

# Documentation implementation de la plateforme MAELIA

Maroussia Vavasseur

8 juillet 2013

## Introduction

Ce présent document présente dans les grandes lignes les choix d'implémentations de la plateforme MAELIA. Il a pour but de faciliter la prise en main du code pour d'éventuels développeurs. L'implémentation de la plateforme a été pensée pour être le plus modulaire possible. Ainsi, il est possible de ne lancer qu'une partie du modèle complet (le modèle agricole par exemple). De plus, il est possible de choisir parmi différentes implémentations d'un même modèle (pour la croissance des plantes par exemple). Les différents modules indépendants de la plateforme vont être présentés en détail ainsi que la manière dont ces modules communiquent entre eux dans le cas où la simulation est lancée sur la totalité du modèle. Quatre grands modules peuvent être lancés indépendamment. Ils sont hiérarchisés sous forme de package regroupant tout les fichiers (ou classes) gaml appartenant au même modèle :

- Package « main » (13 %) <sup>1</sup>
- Package « modeleCommun » (14 %) <sup>1</sup>
- Package « modeleAgricole » (36 %) <sup>1</sup>
- Package « modeleHydrographique » (27 %) <sup>1</sup>
- Package « modeleNormatif » (10 %) <sup>1</sup>
- Package « testsUnitaires » (? %) <sup>1</sup>

Qu'ils soient lancés seuls ou ensembles, ces modules sont toujours lancés à partir du package « main » (à l'exception du package « testsUnitaires » qui est indépendant). Les différentes classes gaml des 6 packages vont être présentés dans l'ordre alphanumérique.

---

1. Au moment de la rédaction de ce document, la plateforme MAELIA possède plus de 16000 lignes de codes. Les pourcentages donnent une idée de l'importance des différents packages.

# Table des matières

Introduction . . . . .	1
Package « main » . . . . .	5
Experiment.gaml . . . . .	6
main.gaml . . . . .	8
Package « modeleCommun » . . . . .	20
commune.gaml . . . . .	21
contourZoneMaelia.gaml . . . . .	23
dateCourante.gaml . . . . .	24
donneesGlobales.gaml . . . . .	27
ecritureFichiers.gaml . . . . .	28
palettesCouleurs.gaml . . . . .	29
timeStamp.gaml . . . . .	30
typeDeSol.gaml . . . . .	31
zoneMeteo.gaml . . . . .	33
zoneMeteoMoyenne.gaml . . . . .	34
Package « modeleAgricole » . . . . .	35
Agriculteurs / agriculteur.gaml . . . . .	36
Agriculteurs / agriculteurComplexe.gaml . . . . .	40
Agriculteurs / agriculteurJeuOC.gaml . . . . .	42
ITKs / itk.gaml . . . . .	43
ITKs / itkComplexe.gaml . . . . .	44
ITKs / itkJeuOC.gaml . . . . .	45
PlansAssolement / planAssolement.gaml . . . . .	46
StrategiesIrrigation / strategieIrrigation.gaml . . . . .	47
StrategiesIrrigation / strategieIrrigationComplexe.gaml . . . . .	48
StrategiesIrrigation / strategieIrrigationJeuOC.gaml . . . . .	49
StrategiesRecolte / strategieRecolte.gaml . . . . .	50
StrategiesRecolte / strategieRecolteComplexe.gaml . . . . .	51
StrategiesRecolte / strategieRecolteJeuOC.gaml . . . . .	52
StrategiesSemis / strategieSemi.gaml . . . . .	53
StrategiesSemis / strategieSemiComplexe.gaml . . . . .	54
StrategiesSemis / strategieSemiJeuOC.gaml . . . . .	55
SystemesDeCultures / systemeDeCulture.gaml . . . . .	56
SystemesDeCultures / systemeDeCultureComplexe.gaml . . . . .	57
SystemesDeCultures / systemeDeCultureJeuOC.gaml . . . . .	58
clcRPG.gaml . . . . .	59
Cultures / culture.gaml . . . . .	60
Cultures / cultureIrrigable.gaml . . . . .	62
especeCultivee.gaml . . . . .	64
exploitation.gaml . . . . .	65

