White paper on a strategy for full convergence of I/O and coupling tools
Deliverable D2.5

CECI, Université de Toulouse, CNRS, CERFACS - TR-CMGC-18-92

About this document

Other contributing authors:
Yann Meurdesoif (IPSL), Marie-Pierre Moine (Cerfacs)

Internal reviewers:
Mick Carter (Met Office), Kim Serradell (BSC)

Index

1. Abstract /publishable summary

It has been recognised for a while that the main functions of I/O servers (like XIOS) and couplers (like OASIS), i.e. communication and transformation of data, have much in common. The question of "convergence" between XIOS and OASIS therefore naturally arises. In this document, we perform a technical analysis of the similarities and differences between XIOS and OASIS regarding their general architecture, specific functionalities, API, configuration and development resources. We then analyse advantages and disadvantages of such a convergence and propose a plan regarding this strategic question.

The analysis leads to the conclusions that 1) XIOS already contains most of the functionality required to act as a coupler and 2) that merging the two tools would not be a practical way forward as the effort of merging would be higher than the effort of extending XIOS. But at this point, the authors also state that it is too early to decide if we should encourage the climate modelling community to switch to XIOS for coupling as this functionality in XIOS is not fully developed, tested and validated yet. Only if the proposed steps, including more development and testing, are successfully achieved, we would then be confident that XIOS can be considered as a community coupler.

2. Conclusion & Results

The technical analysis we report here leads to the conclusion that XIOS has a lot of potential to evolve from an IO server to become also a coupler. It also concludes that combining the two pieces of software with their current functions would not be a practical way forward. Therefore, the convergence of XIOS and OASIS, if it happens, would in practice imply adding few missing transformations in XIOS, extending its communication so to support coupling between two components, and proposing this evolved tool as a unique tool for IO and coupling. A very positive aspect of the convergence would be to avoid duplication of development efforts and to gather the current OASIS and XIOS human resources around one single tool. The transition should not be, in principle, to difficult for the users, as the two tools share globally a similar philosophy and interface: at each time step, models expose their data through a minimalist set of embedded calls, and the workflow applied to the data (transformation, remapping, communication) is configured externally by the user and can be modified without recompiling.

However, as coupling functionality in XIOS is not fully developed and has therefore not been tested and validated yet, it is obviously too early to decide if we should encourage the climate modelling community to switch to XIOS for coupling. In particular, a full validation of XIOS coupling functionality and workflow performance has to be completed. Only if the proposed steps, including more development and testing, were successfully achieved, we would then be confident that XIOS can be considered as a community coupler and encourage the climate modelling community in Europe to use it as such.

3. Project objectives

Given the importance of the coupler and of the IO server tools in the efficient execution of numerical simulations based on Earth System Models (ESMs), this deliverable contributes directly and indirectly to the achievement of a vast majority of the macro-objectives and specific goals indicated in section 1.1 of the Description of the Action:

Macro-objectives	Contribution of this deliverable?
Improve the efficiency and productivity of numerical weather and climate simulation on high-performance computing platforms	Yes
Support the end-to-end workflow of global Earth system modelling for weather and climate simulation in high performance computing environments	Yes
The European weather and climate science community will drive the governance structure that defines the services to be provided by ESiWACE	Yes
Foster the interaction between industry and the weather and climate community on the exploitation of high-end computing systems, application codes and services.	Yes
Increase competitiveness and growth of the European HPC industry	No

Specific goals in the workplan	Contribution of this deliverable?
Provide **services** to the user community that will impact beyond the lifetime of the project.	Yes
Improve **scalability** and shorten the time-to-solution for climate and operational weather forecasts at increased resolution and complexity to be run on future extreme-scale HPC systems.	Yes
Foster **usability** of the available tools, software, computing and data handling infrastructures.	Yes
Pursue **exploitability** of climate and weather model results.	Yes
Establish governance of common software management to avoid unnecessary and redundant development and to deliver the best available solutions to the user community.	Yes
Provide **open access** to research results and **open source** software at international level.	Yes
Exploit **synergies** with other relevant activities and projects and also with the global weather and climate community	Yes

4. Detailed report on the deliverable

4.1. Introduction

It has been recognised for a while that the main functions of I/O servers and couplers, i.e. communication and transformation of data, have much in common. With about 45 user groups in the European climate modelling community and around the world, the OASIS software, developed by Cerfacs and CNRS in France, is widely used for model coupling (Craig et al, 2017; Valcke et al, 2018). More recently, the number of users of the XIOS I/O server (http://forge.ipsl.jussieu.fr/ioserver) developed by IPSL also in France is steadily increasing, thanks to its use in the NEMO ocean community model. XIOS is, in particular, used in the French coupled models IPSLCM6 and CNRM-CM6, where it ensures the direct production of CF-NetCDF data files satisfying the CMOR3 format and automatically answering the CMIP6 Data Request. First steps toward the implementation of XIOS in IFS, the atmospheric component of EC-Earth, which is an Earth System Model developed by a European consortium gathering 27 research institutes from 10 countries in Europe, have also been taken. Within ESiWACE, IPSL plans to further develop XIOS to allow it to be used also as a coupler. Furthermore, an overview of the two tools leads to the conclusion that OASIS and XIOS share a similar software architecture: at each time step, models expose their data through a minimalist set of embedded API calls, and the workflow applied to the data (transformation, remapping, communication) is configured externally by the user using control files and can be modified without recompiling. The question of "convergence" between XIOS and OASIS therefore naturally arises.

In this document, we perform a technical analysis of the similarities and differences between XIOS and OASIS regarding their general architecture (section 4.2), specific functionality (section 4.3), API (section 4.4), configuration (section 4.5) and user support (section 4.6). We then analyse advantages and disadvantages of such a convergence (section 4.7) and propose a plan regarding this strategic question (section 4.8).

