HIDROMOD

MOHID meeting 7-8 of June, 2018 Lisbon

One Idea => One Module

Taking advantage of MOHID unique code structure

Paulo Leitão

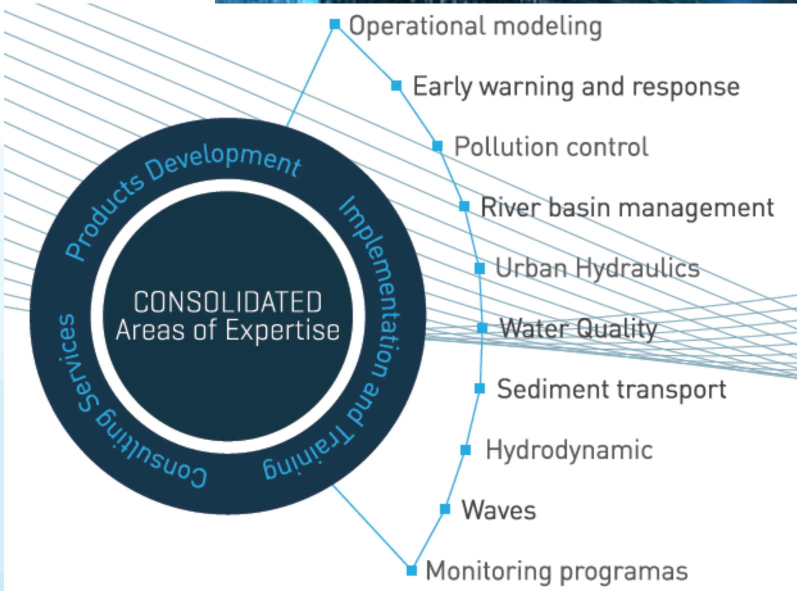
Overview

- Why MOHID is important to Hidromod ?
- Open source (Strengths vs Threats)
- New good idea => new module
- Steps to add a new module
  - Module Litter example
- Conclusions

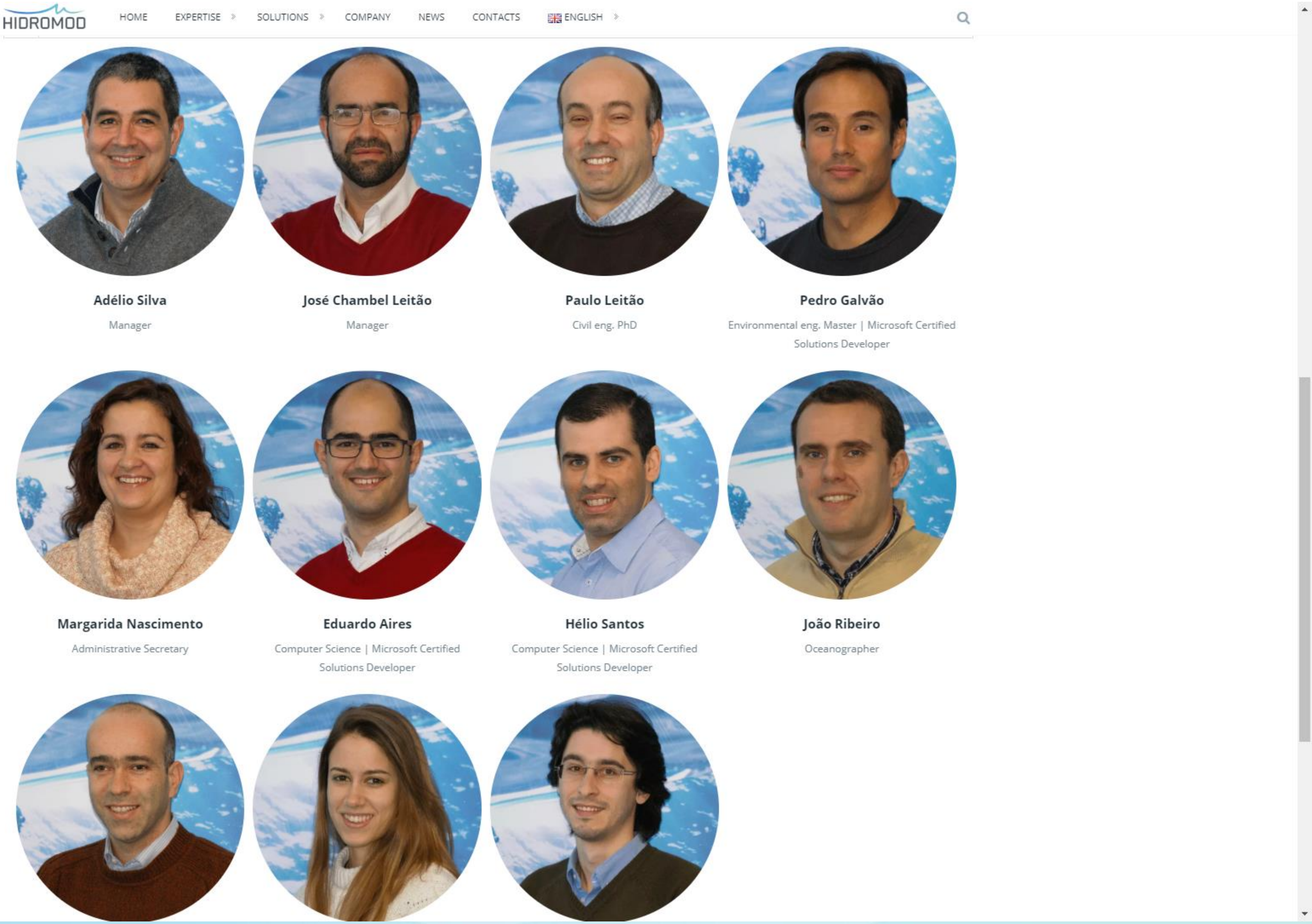
HIDROMOD

- HIDROMOD is an international company acting in the areas of:
  - ✓ **Consultancy**: Whole water cycle and information technologies
  - ✓ **Services**: Forecast systems, Early warning systems, Professional support (e.g. *Portugal, Spain, France, Brazil, Argentina, Colombia, Malaysia, Oman*)
  - ✓ **Innovation**: Implementation of new approaches where modelling and technology are efficiently blend

