transformations/regridding
- ☺ concurrent coupling

- ☹ sequential coupling
- 😐 (efficient)

→ probably best solution to couple independently developed codes

Some coupling software used in the climate community

OASIS

- Separate coupler executable (regridding) + communication library:
  - ➢minimal level of interference in the component codes
- Developed by CERFACS since 1991, with NLE-IT and CNRS since 2001
- Written in F90 and C; open source license (LGPL)
- Public domain libraries: external:
  MPI1/2; NetCDF;libXML; included: GFDL mpp_io; LANL SCRIP
- Large community of users: ~25 climate modelling groups internationally

- Initialization: `call prism_init(...)`
- Grid definition: `call prism_write_grid (...)`
- Local partition definition: `call prism_def_partition (...)`
- Coupling field declaration: `call prism_def_var (...)`
- **Coupling field exchange**: in model time stepping loop

```
call prism_put (…, time, var_array. …)
call prism_get (…, time, var_array, …)
```

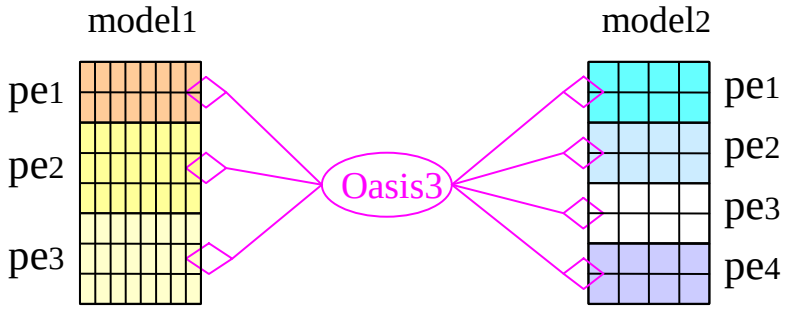
  - user's external configuration
    - for source or target (end-point communication)
    - for coupling frequency (effective sending or receiving)

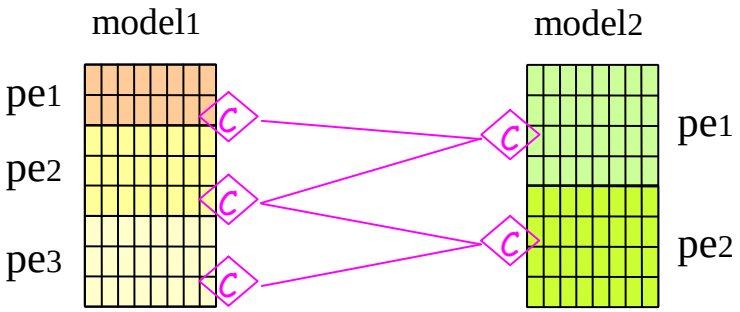
Some coupling software used in the climate community

Communication and data exchange with OASIS

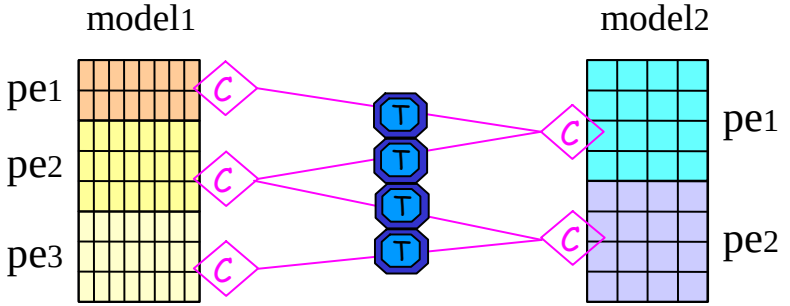
- separate sequential process
  - ✓neighbourhood search
  - ✓weight calculation
  - ✓interpolation per se during the run

OASIS3





OASIS4



Same grid, different decomposition
- direct repartitioning

Different grid and decomposition
- interpolation in parallel Transformer

- Parallel communication including repartitioning:
  - based on geographical description of the partitions
  - parallel calculation of communication patterns in source PSMILe
  - one-to-one, one-to-many
  - extraction of useful part of source field only

Some coupling software used in the climate community

User-defined external coupling configuration

- total run time, component models, number of coupling fields;
- for each exchange: coupling or I/O period, source and target, transformation/regridding