4.2. General architecture

OASIS architecture

The OASIS[1] coupler, OASIS3-MCT in its latest version, is a portable set of Fortran and C routines released under the terms of a Lesser GNU General Public License (LGPL). After compilation, OASIS is a coupling library that needs to be linked to the component models, and which main functions are to interpolate and exchange coupling fields between the components to form a coupled system. Low-intrusiveness, lightness and flexibility are OASIS key design concepts. OASIS uses the Model Coupling Toolkit (MCT) from Argonne National Laboratory (Larson et al, 2005) as a lower layer to perform in parallel all transformations including remapping on the source or target component processes and all coupling exchanges between the component processes via Message Passing Interface (MPI). Exchange fields, frequency and remapping are externally defined through a single formatted text file (the *namcouple*).

---

[1] This document analyses OASIS3-MCT, the latest version of the coupler, but for sake of simplification we refer to it with the general name "OASIS" throughout the document.
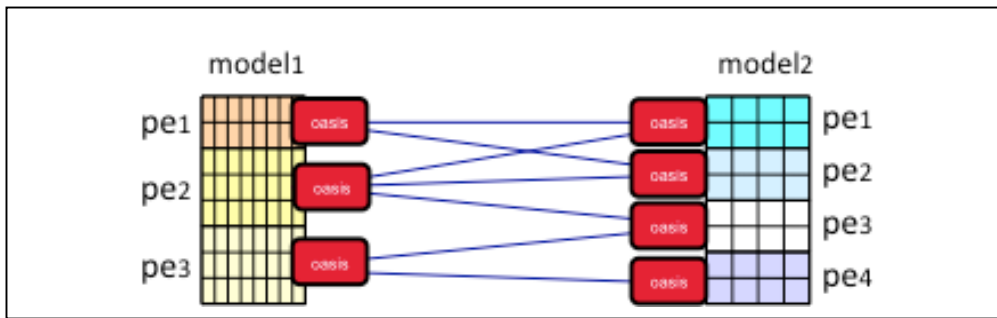
Figure 1 – OASIS general architecture

XIOS architecture

The XIOS IO server is written in C++ and is released under an Open Source CECILL Licence. XIOS is composed of a client library that needs to be linked to the model and either one or two levels of servers.
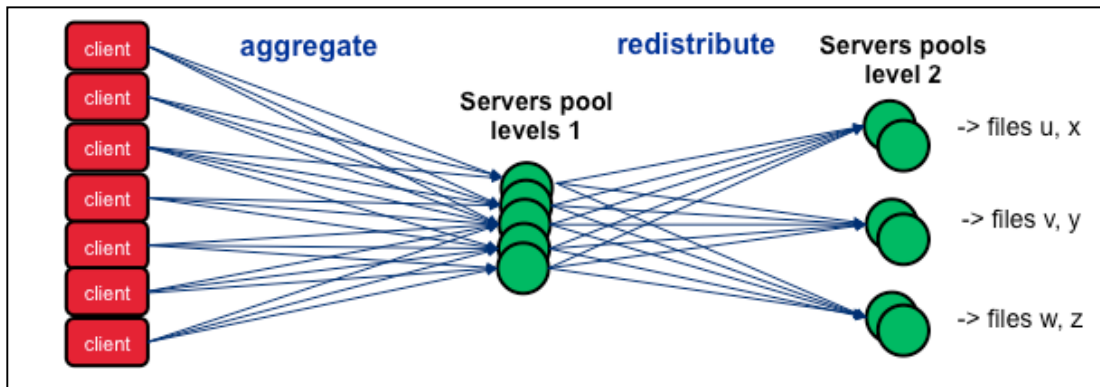

Fig 2: XIOS general architecture

The client part of XIOS manages the fields transferred through the XIOS API as output (xios_send_field) or input (xios_recv_field). It can transform and process a field either before sending it to the level1 server (output), or after receiving it from the servers (input). The series of transformation applied to a field by the clients is called the XIOS workflow. For each field, the workflow is not hard-coded in the model but configured externally by the user in one (or more) XML files.

Initially, the first level of servers was introduced so that the writing/reading to/from disk can be done in parallel by additional processes and therefore does not slow down the model processes. A redistribution of the data is performed between the clients and the level-1 servers.

More recently, a 2nd level of servers was introduced so to allow a second redistribution of the data and to ensure that each disk file is accessed only by one server writing a whole field. The motivation was to increase the performance by avoiding problems linked to parallel writing, i.e. the high cost of opening and closing files and the inability to compress NetCDF data "on the fly" (this limitation should be removed in future HDF5 1.10 versions).

The design of the servers changed with the introduction of the second level, achieving an important step toward coupling support. Indeed, the XIOS servers, instead of just writing the data to disk in parallel, now get the full grid definition (mesh and MPI distribution) and when they receive data, they can re-enter the XIOS workflow, i.e. apply transformations to the data. Therefore, the communication of data between the clients and the level-1 servers or between level-1 and level-2 servers now involves a reorganisation of data between a parallel source distribution and a different parallel target distribution with possibly data transformation on both sides. The communication of data in XIOS is therefore very

much alike the communication of data between two parallel components in a coupled model. The implementation of the full coupling functionality has not yet been completed, as other developments related to CMIP6 imperatives were given a higher priority, but this major evolution is still planned before the end of ESiWACE.

4.3. Specific functions

We compare here the specific functions of OASIS and XIOS in term of field transformations and other diverse aspects. First, Table 1 compares arithmetic, temporal and spatial transformations available in the two tools.