- Main characteristics:
  - ✓ Highly qualified staff with several Ph.D. and Ms.C.
  - ✓ Over 450 projects in the last 25 years
  - ✓ 1/3 – R&D Projects



HIDROMOD – Team



Adélio Silva
Manager

José Chambel Leitão
Manager

Paulo Leitão
Civil eng. PhD

Pedro Galvão
Environmental eng. Master | Microsoft Certified Solutions Developer

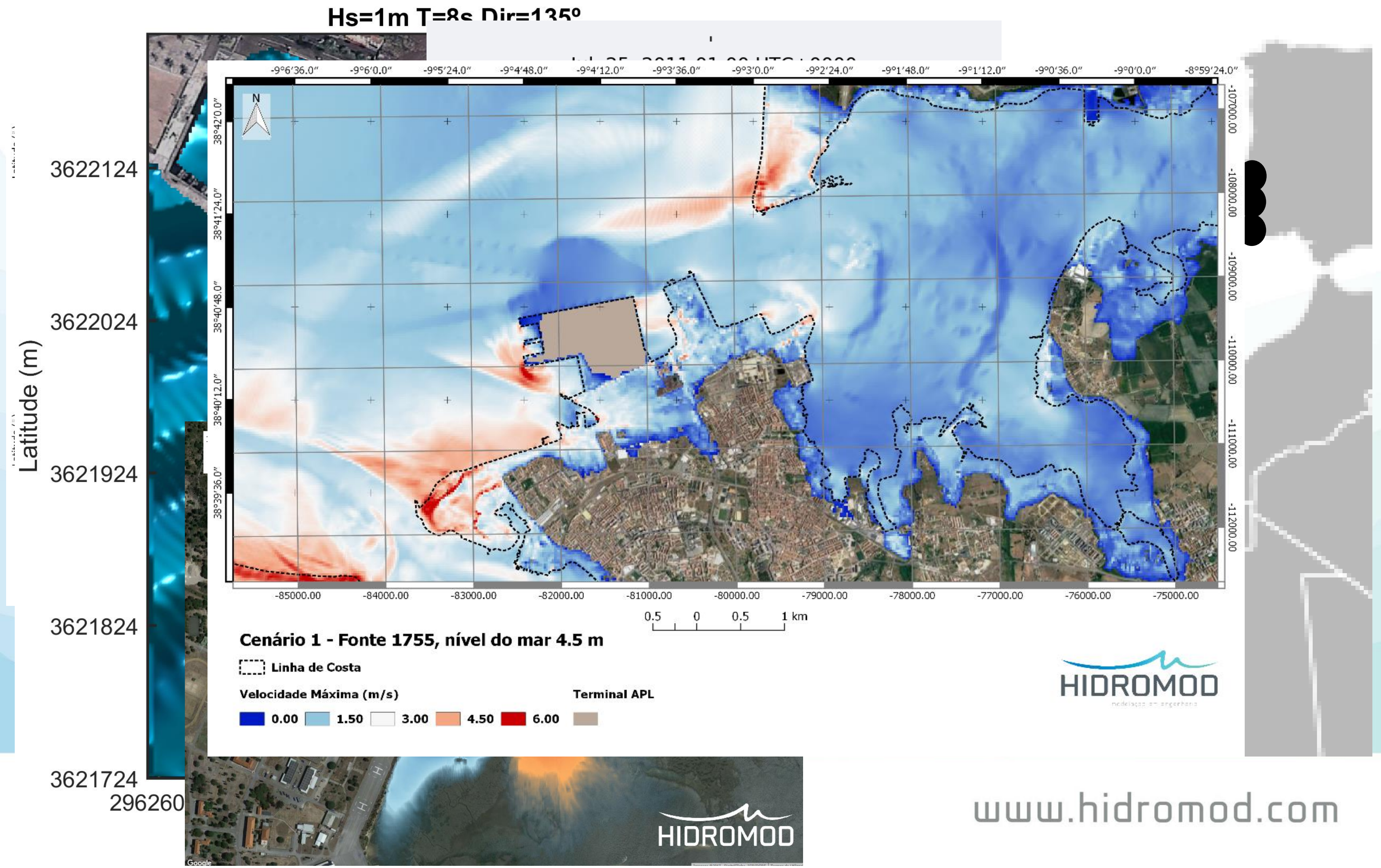
Margarida Nascimento
Administrative Secretary