Ilots / ilot.gaml . . . . .	66
Ilots / ilotHorsZone.gaml . . . . .	69
marcheAgricole.gaml . . . . .	70
Parcelles / parcelle.gaml . . . . .	71
Parcelles / parcelleJeuOC.gaml . . . . .	75
Parcelles / parcelleHorsZone.gaml . . . . .	76
strategieSIR.gaml . . . . .	77
Package « modeleHydrographique » . . . . .	78
barrage.gaml . . . . .	79
clc.gaml . . . . .	81
coursDeau.gaml . . . . .	83
equipement.gaml . . . . .	84
equipementDeCaptage.gaml . . . . .	86
equipementDeCaptageAEP.gaml . . . . .	87
equipementDeCaptageIND.gaml . . . . .	88
equipementDeCaptageIRR.gaml . . . . .	89
equipementDeRejet.gaml . . . . .	90
equipementDeRejetAEP.gaml . . . . .	91
equipementDeRejetIND.gaml . . . . .	92
grosseIndustrie.gaml . . . . .	93
hru.gaml . . . . .	94
hruRPG.gaml . . . . .	97
lac.gaml . . . . .	99
mnt.gaml . . . . .	100
nappePhreatique.gaml . . . . .	101
noeudHydrographique.gaml . . . . .	102
ressourceEnEau.gaml . . . . .	103
retenueCollinaire.gaml . . . . .	105
zoneHydrographique.gaml . . . . .	106
zoneHydrographiqueSWAT.gaml . . . . .	111
zoneHydrographiqueJeuOC.gaml . . . . .	113
Package « modeleNormatif » . . . . .	114
distributeurDeauBrute.gaml . . . . .	115
distributeurDeauPotable.gaml . . . . .	116
gestionnaire.gaml . . . . .	117
gestionnaireDeBarrage.gaml . . . . .	118
organismeUnique.gaml . . . . .	119
pointDeReference.gaml . . . . .	120
pointDeReferenceCalibration.gaml . . . . .	122
policeDeLeau.gaml . . . . .	123
prefet.gaml . . . . .	124
secteurAdministratif.gaml . . . . .	125
uniteDeDefinitionDuVP.gaml . . . . .	126
uniteDeGestion.gaml . . . . .	127
zoneAdministrative.gaml . . . . .	128
Package « main » . . . . .	131
launcher.gaml . . . . .	132
testModeleAgricole.gaml . . . . .	133

# Table des figures

1	Exemple d'experiment . . . . .	6
2	Bloc d'initialisation complet . . . . .	9
3	Bloc d'initialisation du module commun . . . . .	10
4	Bloc d'initialisation du module hydrographique . . . . .	11
5	Bloc d'initialisation du module agricole . . . . .	12
6	Bloc d'initialisation du lien entre les modules hydrographique et agricole . . . . .	13
7	Bloc d'initialisation du module normatif . . . . .	14
8	Bloc d'ordonnement du modèle complet . . . . .	16
9	Bloc d'ordonnement des processus se réalisant en début d'année . . . . .	17
10	Bloc d'ordonnement des processus se réalisant en fin d'année . . . . .	18
11	Bloc d'ordonnement des processus se réalisant chaque jour . . . . .	19

## Package « main »

- Experiment.gaml
- main.gaml

## Experiment.gaml

Il y a autant de classes experiment qu'il y a de modèles prédéfinie (par exemple, il y a une classe experiment qui ne va simuler que le modèle agricole). C'est à partir de ces classes que se lance le/les modèles. C'est ici que se trouvent le seul bloc « init » ainsi que le seul « reflex » GAMA de toute la plateforme. C'est également là qu'est défini le contour d'affichage de la simulation (on utilise l'agent contourZoneEtude), à l'aide du bloc « environment » de GAMA.