	OASIS	XIOS
2D spatial remapping	• 2D 1st and 2nd conservative remapping • 2D nearest-neighbour • 2D bilinear • 2D bicubic • Pre-defined set of weights	• 2D 1st and 2nd conservative remapping
Vertical interpolation	None	• Polynomial interpolation of any order • Interpolation from any vertical coordinate system onto pressure levels[1]
Other spatial transformation	None	• extraction of a subdomain or part of an axis (zoom) • reduction: domain->axis (e.g. zonal mean) , domain-> scalar, axis->scalar • transects • vertical or global sum • area or height normalisation • point station extraction
Time transformation[2]	• instant • accumulation, averaging • minimum, maximum	• instant, • accumulation, averaging, • minimum, maximum • only once during the run • diurnal cycle • seasonal cycle
Other transformations	• unit rescaling • forced global conservation	• unit rescaling
Field combination	No	Yes (arithmetic operation performed on each point)
Other		• Eliassen Palm flux calculation • connectivity of unstructured grid • expansion of local domain at first neighbour
[1] For vertical coordinate systems that evolves during the run, like sigma or hybrid system, the vertical interpolation weights are of course recalculated on-line during the run [2] With OASIS, these time transformations are necessarily performed on the field provided at each time step, whereas with XIOS it is possible to specify a lower frequency		

Table 1: comparison of arithmetic, time integration and spatial transformations available in OASIS and XIOS

The algorithms of the conservative remapping in OASIS and in XIOS are very different. OASIS uses the SCRIP library (see https://github.com/SCRIP-Project/SCRIP) calculating the mesh intersections in the spherical coordinate system (which causes some specific problems near the poles), while XIOS bases its calculations on a geometrically exact, local and explicit algorithm. More details can be found in [Jones 1999] and in [Kritsikis et al, 2017] for OASIS and XIOS respectively.

To be complete, we specify here that while XIOS explicitly supports 1D, 2D[2] and 3D fields, OASIS supports only 1D and 2D fields. Also, both XIOS and OASIS support regular latitude-longitude, logically rectangular, unstructured and Gaussian-reduced grids. In addition, XIOS supports grids of any dimension combining any of these 2D horizontal layer and vertical axis.

Regarding other functionality, we note that both OASIS and XIOS support masked grids, and XIOS also recognises missing values.

XIOS manages a real calendar (Gregorian, Julian, 360-day, fixed 365-day, fixed 366-day, user-defined) whose type can be specified in the configuration XML file(s) or through the API. This is not the case in OASIS where the time is expressed in seconds since the beginning of the run (or any other arbitrary units as long as they are the same in the different models and in the *namcouple*).

Finally, regarding the output format, it is interesting to note that XIOS can produce fully CF1.7 compliant NetCDF files if the user provides all required attributes (through the API or in the XML configuration file).

4.4.   API

The API for XIOS and OASIS are very much alike. They both include an initialisation phase into which MPI communicators, grid cells and decomposition, and coupling or IO fields are defined. In the time step loop, both APIs propose simple calls to send or to receive the fields. Finally, both APIs include similar termination calls.

One important difference however is that with XIOS, all the definitions can be either done through the API or in the XML configuration file (see 4.5.2). In practice, it is recommended to use the XML file to provide all definitions but for the description of the grid cells and parallel distribution, as these will be specific to each MPI process.

In the following paragraphs, we provide a brief comparative overview of XIOS and OASIS main API routines, without describing all arguments, so to illustrate the two APIs similarity.

Coupling initialisation

For OASIS, the coupling initialisation involves the two following calls:

- `Oasis_init_comp (compID, "compname", ierror)`

  must be called by all tasks of each component with same "compname" to establish a local communicator;

- `Oasis_get_localcomm (localcomm, ierror )`

  returns the value of a local communicator gathering the component tasks.

For XIOS, the equivalent calls are

- `xios_initialize("client", comm)`

---

[2] Here 2D means 2D on the sphere

returns a communicator gathering the client processes ;

- `xios_context_initialize("hello_word",comm)`

returns a communicator associated to all tasks involved in a specific context (which can be seen as the "component" counterpart of OASIS; e.g. ocean or atmosphere)

Partition definition

Climate component models normally run over many cores using a domain decomposition into individual partitions, each one using a specific MPI task. With OASIS, all processes exchanging coupling data have to express the local partitioning of the different grids in a global index space covering the whole grid:

- `oasis_def_partition (partID, ig_paral, ierror)`

with `ig_paral` being a vector of integers describing the local partition in the global index space.

The equivalent call for XIOS is

- `xios_set_domain_attr("2Ddomain",ibegin=ib,ni=ni,jbegin=jbeg, nj=nj)`

where `ibeg, jbeg, ni, nj` are integers describing the position of the local partition in a global index space. (As noted above, it is possible with XIOS to define the local partition using the XML file, but this is not recommended.)

Grid definition

With OASIS, the field grids must be defined in NetCDF grid data files grids.nc, masks.nc and areas.nc. These can be created by the user before the run or by the models in the initialisation phase with the following calls:

- `oasis_write_grid ("gridname", nx, ny, lon, lat, partID)`
- `oasis_write_corner ("gridname", nx, ny, nc, clon, clat, partID)`
- `oasis_write_mask ("gridname", nx, ny, mask, partID)`
- `oasis_write_area ("gridname", nx, ny, area, partID)`

where `lon, lat, clon, clat, mask` and `area` are arrays containing respectively the grid cell centre longitude and latitude, the grid cell corner longitude and latitude, the mask and the grid cell surface for the local partition.

The equivalent call for XIOS, for example for defining the grid cell centre longitudes and latitudes, is

- `xios_set_domain_attr("2Ddomain",lonvalue_2d=lon, latvalue_2d=lat)`

where `ibeg, jbeg, ni, nj` are integers describing the position of the local partition in a global index space. The same function can be used to define the corner longitudes and latitudes (attributes `bounds_lon_2d` and `bounds_lat_2d`), and the mask (attribute `mask`). The grid cell surface, however, is not an attribute used in XIOS for now, as the global conservation operation for which it is needed is not available (see Table 1 above). (Again, as explained above, it is possible with XIOS to define the field grids using the XML file, but this is not recommended.)