Eduardo Aires
Computer Science | Microsoft Certified Solutions Developer

Hélio Santos
Computer Science | Microsoft Certified Solutions Developer

João Ribeiro
Oceanographer

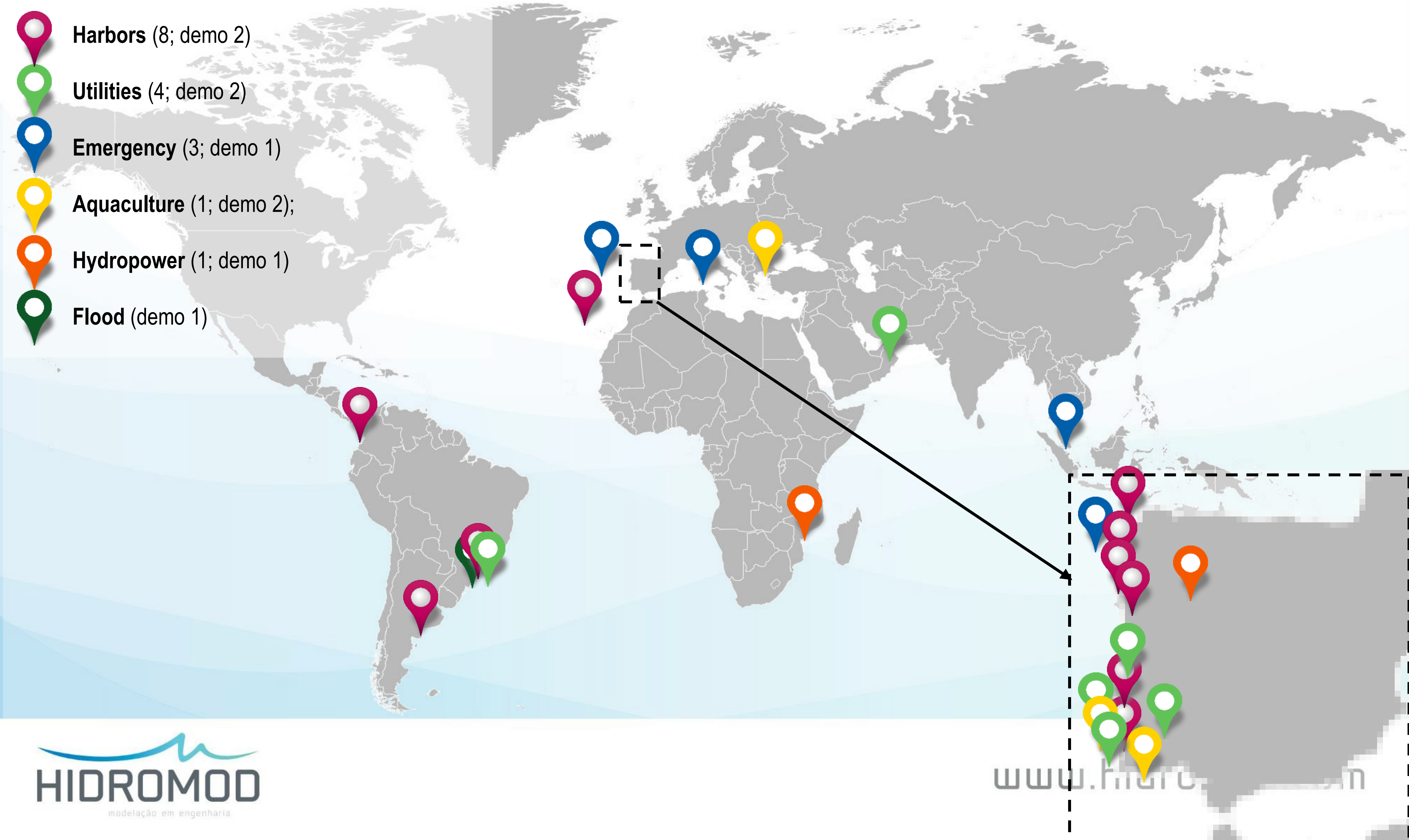
MOHID – consultancy (Last 12 months)



www.hidromod.com

Forecast Services – MOHID (model & tools)

HIDROMOD's tailored forecast services (with signed contracts (18) or R&D demos (9))



- **Harbors** (8; demo 2)
- **Utilities** (4; demo 2)
- **Emergency** (3; demo 1)
- **Aquaculture** (1; demo 2);
- **Hydropower** (1; demo 1)
- **Flood** (demo 1)

HIDROMOD

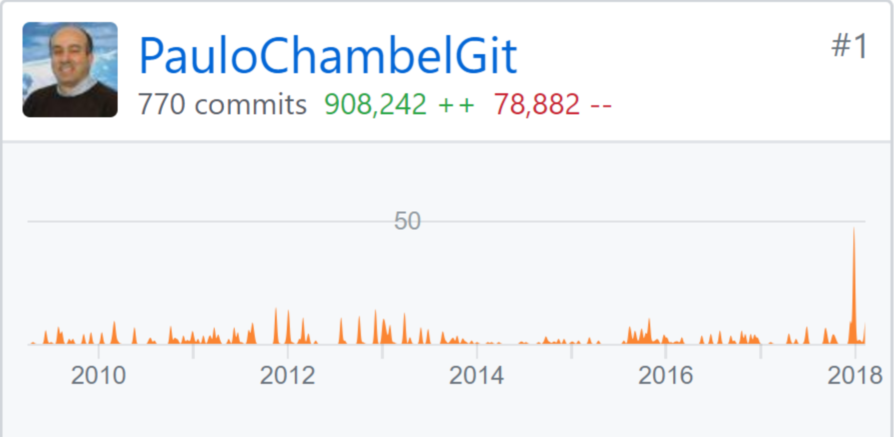
modelação em engenharia

www.hidro...m

MOHID – Innovation

Why does Hidromod invests in MOHID?

- Open source project (costs are linked with knowledge/learning);
- It allows us to adapt to client specific needs;
- It allows us to keep up with the competition, from a scientific point of view;
- Simple input/output allowing easy integration with other technologies.