Field transformations/regridding

- on 2D scalar or vector fields: nearest-neighbour, linear, cubic, conservative regridding
- on 3D scalar fields: 3D nearest neighbour, trilinear interpolations (OASIS4)
- other spatial transformations: flux correction, merging, etc. (OASIS3)
- time accumulation, time averaging, general algebraic operations

➢on different types of grids: lat-lon, stretched or rotated (logically rectangular), gaussian reduced, unstructured (OASIS3 only)

Future plans

- continue to provide active user support for both OASIS3 and OASIS4
- within IS-ENES -> on-line services (documentation, tutorial, FAQs, forum. ...)
  -> OASIS4 development (parallel conservative remapping, global conserv., ...)
- collaborate with SCALES project (parallel global search for unstructured grids)

OASIS User meeting, May 25-26, Toulouse
On going call for applicant s for dedicated User support

Some coupling software used in the climate community

Earth System Modeling Framework

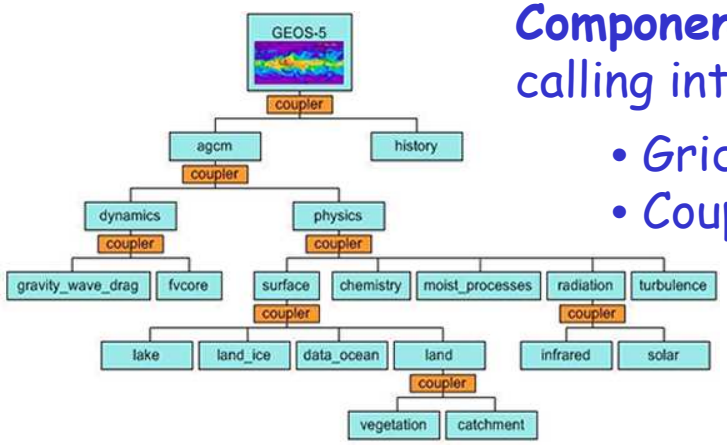
Open source **software** for **building** climate and weather modeling **components**, and **coupling** them into applications

- easier exchange modeling components amongst modeling centers

- Multi-agency governance (NASA, DoD, NSF, NOAA) with many partners (DoE, U. Michigan, MIT, UCLA, U. Maryland, COLA, CCA,…)
- Developments: Core team, Joint Specification Team, Change Review Board
- Technical:
  - Mainly written in C++, with full F90 interface and partial C/C++ interface
  - Runs on most high performance parallel computing platforms,
  - Free user support, 2500+ tests bundled with source distribution
- User community:
  - 70 ESMF components and applications in the community
  - GEOS-5 model at NASA Goddard Space Flight Center
  - Earth System of the Battlespace Environments Institute
  - New NWP system at the National Centers for Environmental Prediction

Some coupling software used in the climate community

Earth System Modeling Framework



Component-based design: component is a code with well-defined calling interface and a coherent function

- Gridded Components: scientific code
- Coupler Components: data transformation/transfer

➢ The user uses ESMF functions and methods to build a climate model as a hierarchy of nested components

ESMF "Superstructure"

1. Define Gridded Components : slip code into init, run and finalize methods
2. Wrap native data structures into ESMF data structure
3. Write Coupler Components
4. Register init, run and finalize methods to ESMF comp (in driver application)
5. Schedule components and exchange data
6. Execute the application

```
subroutine myOceanRun (.. , impState, expState, clock, …)
 type(ESMF_State)    :: impState
```

```
subroutine  oceanToAtmCpl (.. ,)
 call ESMF_FieldRedist(oceanField, atmField, …)
```

```
...
call ESMF_GridCompSetEntryPoint (oceanComp,
     ESMF_SETRUN, myOceanRun, …)
…
call ESMF_GridCompRun(oceanComp,  …)
call ESMF_CplCompRun (oceanToAtmCpl, …)
call ESMF_GridCompRun(atmComp,  …)
```

Some coupling software used in the climate community

Earth System Modeling Framework

ESMF "Infrastructure" functionality:

- calendar management; message logging
- linear and higher order interpolation in up to 3D (ESMF_FieldRegrid function)
- extension to multiple executable coupling
- in-place coupling from within a Gridded Component (not only through the interface)