Coupling field declaration

With OASIS, it is mandatory for each component to declare the coupling fields it will send or receive with a call to:

- `oasis_def_var (varID, "fieldname", partID, dims, kinout, shape, type, ierror)`

  where `"fieldname"` is a symbolic name for the field used in the *namcouple* configuration file (see 4.5.1)

With XIOS, there is no need to declare the fields through the API as it can be conveniently done in the XML master file (see 4.5.2). For example, a field with its name, unit, and a reference to its grid (defined somewhere else in the XML file) can be declared with

```
<field id="temp" name="temperature" unit="K" grid_ref="grid_3d"/>
```

Sending and receiving fields

With OASIS, each process sends or receives its part of the coupling field by calling, within the time step loop:
- `oasis_put (varID, date, fldou, info)`
- `oasis_get (varID, date, fldin, info)`

  where `varID` is the field ID returned by the corresponding `oasis_def_var`, `date` is the time (in seconds or other arbitrary time units, see 4..), and `fldou` or `fldin` are the arrays containing the field to send or to receive, respectively.

With XIOS, the principle is the same; each process sends or receives its part of I/O field by calling within the model time step:
- `xios_send_field("field_id",fldou)`
- `xios_recv_field("field_id",fldin)`

  where `field_id` is the id of the field as declared in the XML file and `fldou` or `fldin` are the arrays containing the field to send or to receive, respectively.

  These calls do not contain any reference to the date at which the send or receive is performed; the date needs to be updated before the calls with:
- `xios_update_calendar(ts)`

  with ts being the current timestep, whose length is predefined with a call to:
- `xios_set_timestep(dtime)`

In both APIs, there is no reference to the target (for the send call) or to the source (for the receive call) or to the transformations to apply to the field. The source and the target (i.e. a specific file for XIOS, or another component model for OASIS), and the transformations are defined in the external configuration files. In both cases, the send or receive routines analyse the time or date at which they are called and appropriate actions are performed by the XIOS or OASIS library as configured externally by the user for that particular run.

Termination

With OASIS, the finalisation routine to call is:
- `oasis_terminate(ierror)`

With XIOS, two finalisation subroutines, one general and one for the context have to be called:
- `xios_context_finalize`
- `xios_finalize`

Although we have not exposed here all details of the APIs, the authors can confirm, based on this analysis and on their knowledge of the two tools, that the general architecture of OASIS and XIOS APIs

are very similar, with in-place calls for the initialisation, some definitions, and then, most importantly, send and receive minimal calls implemented in the time step loop and configured by the user externally for each specific run. Therefore it should not be too hard for a user of one tool to switch to the other tool.

4.5. Configuration

A particular coupling or I/O configuration with OASIS or XIOS is not hard-coded in the component models. Interfacing a model with OASIS or XIOS APIs offers the potential to realise coupling exchanges or IO but their activation needs to be specified externally by the user for each specific run. The OASIS or XIOS library attached to the model processes will then, during the simulation, act according to this external configuration. This implies in particular that no new modification of the code is needed in the model when setting up different couplings or different IOs with that model.

OASIS configuration

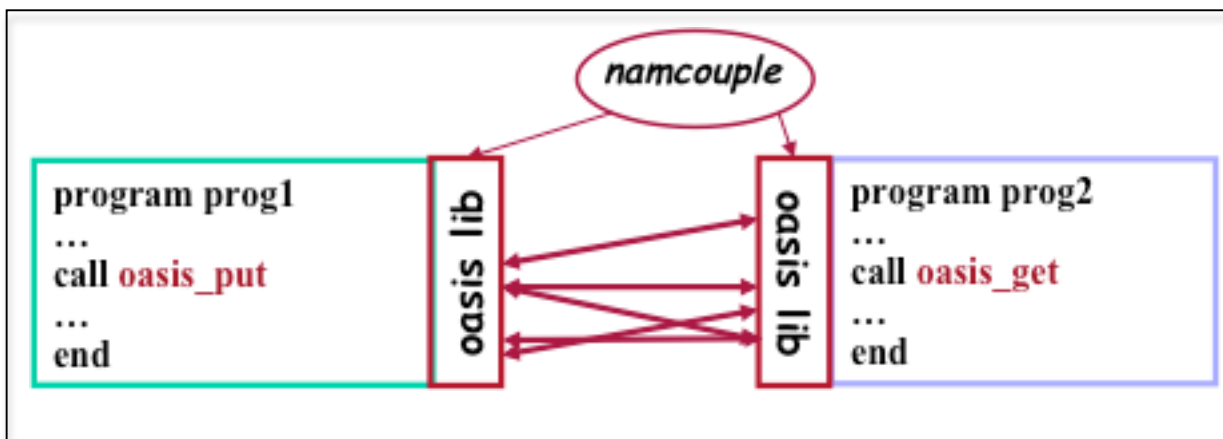For OASIS, the coupling exchanges are configured through the *namcouple* text file.

Figure 3 – Schematic of OASIS configuration

The first part of the namcouple is devoted to configuration of general parameters such as the total number of coupling fields, the total run time, the desired debug level, and other general options. The second part gathers specific information on each coupling field:

- the symbolic name of the coupling field in the source and in the target models (see `oasis_def_var` in 4.4); the association of these two symbolic names in the *namcouple* defines a coupling exchange

- the coupling period (in seconds, or any other arbitrary units as long as they are the same than the ones used in the models)

- different details related to the coupling field, e.g. the acronym of its source and target grids (see `gridname` in 4.4), the name of its coupling restart file, etc.

- the list of transformations and/or remapping to be performed on this field and related parameters, in OASIS-specific order (i.e. time transformation, pre-processing, remapping, post-processing)

Figure 4 shows an extract of an OASIS *namcouple* for a coupling field named `CONSFTOT` and `SOHEFLDO` in the source and target models respectively, coupled every 86400 seconds, associated to a

source grid and to a target grid with acronym `atmo` and `toce` respectively. 4 transformations will be performed on that field and an additional line of specification is provided for each: first a time transformation (`LOCTRANS`) specifying that the field will be accumulated for all time steps over the coupling periods (`ACCUMUL`); then a `CHECKIN` operation indicating that some statistics on the source field will be registered in the OASIS log file; then a `SCRIPR` remapping involving a bilinear algorithm (`BILINEAR`); and finally a `CHECKOUT` operation similar to the `CHECKIN` but on the target field after remapping.

```
# Field 2
CONSFTOT   SOHEFLDO   6      86400                4      flxat.nc    EXPORTED
atmo        toce  LAG=+14400        SEQ=+2
P    0     P     2
LOCTRANS  CHECKIN  SCRIPR  CHECKOUT
#
ACCUMUL
INT=1
BILINEAR  LR  SCALAR  LATLON  1
INT=1
```

Figure 4 – Example of a part of a *namcouple* configuration file for OASIS

At runtime the *namcouple* is read by the OASIS library and appropriate actions are performed accordingly when the model calls an `oasis_put` to send a coupling field or an `oasis_get` to receive one (see 4.4).