PauloChambelGit

770 commits 908,242 ++ 78,882 --

MOHID – Threats

- Threats:
  - Open source project => constant changes to test new hypothesis
- Solutions:
  - Test all ideas - source code repositories – branch
  - Implement new good and tested ideas => new module

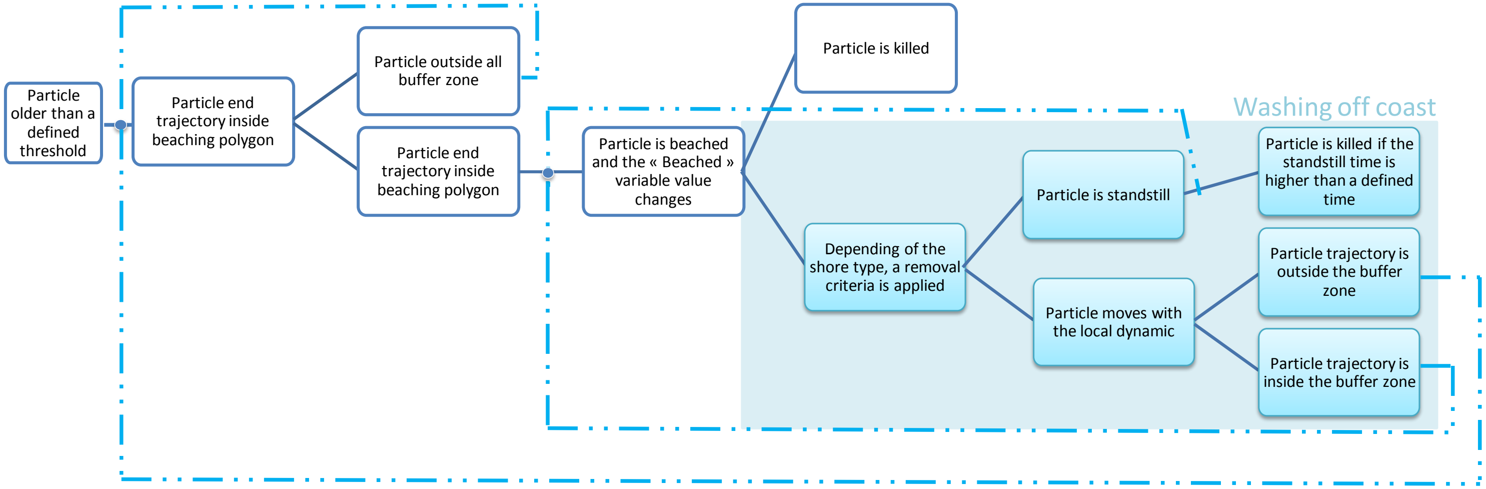
MOHID – Steps to add a new module

- Step 1:
  - Go to Software/MOHIDBase1/ModuleGlobalData.f90 and increment the parameter MaxModules in the number of modules you want to add;
- Step 2:
  - Copy the Software/Shell/ModuleShell.f90 and replace by the name you want (e.g. ModuleLitter.f90);
- Step 3:
  - Open the new module and replace the name Shell by a proper name (e.g. replace Shell by Litter);
- Step 4:
  - Your new module is ready to use. You only need to declare the module using the “Use” instruction (e.g. Use ModuleLitter in the ModuleLagrangianGlobal)

MOHID – ModuleLitter example

LEMA
Intelligent marine Litter removal and
Management for local Authorities

LIFE15 ENV/ES/000252

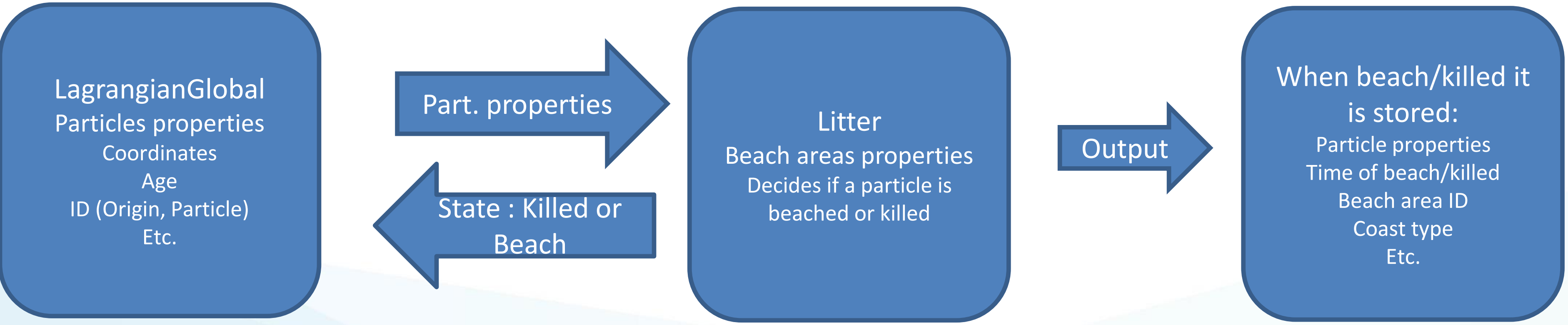


Particle older than a defined threshold
Particle end trajectory inside beaching polygon
Particle outside all buffer zone
Particle end trajectory inside beaching polygon
Particle is killed
Particle is beached and the « Beached » variable value changes
Washing off coast
Depending of the shore type, a removal criteria is applied
Particle is standstill
Particle is killed if the standstill time is higher than a defined time
Particle moves with the local dynamic
Particle trajectory is outside the buffer zone
Particle trajectory is inside the buffer zone

rivages PRO TECH

suez

Beaching – Implementation – Data Flux



LagrangianGlobal
Particles properties
Coordinates
Age
ID (Origin, Particle)
Etc.