Elle fait appel aux 2 actions principales du main, à savoir l'initialisation et l'ordonnanceur (voir plus loin). C'est ici que sont définis les modules à exécuter (par exemple agricole, hydrographique, etc) ainsi que les affichages souhaités dans le mode GUI de la plateforme (agents ou graphiques à afficher lors de l'exécution). Il est très facile de rajouter un experiment à GAMA, et ainsi définir une nouvelle simulation. Pour se faire il suffit de jouer avec les paramètres de la simulation présentés dans le document « [/Tasks/T8. Documentation MAELIA\\_GAMA/T8.2. Plateforme/ManuelUtilisateur.pdf](#) ».

```
experiment ModeleCompleat type: gui {
  parameter 'executerModeleHydrographique' : ' var: executerModeleHydrographique among: [true];
  parameter 'nomChoixModeleHydrographique' : ' var: nomChoixModeleHydrographique among: ['SWAT'];
  parameter 'executerModeleAgricole' : ' var: executerModeleAgricole among: [true];
  parameter 'nomChoixModeleCroissancePlante' : ' var: nomChoixModeleCroissancePlante among: ['Jeu0C'];
  parameter 'executerModeleNormatif' : ' var: executerModeleNormatif among: [true];

  output {
    display evolutionEcoulementEauZH{ //type: opengl autosave: true
      species zoneHydrographiqueSWAT aspect: evolutionDebitAspect;
      species zoneAdministrative aspect: nbJoursRestrictionAspect transparency: 0.3;
      species barrage aspect: evolutionDebitLache;
    }
    display prelevementSvsRJournalierZoneMaelia refresh_every: 1 {
      chart name: 'prelevement Journalier Zone Maelia' type: series background: rgb('white') {
        data 'Prelevements Souhaites' value: volumePrelevementSouhaitesZoneMaelia style: line color: rgb('blue');
        data 'Prelevements Reels' value: volumePrelevementReelsZoneMaelia style: line color: rgb('orange');
      }
    }
  }
}
```

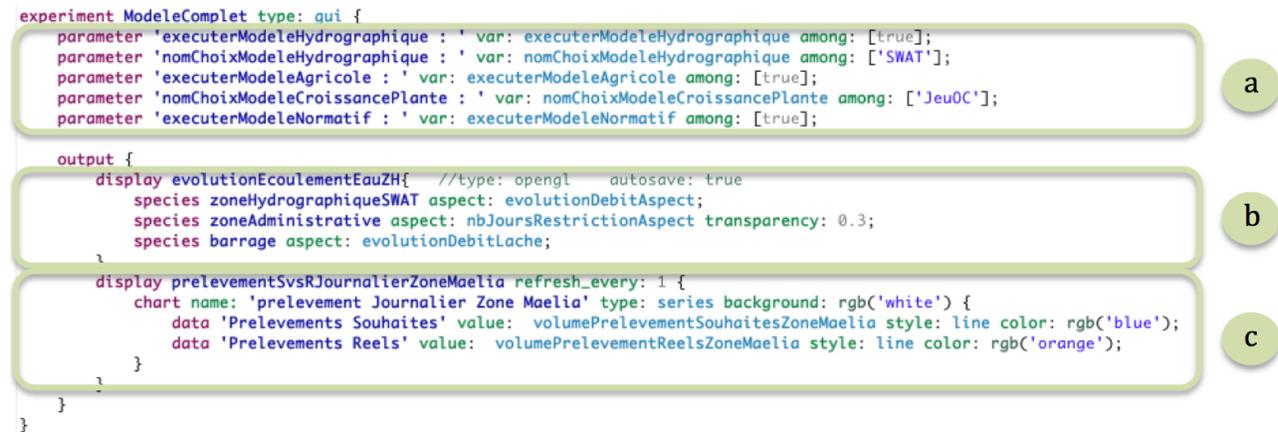


FIGURE 1 – Exemple d'expérience

### bloc a

Les paramètres inscrits ici indiquent quels modèles vont être pris en compte lors de l'exécution. Pour cet exemple, le modèle hydrographique est pris en compte, et le modèle choisi est « SWAT » (si on ne spécifie pas de modèle, le modèle simple sera pris par défaut). De même, le modèle agricole est pris en compte, et le modèle de croissance des plantes choisi est le modèle « Jeu d'OC » (là encore, si on ne spécifie pas de modèle, le modèle simple sera pris par défaut). Enfin, le modèle normatif est également pris en compte. Pour ce dernier pas de choix possible entre différents sous modèles pour le moment.