XIOS configuration

For XIOS, the configuration is done through the iodef.xml master file and possibly additional XML files sourced in iodef.xml.
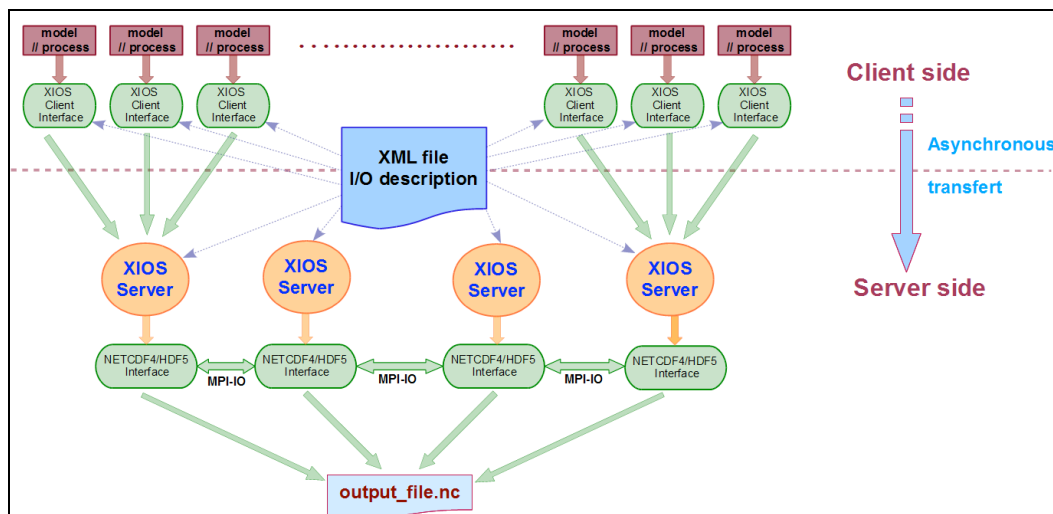

Figure 5 – Schematic of XIOS configuration

These files contain elements grouped in different contexts, similar to different namespaces, each one having its own MPI communicator so to avoid interference between the different models:

- Elements *field* describe the different fields that the model can potentially produce (through `xios_send_field`) or receive (through `xios_recv_field`)
- Elements *axis, domain* and *grid* describe respectively the 1D-axis, 2D-domain and multi-dimension grids (composed of 1D-axis and 2D-domain), onto which the model produces the fields or onto which the fields will be remapped and stored in the files
- Elements *variable* define parameters used in models or by XIOS. For instance constants used in XIOS arithmetic operations or as NetCDF attribute in output files are defined as *variables*.
- Elements *file* define the content of the input or output file for a particular run and therefore configures the I/Os of a specific simulation

Note that each element may have different attributes that enable to give precisions on the temporal and spatial operations of different fields.

Without providing an exhaustive list of all XML elements and attributes, we give here few examples to illustrate how to specify particular operations on I/Os field.

The XML from figure 6 specifies to output a file named *hello_world* containing the daily average (*output_freq="1d"*) of a field *a_field* produced on a 3D grid composed of a 100x100 horizontal domain and a vertical axis with 100 levels.

```xml
<xios>
  <context id="hello_word" >     <axis_definition>
      <axis id="vertical_axis" n_glo="100" />
    </axis_definition>

    <domain_definition>
      <domain id="horizontal_domain" ni_glo="100" nj_glo="100" />
    </domain_definition>

    <grid_definition>
      <grid id="grid_3d">
        < domain domain_ref="horizontal_domain" >
        < axis   axis_ref="vertical_axis" >
      </grid>
    </grid_definition>

    <field_definition >
      <field id="a_field" operation="average" grid_ref="grid_3d" />
    </field_definition>

    <file_definition type="one_file" output_freq="1d" enabled=".TRUE.">
      <file id="output" name="hello_world" >
        <field field_ref="a_field" />
      </file>
    </file_definition>
  </context>
</xios>
```

Figure 6 – Example of a part of a XML configuration file for XIOS

Note that the operations and transformations performed on the fields by the XIOS client part can be specified in the *file* and in the *field* elements. For example, the figure 7 illustrates how to specify to remap a field with id "*field_src*" defined on a rectilinear source grid into another field with id