Part. properties

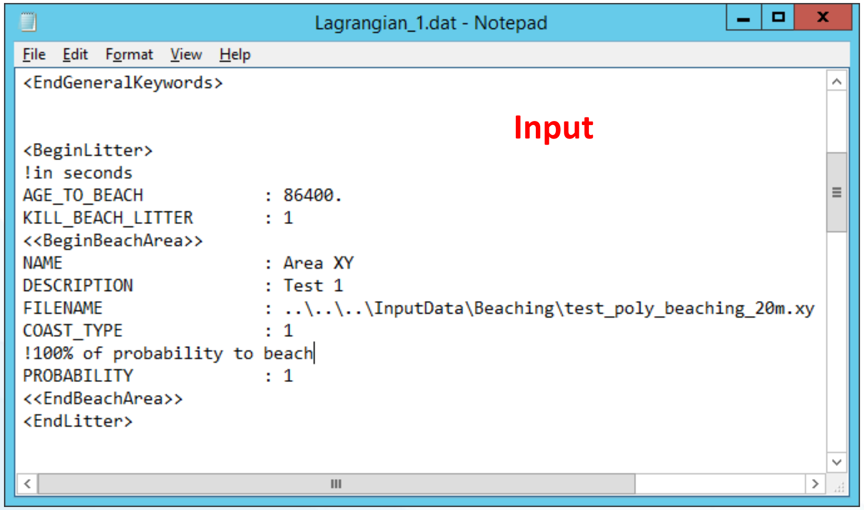
State : Killed or Beach

Litter
Beach areas properties
Decides if a particle is beached or killed

Output

When beach/killed it is stored:
Particle properties
Time of beach/killed
Beach area ID
Coast type
Etc.

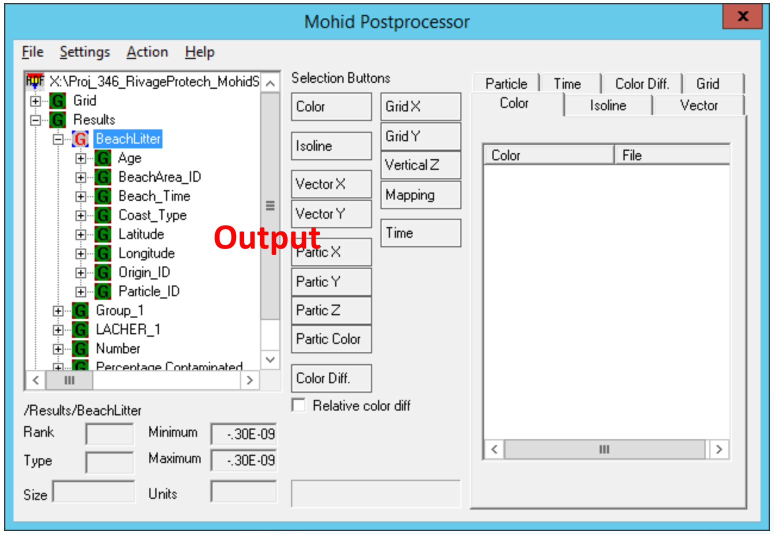
Beaching – Implementation – Input/Output



Lagrangian_1.dat - Notepad

Input

```
<EndGeneralKeywords>


<BeginLitter>
!in seconds
AGE_TO_BEACH              : 86400.
KILL_BEACH_LITTER         : 1
<<BeginBeachArea>>
NAME                      : Area XY
DESCRIPTION               : Test 1
FILENAME                  : ..\..\..\InputData\Beaching\test_poly_beaching_20m.xy
COAST_TYPE                : 1
!100% of probability to beach
PROBABILITY               : 1
<<EndBeachArea>>
<EndLitter>
```