Future plans

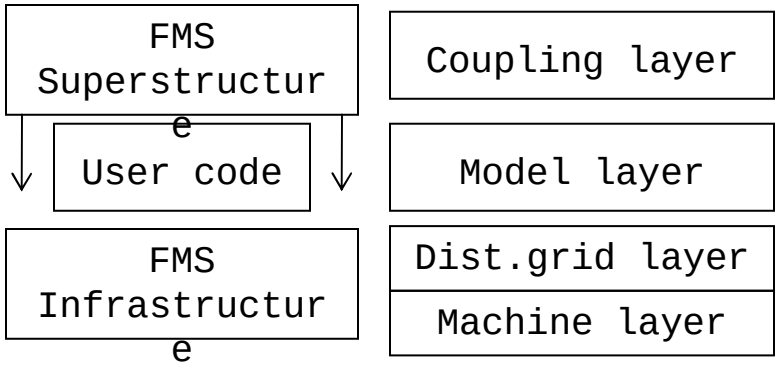
- conservative interpolation; handling of masking and fractional areas
- ESMF chosen in 2008 for the National Unified Operational Prediction Capability (NUOPC - consortium of U.S. operational weather and climate centers)
  - ESMF-based, managed, multi-model ensemble in the 2015 time frame.
- New objective: web-based coupling services for new customers e.g. hydrology, space weather (ease of configuration, ease of use, looser coupling, heterogeneous coupling, language interoperability)
- Metadata standardisation for components, simulations, and output datasets (CURATOR project in interaction with the Earth System Grid (ESG) data distribution portal and the EU Metafor project).

Some coupling software used in the climate community

The Flexible Modeling System (FMS)

Modeling framework for the construction of coupled climate models

- Developed since more than 10 years at/for GFDL
- Written in F90
- FMS "Infrastructure" layer:
  - I/O, exception handling,
  - Operations on distributed gridded fields (expressed independently of the underlying platform)



- FMS "Superstructure" layer:
  - Domain-specific coupling layer for ESM
  - Single executable with serial or concurrent exchange of components
  - Includes data assimilation (parallel ensemble adjustment Kalman filter)
- Computational efficiency: FMS shown to be scalable with up to O(1000) pes

Some coupling software used in the climate community

The Flexible Modeling System (FMS)

FMS "Superstructure" layer:

- Only components considered with one specific "slot" for each (unlike ESMF)
  - atmosphere, ocean surface, land surface, and ocean ("stubs" , i.e. no component, or "data" from dataset possible)
- Hard-coded lists of exchange fields between components
- Regridding, redistribution, or direct exchanges between components
- Constraints:
  - Interface fluxes must be globally conserved
    - ➢ *exchange grid* :set of cells defined by the union of all vertices of source and target grids
    - ➢ ocean and atmosphere sea-land masks adjusted to remove mistmatch
  - No limitation on the individual component timestepping
  - No restriction on the discretization of a component
  - Exchanges consistent with physical processes occurring near the surface
    - ➢ Implicit calculation of vertical diffusive fluxes over the whole column
    - ➢ Up-down sweep for tridiagonal matrix resolution is done going through the exchange grid where flux are calculated

Some coupling software used in the climate community

The Model Coupling Toolkit

Library and related datatypes for **creating parallel couplers** and **parallel coupled models** (not a ready-to-use coupler)

- Developed by the U.S. DoE within SciDAC program.
- Written in F90; includes C++ and Python interface
- Supported platforms: IBM, SGI, Compaq, NEC, Cray X1 and Linux
- Users: CCSM3 (Community Climate System Model – NCAR) and CCSM4 couplers
  WRF/ROMS Hurricane model (NCAR, NOAA)
  Coupled ocean-atmosphere model at Oregon State University
- MCT library provides **functions** for:
  - component model registry and domain decomposition description (incl. unstructured grids)
  - **time averaging** and **accumulation**; spatial integration; merging of source fields
  - **parallel communication**
    - for **single** or **multiple** executable systems, with **sequential** or **concurrent** execution
    - **in-place coupling** from within a component, or coupling at the component **interface**
    - data **redistribution** (MCT_Send/Recv on disjoint sets of pes, REARRANGER – memory copy- for same sets of pes)
    - no calculation of interpolation weights but **sparse matrix-vector multiply**

Some coupling software used in the climate community

TDT – A library for Type Data Transfer

A library to **transfer data** between programs in a **platform** and **programming language independent** way (heterogeneous coupling)