"*field_dest*" on an unstructured target grid. Note that the type of interpolation, here given by *order="2"* is specified in the target grid definition.
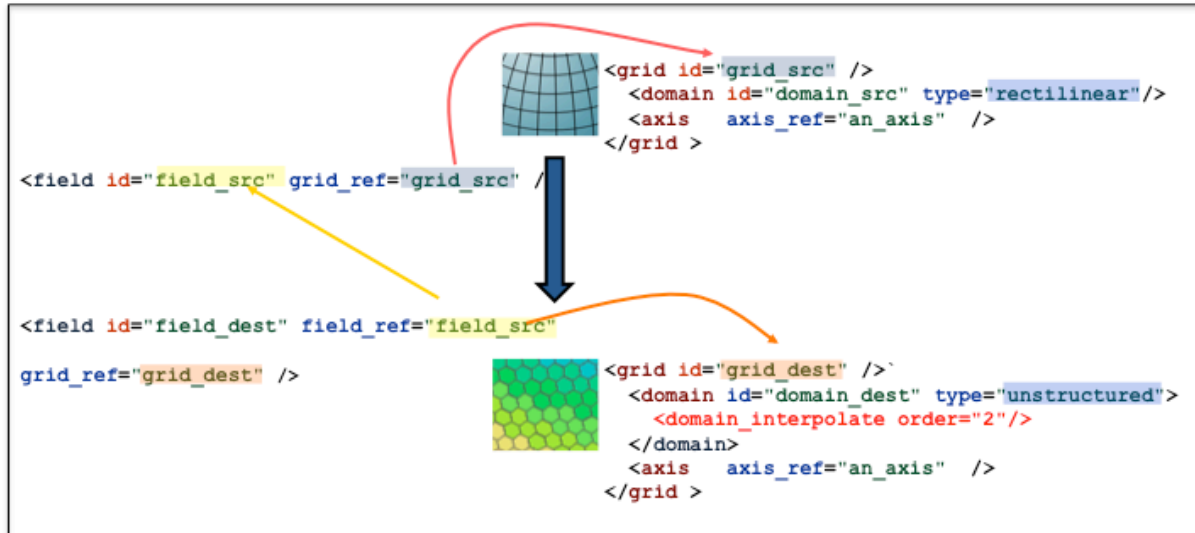

Figure 7 – Schematic view of how to specify a remapping in XIOS XML configuration file

Each transformation can be considered as a "filter" and the XML configuration allows the user to chain the filters in an arbitrary and totally flexible way (whereas in OASIS, the transformation can be turned on or off by the user but only in a specific pre-defined order). For example, it is possible to compute the monthly averaging of the daily maximum of temperature with:

```
<field id="T"          operation="maximum"/>
<field id="daily_Tmax" operation="average" freq_op="1d"> @T </field>
<field id="ave_daily_Tmax"       freq_op="1mo"> @daily_Tmax </field>
```

Note that in this case, the fields are referred to prefixed by "@": this indicates that the reference is to the result of the temporal operation and not to the original instant field. In particular, this allows controlling the order of the operations.

More complex examples can be found in [Meurdesoif et al, 2018]; for example, slide 16 explains how to calculate the zonal mean of monthly average of temperature interpolated on 850, 500 and 350 hPa pressure levels.

4.6.   User support

Given its long history of community use, an extended environment of user support has been developed around OASIS. It includes a complete web site hosted by the ENES portal (https://portal.enes.org/oasis) with, in particular, instructions on how to download the sources managed by SVN, how to get started, up-to-date documentation and tutorial. The web site also presents information on OASIS publications and presentations, coupled models using OASIS, and coupler governance. Trainings with hands-on are regularly organised at Cerfacs, once a year on average and are widely advertised in the European climate modelling community. Personal technical help is available to climate modelling labs through contracts established directly with CERFACS; such dedicated user support was also funded under IS-ENES1 and will be again under IS-ENES3. Users can also ask for help on the OASIS forum where issues are mainly answered by OASIS developers, who also regularly reply to the mails posted on the help

mailing list. Any user can also ask to follow OASIS developments that are managed under Redmine (https://inle.cerfacs.fr/projects/oasis3-mct).

The user support for XIOS is more recent and somewhat less developed. The available information is gathered on a wiki page http://forge.ipsl.jussieu.fr/ioserver/wiki, where the user can find links to the sources managed by SVN, the documentation (user and reference guides), and presentations. The developments are managed under Trac, and people can register to two different mailing lists, one for XIOS users and one for XIOS developers. A two-day tutorial has been organised once in 2015 but others are planned in the IS-ENES3 framework.

4.7.   Advantages and disadvantages of XIOS and OASIS convergence

Given the analysis presented above, it seems quite obvious that XIOS already has, as an IO server, many functions required to use it as a coupler. Extending XIOS to support full coupling and proposing it as a unified tool for coupling and I/O seems feasible. If it is realised, the convergence of XIOS and OASIS would in practice involve: 1- including in XIOS the few interpolations and forced global conservation present in OASIS but missing in XIOS; and 2- extending XIOS so to support coupling exchanges between two components, as is planned in ESiWACE and for which first steps have been achieved with the development of the second level of servers (see section 4.2). Many advantages can be associated to this "convergence" of XIOS and OASIS but also some disadvantages.

Technical aspects

Regarding field transformations, we have seen that besides few missing interpolations (2D nearest-neighbour, bilinear, bicubic) and the forced global conservation, XIOS is already richer than OASIS in terms of arithmetic, time integration, or spatial transformations; full support of 3D fields and operations like vertical interpolation, field combination, reduction, transects, are particularly noteworthy (see section 4.3). The way XIOS is configured though XML file(s) is also richer and more flexible than the quite rigid OASIS configuration based on the *namcouple* text file (see section 4.5). Using XIOS as a coupler would mean that all these developments, achieved originally for extended IO support[3], would also be available for coupling, which is definitively positive.

Another more minor advantage of the convergence would be to avoid possible communication conflicts between coupling and IOs. Today specific and well-defined communicators are defined in a coupled model using both XIOS and OASIS thanks to specific routines (`oasis_get_intracomm` and `oasis_get_intercomm`, not presented in section 4.4) called by XIOS when it is compiled with the CPP key "`use_oasis`". Also, different initialisation routines have to be called in the model whether it is using only XIOS or both XIOS and OASIS. While not difficult to do in practice, these constraints are certainly a source of error for non-expert users.

Also, runtime communication conflicts between OASIS and XIOS potentially slowing down the runs are possible although they have not been studied in detail yet. With a merged tool, all communications between the different executables of a coupled model would be managed by one single tool and would therefore be less complex to control.

---

[3] The recent work done to implement OpenMP in the client part of XIOS, exploiting multithreading for the client workflow, would also be available in a coupled framework, in particular the management of the communication by the threads, currently between the clients and the servers, but ultimately between two component models of a coupled system.