Mohid Postprocessor

Output

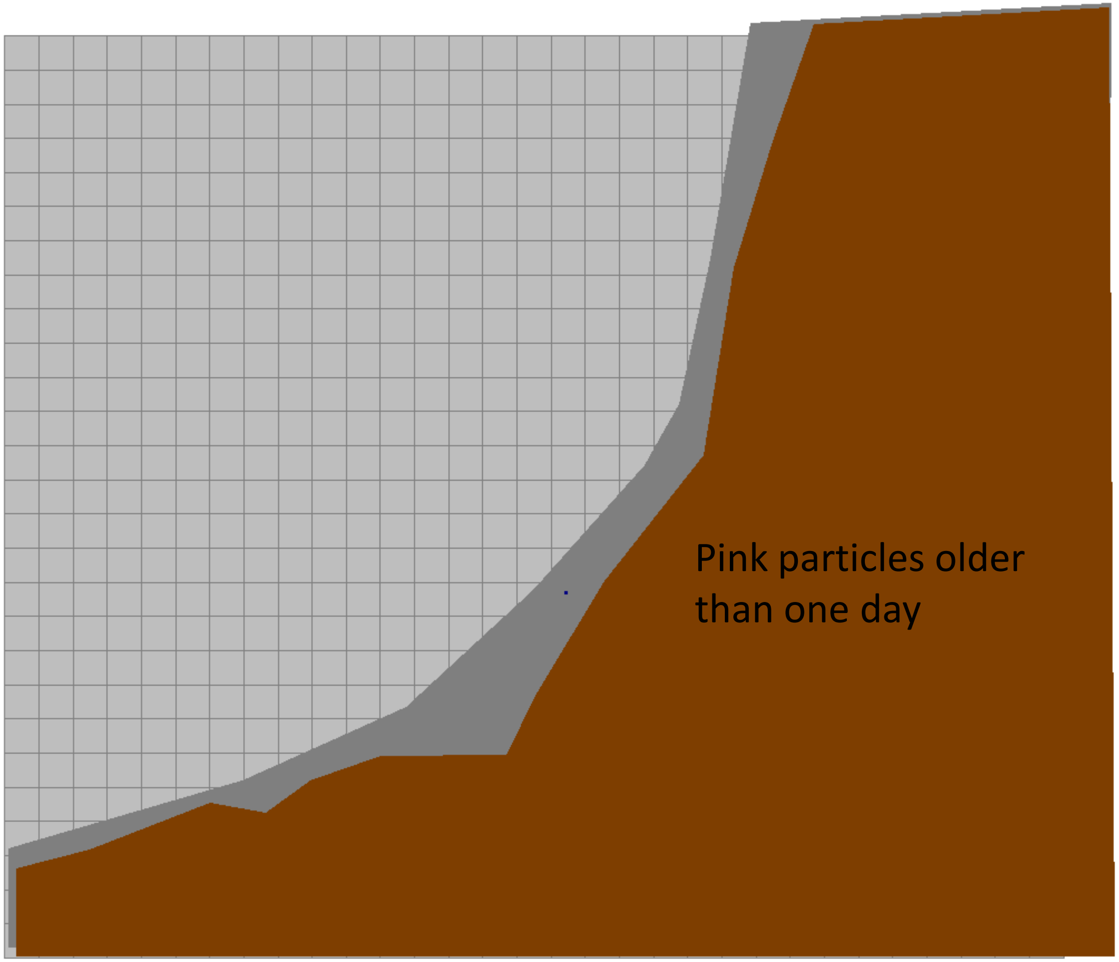
```
X:\Proj_346_RivageProtech_MohidS
  Grid
  Results
    BeachLitter
      Age
      BeachArea_ID
      Beach_Time
      Coast_Type
      Latitude
      Longitude
      Origin_ID
      Particle_ID
    Group_1
    LACHER_1
    Number
    Percentage Contaminated
```