- Developed by the Potsdam Institute for Climate (PIK)
- Core functionality written in C; usable in C, Fortran, Python and Java programs
- Transfer of **primitive data types** (integer, float, char, …) and **complex data structure**
- **Different transfer protocols** supported: Unix Socket, I/O from/to files, MPI
- Data and transfer protocol are **externally described** in XML configuration files
  - ➢ Change in data type or size or change in protocol is transparent for the code
- Limitations:
  - no check of data coherence
  - no data transformation (beside endianness)
  - manual synchronization of read and write

➢ Adapted for rapid-prototyping of coupled systems

- include TDT header file
- declare variables required by TDT
- read the configuration file
- open the communication channel
- perform the read or write
- close the channel

Future plans: active maintenance and user support; no major new features

Some coupling software used in the climate community

The Bespoke Framework Generator

A **generator of wrapper code** around components to build a coupled model with a **chosen coupling technology** based on **external metadata** description

- Developed by the U. Manchester
- Generative Programming Approach:
  - meta-language for user description of the coupled system (XML) following the DCD (Describe, Compose, Deploy) paradigm
  - code generator (XSLT) that generates wrapping code around the components to build the coupled model based on the metadata description
- Target coupling technologies:
  - BFG1: shared memory buffers, MPI, TDT, OASIS3, Globus MPI, Web Services
  - BFG2: MPI, OASIS4, ESMF, (+ argument passing)
- Communication and control
  - in-place calls (`call put(data,tag)`, `call get(data, tag)`)
  - argument passing at the component interface (BFG2 only)
  - component models can be called from within arbitrary loops
- Users: Tyndall Centre (UK), Grid ENabled Integrated Earth system (GENIE)
- Future plans: development in the framework of IS-ENES and METAFOR EU projects

Conclusions

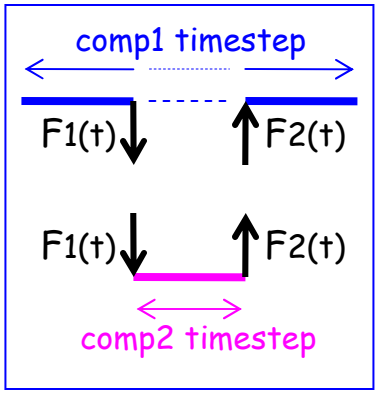
- Different approaches to component coupling:
  - Coupling library/functions (e.g. ESMF, MCT, FMS) to build coupled system using components
    - best solution in a controlled development environment
  - External coupler and communication library (e.g. OASIS):
    - best solution to couple independently developed codes

- The "best" coupling tool does not uniquely exist; it depends on:
  - computing environment
  - required utilities
  - ready to change/adapt your original codes
  - efficiency you want to achieve
  - …

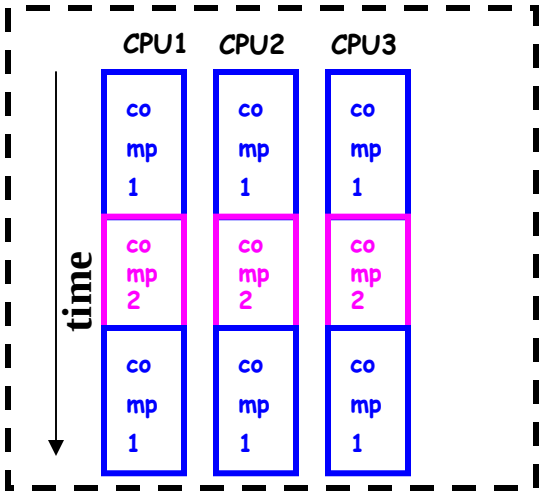
The end

Constraints of the coupling algorithm

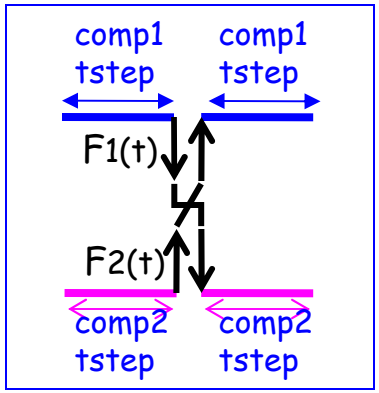
Sequential coupling :



=> sequential execution on the same set of cpus



Concurrent coupling:



=> concurrent execution on different sets of cpus