The main technical disadvantage of having only one merged tool for coupling and IO is that the resulting software may become too big and labyrinthine to manage and to further evolve. There is indeed a non-negligible risk that only few very specialised developers would have enough expertise to understand and modify the code, which is not advisable for a community code.

User adoption

The first advantage of the XIOS and OASIS convergence is that coupled model developers aiming at using community tools for IO and for coupling would have to learn about and use only one single tool instead of two. This would mean interfacing their component model with only one API and mastering only one configuration framework. Also, tasks such as porting models to new platforms would be simpler with only one tool to port.

The drawback however is that a user wanting only to couple two components would necessarily face the power but also the complexity of XIOS in particular regarding its XML configuration (see section 4.5). Simple templates or friendly graphical user interfaces would have to be developed and made available to users. Indeed, XIOS flexibility regarding configuration is definitively an asset but the related complexity could seriously hold back community's adoption.

Finally, it could be hard to convince established OASIS users to switch to another tool they don't master, even if, as we have detailed in section 4.4, their APIs are quite similar and the switch should not technically cause any specific problem. Many users in particular, appreciate reading, and sometimes even modifying, OASIS sources written in Fortran. This would not be the case anymore with XIOS, whose sources are written in C++.

Resources for development

Another very important positive aspect of the convergence would be to avoid duplication of development efforts. Each common function would be implemented once in one tool. For example, we would now like to support 3D fields and grids in OASIS and therefore 3D interpolation. We have seen above that this is already supported in XIOS as it has been implemented specifically for CMIP6. Although we could certainly base OASIS 3D support using XIOS as an example, in practice this would mean recoding in Fortran in OASIS what is already coded in C++ in XIOS.

The convergence could and should also mean gathering the current OASIS and XIOS human resources around one same tool. Currently, outside temporary contract funding, OASIS benefits from 1 FET (75% by CNRS and 25% by Cerfacs) and XIOS from only about 0.5 FET (by IPSL/LSCE). The manpower devoted to the development of community tools is scarce and it would be essential for this convergence to work to establish new collaborations around the converged tool. Instead of competing, the diverse expertise of the different groups (pure technical development, technological watch, project management, user support, etc.) should team up and could have a leveraging effect.

The risk involved here is relatively low as the two main developing groups, IPSL for XIOS and Cerfacs for OASIS, are both located in France, know each other well and already interact very positively. This has proven to be very efficient in the set-up of the CMIP6 simulations for which models of both IPSL and CNRM-CERFACS groups, i.e. IPSLCM6 and CNRM-CM6, use XIOS and its related tool dr2xml (https://github.com/senesis/dr2pub/blob/master/dr2xml.py) to directly produce cmor and CMIP6 compliant output.

4.8. Strategy and conclusions

From XIOS and OASIS comparison (sections 4.2 to 4.6) and from the advantage and disadvantage analysis (section 4.7), we conclude here that XIOS has a lot of potential to evolve from an IO server to become also a coupler.

The option of truly merging code from the two tools is discounted because (i) XIOS already contains most of the functionality required to act as a coupler that could replace OASIS; (ii) user migration from OASIS to XIOS is not too complex to achieve and (iii) the two tools are written in different languages and with different coding styles and the effort of merging functionality would be higher than the effort of extending XIOS.

Hence, this paper proposes that the best way forward is likely to extend XIOS communication so to support coupling between two components and develop XIOS as a unique tool for IO and coupling.

However, as coupling functionality in XIOS is not fully developed and has therefore not been tested and validated yet, it is obviously too early to evaluate if XIOS proves to be an efficient coupler and decide if we should encourage the climate modelling community to switch to XIOS for coupling.

We see the following steps to get to an informed decision:

1. First the coupling functionality in XIOS has to be finalised based on developments done for the second level of servers and has to be fully validated with a simple toy model implementing ping-pong exchanges, i.e. back-and-forth exchanges between two "empty" component models.

2. Then the interpolations in XIOS have to be complemented with 2D nearest-neighbour, bilinear and bicubic schemes, and the forced global conservation has to be added. The analysis of the quality of these interpolations and of the conservative remapping as implemented in XIOS [Kritsikis et al, 2017] then has to be completed. A detailed comparison with other libraries such as the SCRIP library used in OASIS or ESMF (https://www.earthsystemcog.org/projects/esmf/) has to be performed to quantify the benefits and drawbacks, if any, of the XIOS implementations.

3. Different test cases covering the test suite regularly used to validate OASIS development then have to be run to have confidence that XIOS covers at least all current OASIS functionalities.

4. It will then be interesting to run the IS-ENES2 coupling technology benchmark with XIOS and compare with results from other couplers [Valcke et al, 2017]

5. A full analysis of XIOS workflow performance then has to be completed. First steps in this direction have started but this preliminary work has to continue. Optimisations have to be identified and implemented to improve current XIOS performances.

6. Coupling with XIOS would then be implemented in one real coupled system, probably IPSL-CM as a demonstrator, and then in a second real coupled system, probably CNRM-CM, and fully validate XIOS coupling in these two real cases.

Only if all these steps are achieved successfully, then we would be confident that XIOS can be considered as a community coupler and we would encourage OASIS users to use it as such, informing them on the approach taken to get to that conclusion. At that point, some developments could be envisaged so to provide an OASIS-compatible API to XIOS, i.e. an API corresponding to, or very similar to, OASIS API but using XIOS underneath. This could help the transition from OASIS to XIOS, especially for OASIS-only users.

We can remark that, if it happens, the process will be gradual and we evaluate that it will take at least 3 years to complete. In the mean time, it is therefore essential to continue OASIS support and some light development, as proposed in ESiWACE2 and IS-ENES3 projects recently selected by the EU. And even if

we switch to XIOS for HPC coupling, particularly in the exascale perspective, investing in that tool and organising joint IPSL-Cerfacs developments, it is also very likely that we will have to continue user support on a frozen OASIS version for quite some time, so to help a smooth transition from OASIS to XIOS on a longer timeframe.