/Results/BeachLitter

Minimum: -.30E-09

Maximum: -.30E-09

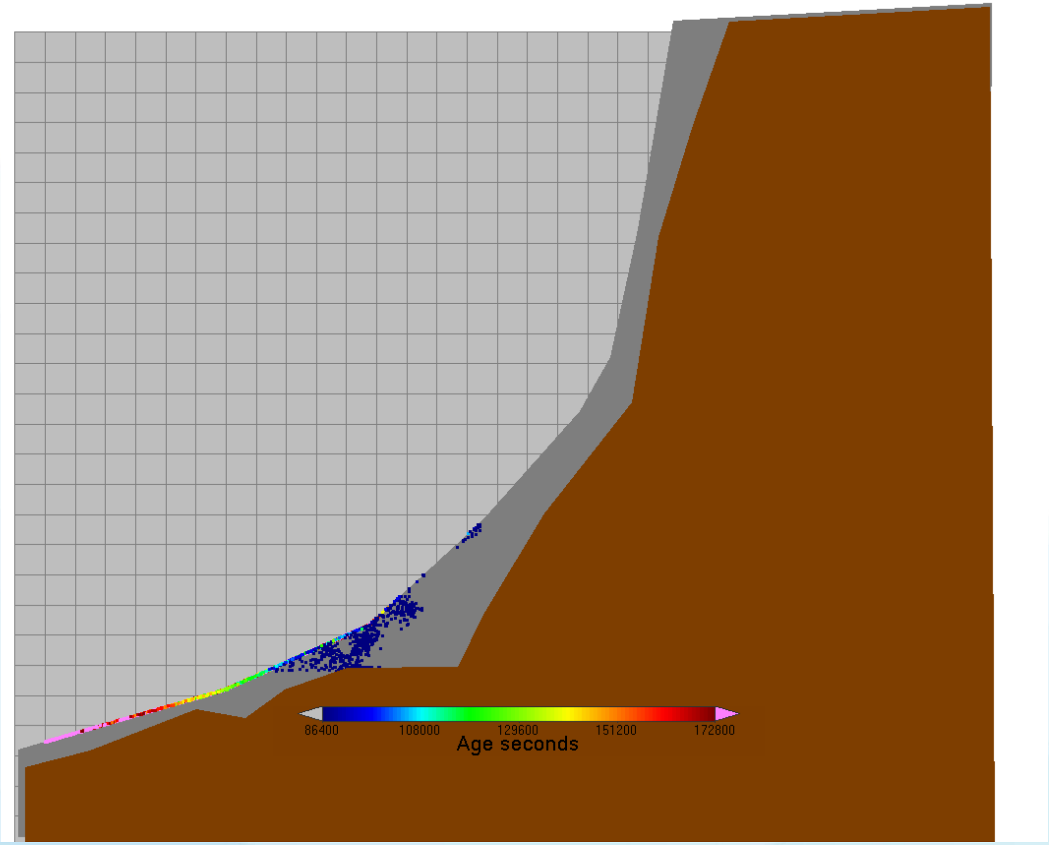
Beaching – Test results – transient

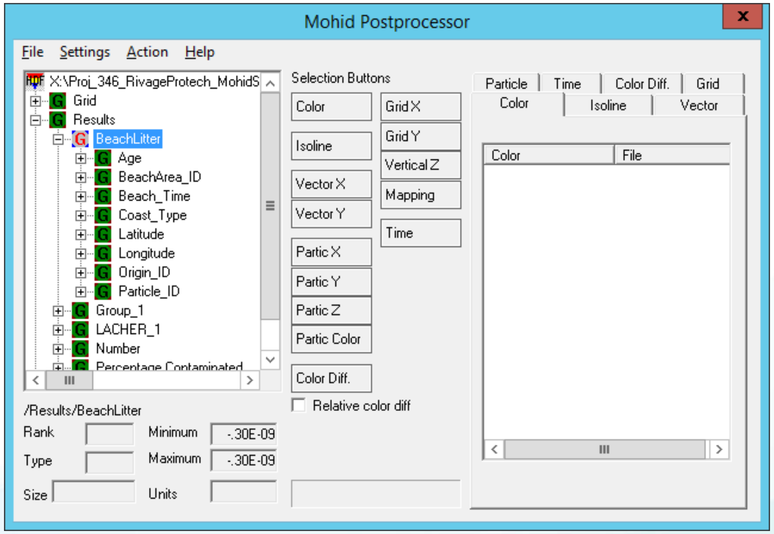


Pink particles older than one day

Light grey – model domain with no land cells
Dark grey – beach litter area polygon
Brown – land polygon

Beaching – Test results – BeachLitter - final





X:\Proj_346_RivageProtech_MohidS
- Grid
- Results
  - BeachLitter
    - Age
    - BeachArea_ID
    - Beach_Time
    - Coast_Type
    - Latitude
    - Longitude
    - Origin_ID
    - Particle_ID
  - Group_1
  - LACHER_1
  - Number
  - Percentage Contaminated

/Results/BeachLitter

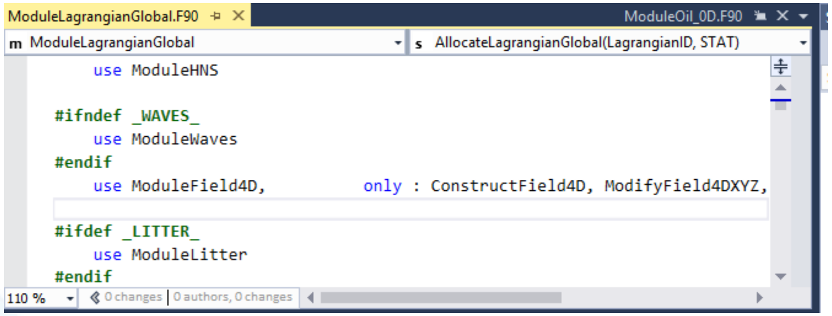
Rank Minimum -.30E-09

Type Maximum -.30E-09

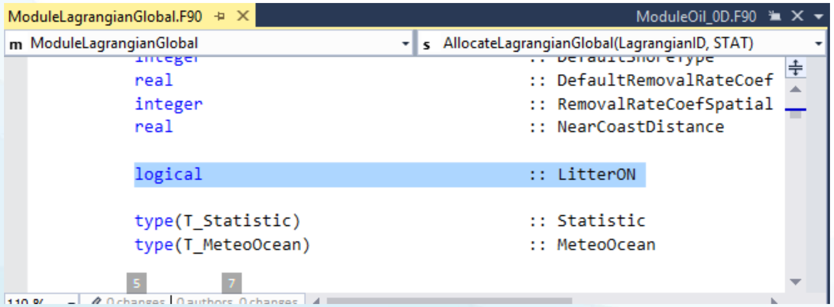
Size Units

☐ Relative color diff

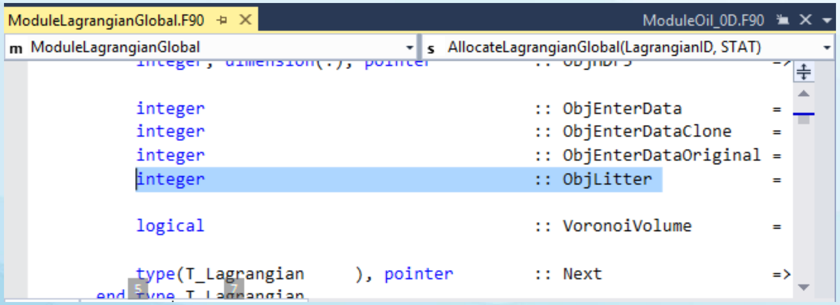
ModuleLagrangianGlobal using ModuleLitter


ModuleLagrangianGlobal.F90
ModuleOil_0D.F90
m ModuleLagrangianGlobal
s AllocateLagrangianGlobal(LagrangianID, STAT)
110 %
0 changes | 0 authors, 0 changes