### bloc b

Ce bloc représente une des vues qui sera affichée lors de l'exécution du modèle. Cette vue affiche de manière superposée les agents géo-référencés de type zoneHydrographiqueSWAT, zoneAdministrative ainsi que barrage. Chacun de ces agents peuvent avoir plusieurs affichages différents implémentés. Dans cet exemple il a été choisi d'afficher l'aspect « écoulement de l'eau » des zoneHydrographique, l'aspect « nombre de jour de restriction » des zoneAdministrative et l'aspect « évolution du débit de lâcher » du barrage. Il est tout à fait possible de définir plusieurs onglets « display » avec d'autres types d'agents, ou d'autre aspect prédéfinis pour ces mêmes agents. Certains aspects prédéfinis sont dynamiques, d'autres non.

### **bloc c**

Ce bloc représente une des vues qui sera affichée lors de l'exécution du modèle. Ici, ce ne sont pas des agents géo-référencés qui s'affichent mais des graphiques dynamiques. Dans cet exemple, 2 courbes montrant l'évolution des volumes de prélèvement souhaités et réels au niveau de la zone d'étude vont s'afficher dans le même graphique. Il est possible d'afficher des résultats globaux (comme pour cet exemple) ou des variables propres à une instance d'agent.

## main.gaml

---

C'est à partir du main que sont créés et initialisés tous les agents de la simulation appartenant à un des 4 packages cités plus tôt. C'est ici aussi qu'est défini l'ordonnancement de la dynamique du modèle, c'est à dire quel processus s'exécute quotidiennement ou annuellement et dans quel ordre.

### GLOBAL

#### *initGlobal*

Action publique appelée depuis la classe Experiment, dans le seul bloc « init » GAMA de toute la plateforme. Elle crée et initialise tous les agents de la simulation. L'ordre de création et d'initialisation de chaque agent n'est pas anodin dans la plupart des cas. Afin d'expliquer clairement dans quel ordre sont créés chaque agents, voici différents diagrammes d'activité.

Remarque : Construction = création + initialisation d'une espèce d'agent.