Finally, we stress here that it would be advisable that this process be realised not only by Cerfacs and IPSL, but together with other climate modelling groups in Europe, under a certain form of community governance. IS-ENES3 will help in this regard, as one of its objectives is the governance of the climate modelling infrastructure, which of course includes community tools.

5. References (Bibliography)

[Meurdesoif et al, 2018]
M.H. Nguyen, R. Lacroix, O. Abramkina,Y.,Wang, A. Caubel, Y. Meurdesoif (IPSL/LSCE), S. Denvil (IPSL), S. Senesi, D. Saint Martin (CNRM), M.P. Moine, S. Valcke (Cerfacs), 2018: XIOS - Output whole CMIP6 data through the new XIOS parallel workflow functionalities, 5th ENES Workshop on High Performance Computing for Climate and Weather, 17-18 May 2018 Lecce,
https://redmine.dkrz.de/attachments/download/3972/Session3_Meurdsoif_XIOS.pdf

[Craig et al, 2017]
A. Craig, S. Valcke, L. Coquart, 2017: Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0, Geosci. Model Dev., 10, 3297-3308, https://doi.org/10.5194/gmd-10-3297-2017.

[Jones 1999]
P.W. Jones, 1999. Conservative remapping: First- and second-order conservative remapping, Mon. Weather Rev., 127, 2204–2210.

[Kritsikis et al, 2017]
E. Kritsikis, M. Aechtner, Y. Meurdesoif, and T Dubos, 2017. Conservative interpolation between general spherical meshes, Geosci. Model Dev., 10, 425-431,
https://doi.org/10.5194/gmd-10-425-2017

[Larson et al, 2005]
J. Larson, R. Jacob, and E. Ong, 2005. The Model Coupling Toolkit:
A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models, Int. J. High Perf. Comp. App., 19, 277–292, https://doi.org/10.1177/1094342005056116 .

[Valcke et al, 2017]
S. Valcke, G. Jonville, R. Ford, M. Hobson, A. Porter and G. Riley, 2017. Report on benchmark suite for evaluation of coupling strategies, UMR 5318 CECI, CERFACS/CNRS, TR-CMGC-17-87, Toulouse, France (http://cerfacs.fr/wp-content/uploads/2017/05/GLOBC-TR-IS-ENES2_D10.3_MAI2017.pdf)

[Valcke et al, 2018]
S. Valcke, T. Craig and L. Coquart, 2018. OASIS3-MCT User Guide, OASIS3-MCT4.0, CECI, Université de Toulouse, CNRS, CERFACS - TR-CMGC-18-77, Toulouse, France. https://cerfacs.fr/wp-content/uploads/2018/07/GLOBC-TR-oasis3mct_UserGuide4.0_30062018.pdf

6. Dissemination and uptake

6.1. Uptake by the targeted audience

As indicated in the Description of the Action, the audience for this deliverable is

X	The general public (PU)
	The project partners, including the Commission services (PP)
	A group specified by the consortium, including the Commission services (RE)
	This reports is confidential, only for members of the consortium, including the Commission services (CO)

6.2. This is how we are going to ensure the uptake of the deliverables by the targeted audience

The question of convergence between IO servers and couplers is of high interest for the climate modelling and weather communities. Therefore this White Paper, which presents the current most advanced strategic thinking on the convergence between IPSL IO server, XIOS, and the widely used coupler, OASIS, will be widely distributed among ESiWACE partners and OASIS and XIOS users. It will also be submitted to OASIS Advisory Board for internal discussions.

First ideas on this strategic topic were already presented at the ENES Strategy Meeting organised in Reading on October 25-27 2016 (see below).

7. The delivery is delayed

☐ Yes ☒ No

8. Changes made and/or difficulties encountered, if any

The production of this deliverable did not encountered any specific difficulty as XIOS developers at IPSL and OASIS developers at CERFACS have been interacting closely on XIOS during the past two years, in the framework of CMIP6 (see 9.1)

9. Sustainability

9.1 Lessons learnt: both positive and negative that can be drawn from the experiences of the work to date

The experience of joint work on XIOS between XIOS developers at IPSL and OASIS developers at CERFACS has been very positive. These developers have been interacting closely, also together with developers from CNRM (Météo France), on XIOS during the past two years. Indeed XIOS is used in all components of CNRM-CM6, the coupled model developed by CNRM-Cerfacs in the framework of CMIP6. Cerfacs developers therefore spent significant time getting familiar with XIOS and learning about all its functionality and limitations. This is an essential step to master both tools and to be able to envisage a convergence between XIOS and OASIS having a global view of the question.

9. 2 Links built with other deliverables, WPs, and synergies created with other projects

There is a strong link between this deliverable and D2.4 "XIOS adapted to many-core architecture: Development of Multithreaded Version of XIOS including on-line Diagnostics". Indeed this last deliverable describes the latest XIOS developments which are important in the evolution of the IO server toward supporting code coupling.

A synergy between ESiWACE on this question of convergence between IO servers and couplers has been naturally established with the EU project IS-ENES2, as this question is of high interest for the climate modelling and weather communities. As detailed above, first ideas on this strategic topic were presented at the ENES Strategy Meeting organised by IS-ENES2 in Reading on October 25-27 2016. This topic will continue to be addressed in IS-ENES3, if funded, as governance of its software infrastructure is one of its identified objectives.

10. Full track of dissemination activities

Type of dissemination and communication activities	Details	Location , dates	Audience	Zenodo record / link to website	Estimated number of persons reached
Participation to a workshop	S. Valcke, Cerfacs, "Convergence of coupling and I/O software tools?" ENES Strategy Meeting	October 25-27 2016, Reading (UK)	Scientific Community - partners of ESiWACE and IS-ENES2 (higher education, research)	https://www.ncas.ac.uk/en/2593 https://portal.enes.org/community/announcements/news/save-the-date-enes-foresight-and-convergence-workshops	40

Intellectual property rights resulting from this deliverable

Not applicable.