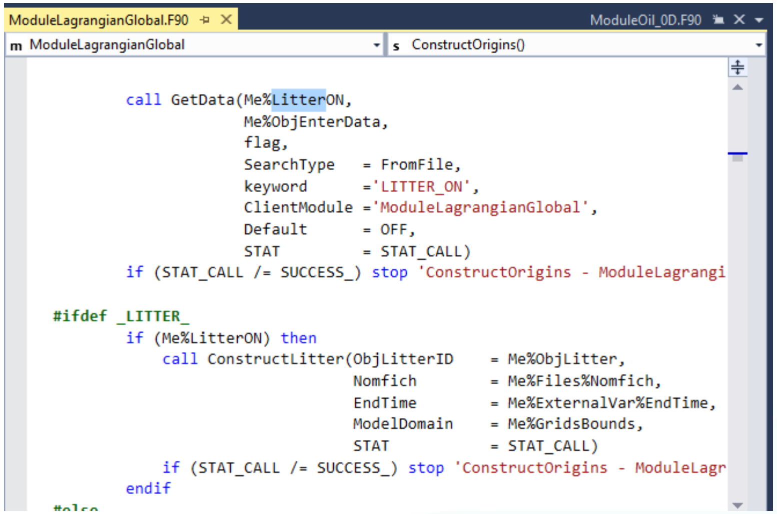
```fortran
        use ModuleHNS

    #ifndef _WAVES_
        use ModuleWaves
    #endif
        use ModuleField4D,            only : ConstructField4D, ModifyField4DXYZ,

    #ifdef _LITTER_
        use ModuleLitter
    #endif
```


ModuleLagrangianGlobal.F90
ModuleOil_0D.F90
m ModuleLagrangianGlobal
s AllocateLagrangianGlobal(LagrangianID, STAT)
5
7
110 %
0 changes | 0 authors, 0 changes

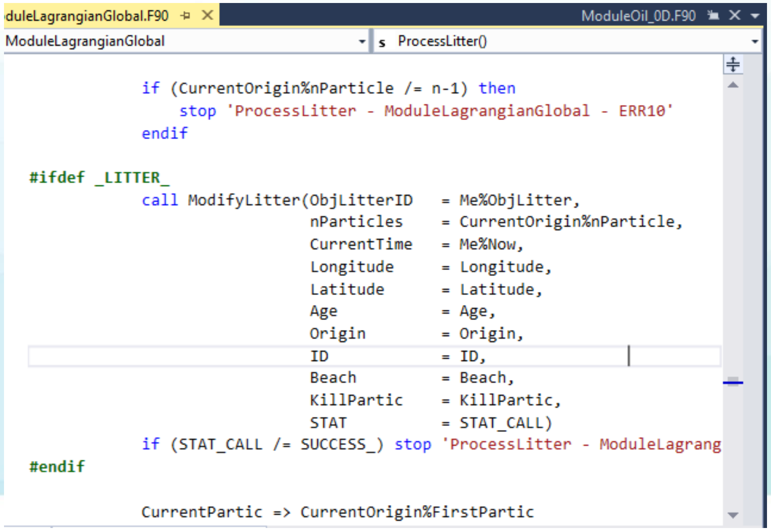
```fortran
            integer                                 :: DefaultShoreType
            real                                    :: DefaultRemovalRateCoef
            integer                                 :: RemovalRateCoefSpatial
            real                                    :: NearCoastDistance

            logical                                 :: LitterON

            type(T_Statistic)                       :: Statistic
            type(T_MeteoOcean)                      :: MeteoOcean
```


ModuleLagrangianGlobal.F90
ModuleOil_0D.F90
m ModuleLagrangianGlobal
s AllocateLagrangianGlobal(LagrangianID, STAT)
5
7

```fortran
            integer, dimension(:), pointer          :: ObjHDFs              =>

            integer                                 :: ObjEnterData         =
            integer                                 :: ObjEnterDataClone    =
            integer                                 :: ObjEnterDataOriginal =
            integer                                 :: ObjLitter            =

            logical                                 :: VoronoiVolume        =

            type(T_Lagrangian     ), pointer        :: Next                 =>
        end type T_Lagrangian
```



```fortran
            call GetData(Me%LitterON,
                         Me%ObjEnterData,
                         flag,
                         SearchType   = FromFile,
                         keyword      ='LITTER_ON',
                         ClientModule ='ModuleLagrangianGlobal',
                         Default      = OFF,
                         STAT         = STAT_CALL)
            if (STAT_CALL /= SUCCESS_) stop 'ConstructOrigins - ModuleLagrangi

    #ifdef _LITTER_
            if (Me%LitterON) then
                call ConstructLitter(ObjLitterID    = Me%ObjLitter,
                                     Nomfich        = Me%Files%Nomfich,
                                     EndTime        = Me%ExternalVar%EndTime,
                                     ModelDomain    = Me%GridsBounds,
                                     STAT           = STAT_CALL)
                if (STAT_CALL /= SUCCESS_) stop 'ConstructOrigins - ModuleLagr
            endif
    #else
```



```fortran
            if (CurrentOrigin%nParticle /= n-1) then
                stop 'ProcessLitter - ModuleLagrangianGlobal - ERR10'
            endif

    #ifdef _LITTER_
            call ModifyLitter(ObjLitterID   = Me%ObjLitter,
                              nParticles    = CurrentOrigin%nParticle,
                              CurrentTime   = Me%Now,
                              Longitude     = Longitude,
                              Latitude      = Latitude,
                              Age           = Age,
                              Origin        = Origin,
                              ID            = ID,
                              Beach         = Beach,
                              KillPartic    = KillPartic,
                              STAT          = STAT_CALL)
            if (STAT_CALL /= SUCCESS_) stop 'ProcessLitter - ModuleLagrang
    #endif

            CurrentPartic => CurrentOrigin%FirstPartic
```

Conclusions

- MOHID – open source project adapted to HIDROMOD’s consultancy and services

- MOHID is very important to HIDROMOD’s work

- MOHID Open source project:
    - All new ideas should be tested via branches
    - Good and tested ideas should generate a new module

- It is easy to add a new module to MOHID
    - The litter module being develop by Suez and Hidromod is a good example of “new idea => new module”

OBRIGADO!

✉ paulo.chambel@hidromod.com