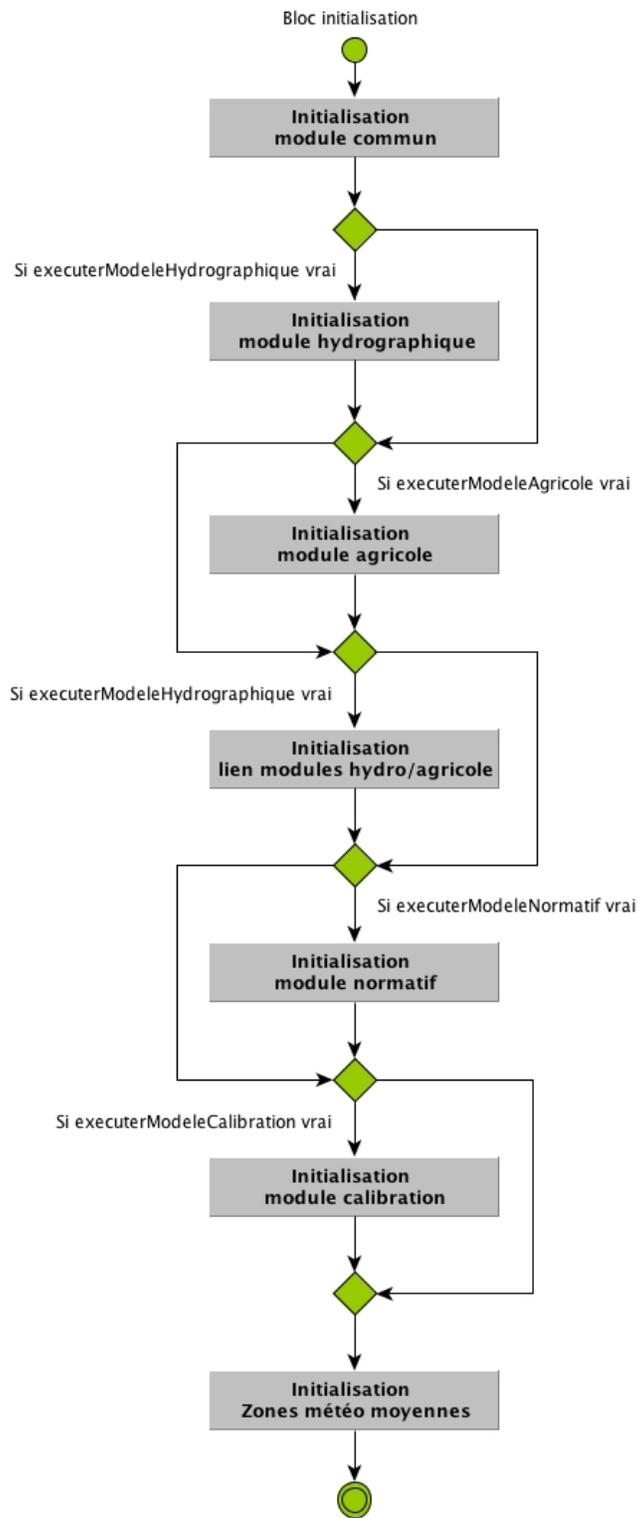


FIGURE 2 – Bloc d'initialisation complet

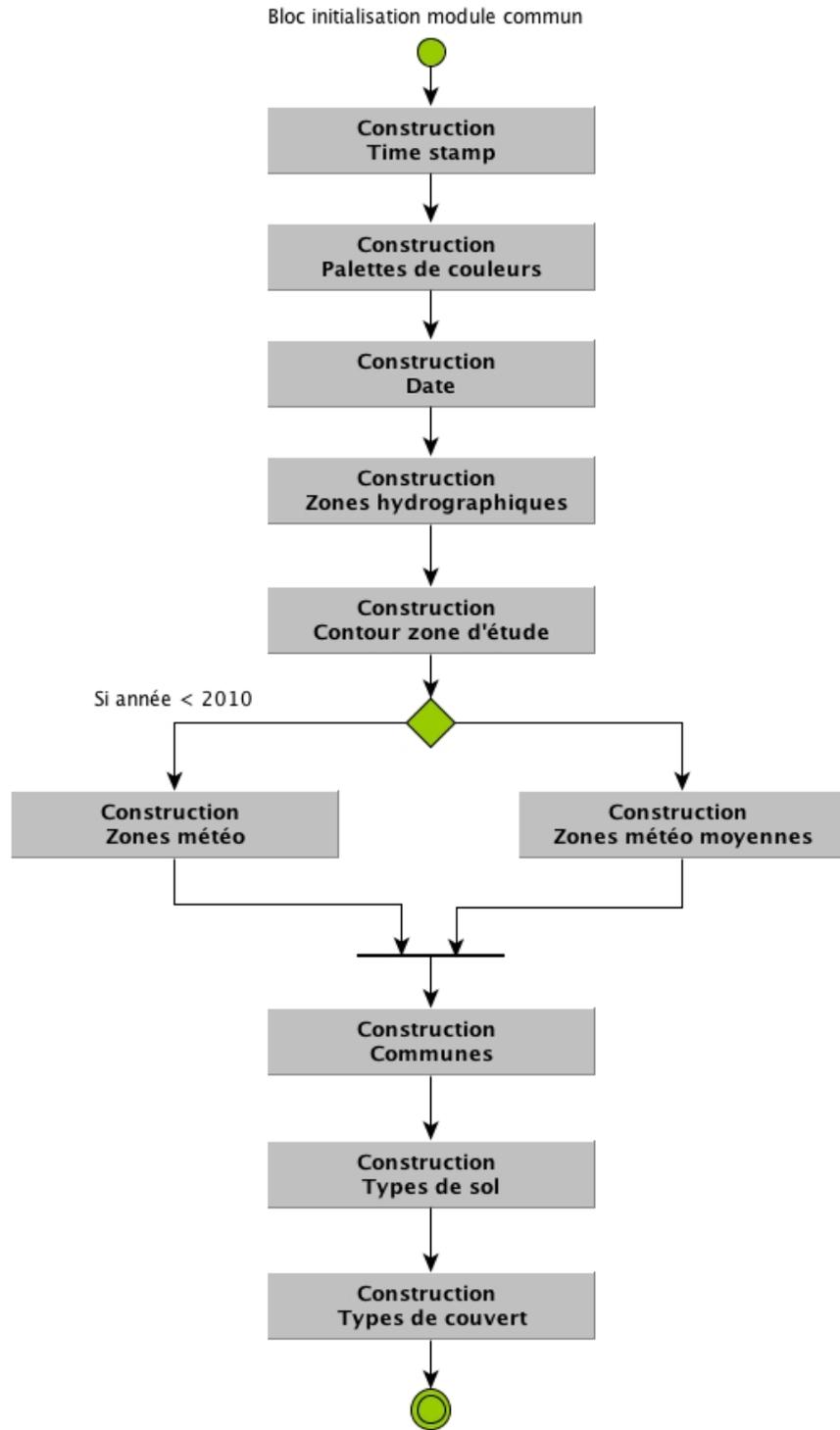


FIGURE 3 – Bloc d'initialisation du module commun

Bloc initialisation module Hydrographique

