Quickstart Intro

These instructions should get you started on building your own coupled system with MOSSCO.

We will build a system that connects a pelagic ecosystem, running in a 1D ocean model to a benthic pelagic ecosystem, using two external models

- FABM (Framework for Aquatic Biogeochemistry)
- GOTM (General Ocean Turbulence Model).

For these models, we will combine several MOSSCO components (you can find them in src/components)

- constant (Delivers constants to other components)
- gotm (Hydrology using GOTM)
- fabm_gotm (Ecology of FABM running in GOTM context)
- fabm_sediment (Ecology of FABM running in ocean soil)

Building MOSSCO

Define your FABM, GOTM, and MOSSCO directories (yours may be different), i.e. where the source code for these models resides:

```
export FABMDIR=$HOME/maecs/fabm
export GOTMDIR=$HOME/maecs/gotm
export MOSSCO_DIR=$HOME/MOSSCO/mossco-code
export MOSSCO_SETUPDIR=$HOME/MOSSCO/mossco-setups
```

If not done so, git clone the source codes into these directories:

```
git clone git://git.code.sf.net/p/fabm/code $FABMDIR
git clone git://git.code.sf.net/p/gotm/code $GOTMDIR
git clone git://git.code.sf.net/p/mossco/code $MOSSCO_DIR
git clone git://git.code.sf.net/p/mossco/setups $MOSSCO_SETUPDIR
```

Or, if you have downloaded them once, make sure they are all up to date:

```
for F in $FABMDIR $GOTMDIR $MOSSCO_DIR $MOSSCO_SETUPDIR ; do (cd $F ; git
pull ) ; done
```

Next, define all environment variables that are need for GOTM, FABM, and MOSSCO (these

may vary on your system)

```
export FORTRAN_COMPILER=GFORTRAN
export NETCDF_VERSION=NETCDF4
unset FABM
export MOSSCO_FABMDIR=$FABMDIR
export MOSSCO_GOTMDIR=$GOTMDIR
export ESMFMKFILE=/opt/esmf/lib/libg/Linux.gfortran.64.mpich2.esmf6/esmf.m
k
```

Finally, build the MOSSCO infrastructure

```
make
```

Configuring your example

One way to create coupled systems in MOSSCO is to specify the coupling, i.e., the participating components and the coupling intervals in a text file (in a human-readable `.yaml` format) and let MOSSCO take care of creating an executable based on this configuration.

Go to the folder `$MOSSCO_DIR/examples/generic`. There, you will find a file called `maecs_omexdia.yaml`

```
cd $MOSSCO_DIR/examples/generic
cat maecs_omexdia.yaml
```

The first lines of this file are:

```
coupling:
  - components:
    - gotm       # The ocean in 1D (physics)
    - fabm_gotm  # The ocean biogeochemistry, choose models in fabm.nml
    interval: 360 s # The coupling interval of these two components
  - components:
    - fabm_gotm
    - pelagic_benthic_coupler # The special coupler between pelagial and b
enthos
    - fabm_sediment
```

The coupled system, or coupling, is described as a list of coupling pairs, where each pair is the name of the component. Optionally, a special coupler component can be named between a coupling pair; also optionally, a coupling interval can be chosen.

You can now create the source code for a coupled system (which will end up in the file `toplevel_component.F90` and a special `Makefile.coupling` snippet) by invoking the create_coupling script

```
./create_coupling.py maecs_omexdia.yaml
```

This will give a output information that it processed seven components.

```
Components to process: ['link_coupler', 'pelagic_benthic_coupler', 'fabm_s
ediment', 'constant', 'gotm', 'fabm_gotm', 'benthic_pelagic_coupler']
```

Now type make to create an executable for your home-brew coupled system

```
make
```

Running a setup

Now that you have an executable `$MOSSCO_DIR/examples/generic/coupling`, you can use this executable in a setup (found in `$MOSSCO_SETUPDIR`) to actually perform a simulation.

```
cd $MOSSCO_SETUPDIR/helgoland
ls -l
```

You can see many namelists here, of which the most important ones are

- `mossco_run.nml` (for controlling the simulation time)
- `fabm.nml` (for controlling the pelagic fabm models to be used)
- `fabm_sed.nml` (for controlling the benthic fabm models)

Depending on your choice of FABM models, you may have to edit or add more namelist files.

Finally, execute your coupled system in the setup: $MOSSCO_DIR/examples/generic/coupling

You will see some screen output, and at least two files will be written

- `netcdf_component.nc` (a netcdf file containing all the output)
- `PET0.Helgoland` (a log file for your simulation)