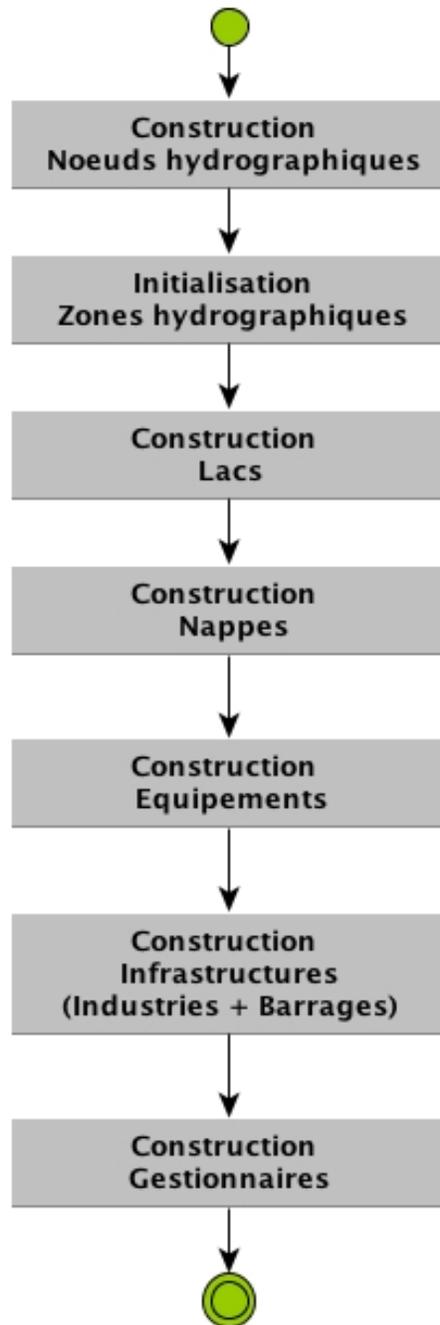


FIGURE 4 – Bloc d'initialisation du module hydrographique

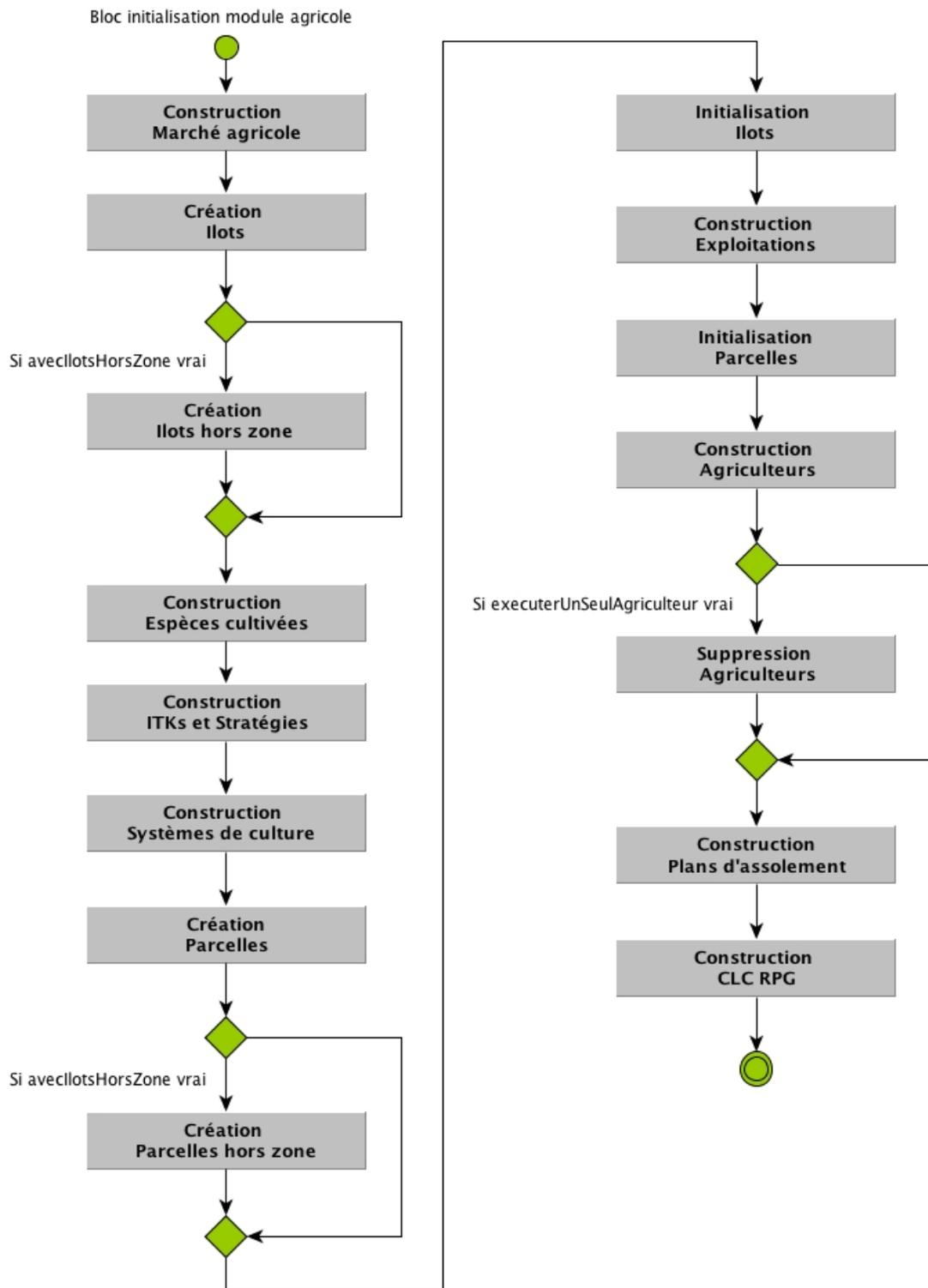


FIGURE 5 – Bloc d'initialisation du module agricole

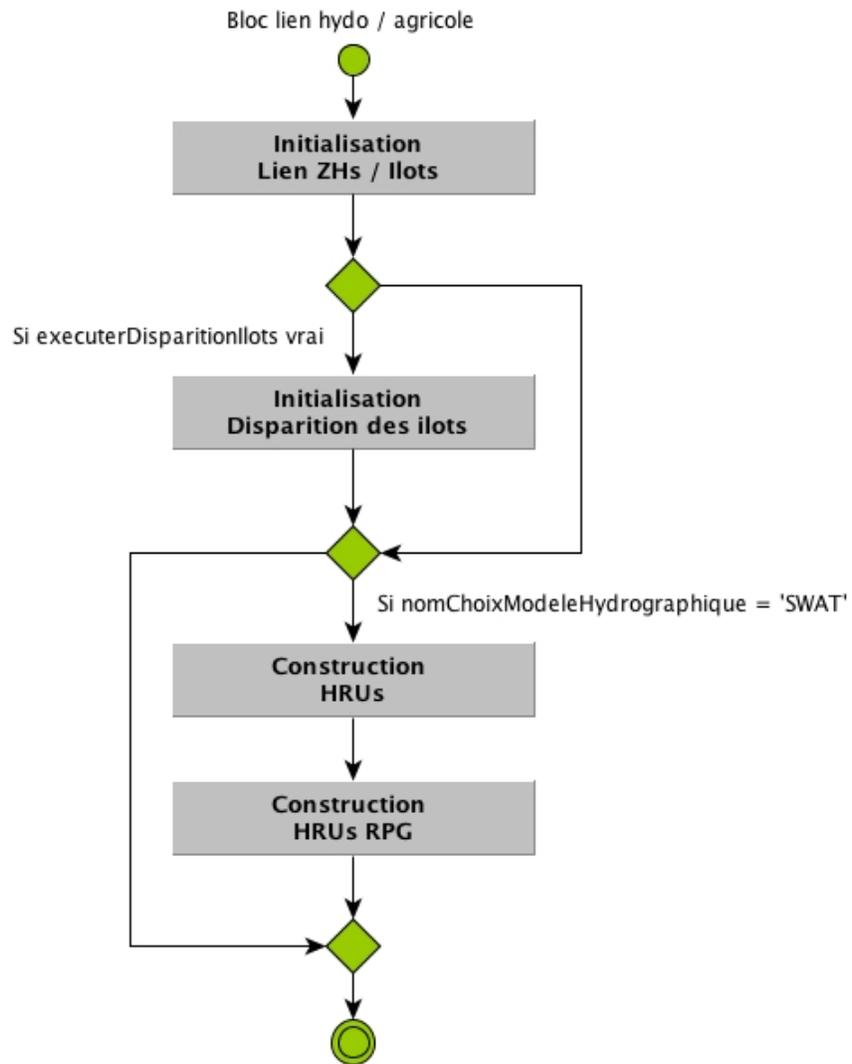


FIGURE 6 – Bloc d'initialisation du lien entre les modules hydrographique et agricole

Bloc initialisation module Normatif

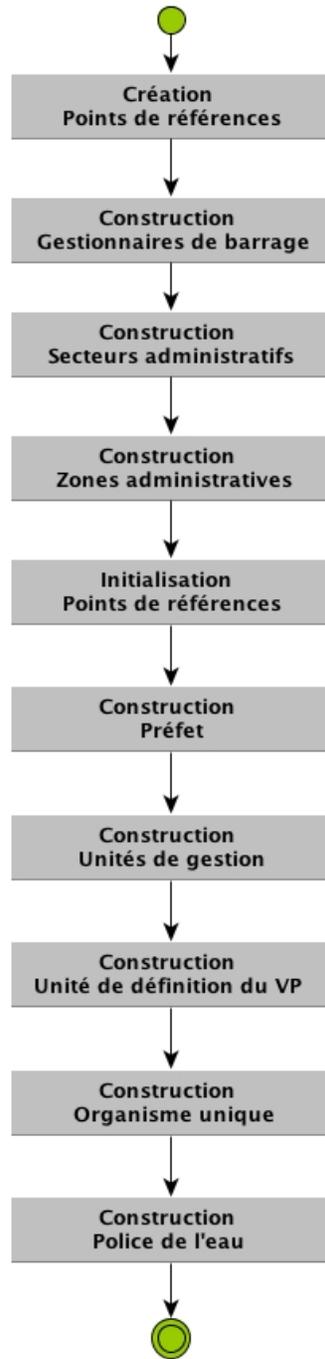


FIGURE 7 – Bloc d'initialisation du module normatif

*reflexGlobal*

Action publique appelée depuis la classe Experiment, dans le seul bloc « reflex » GAMA de toute la plateforme. Elle ordonnance tous les processus du modèle, voici différents diagrammes d'activité.

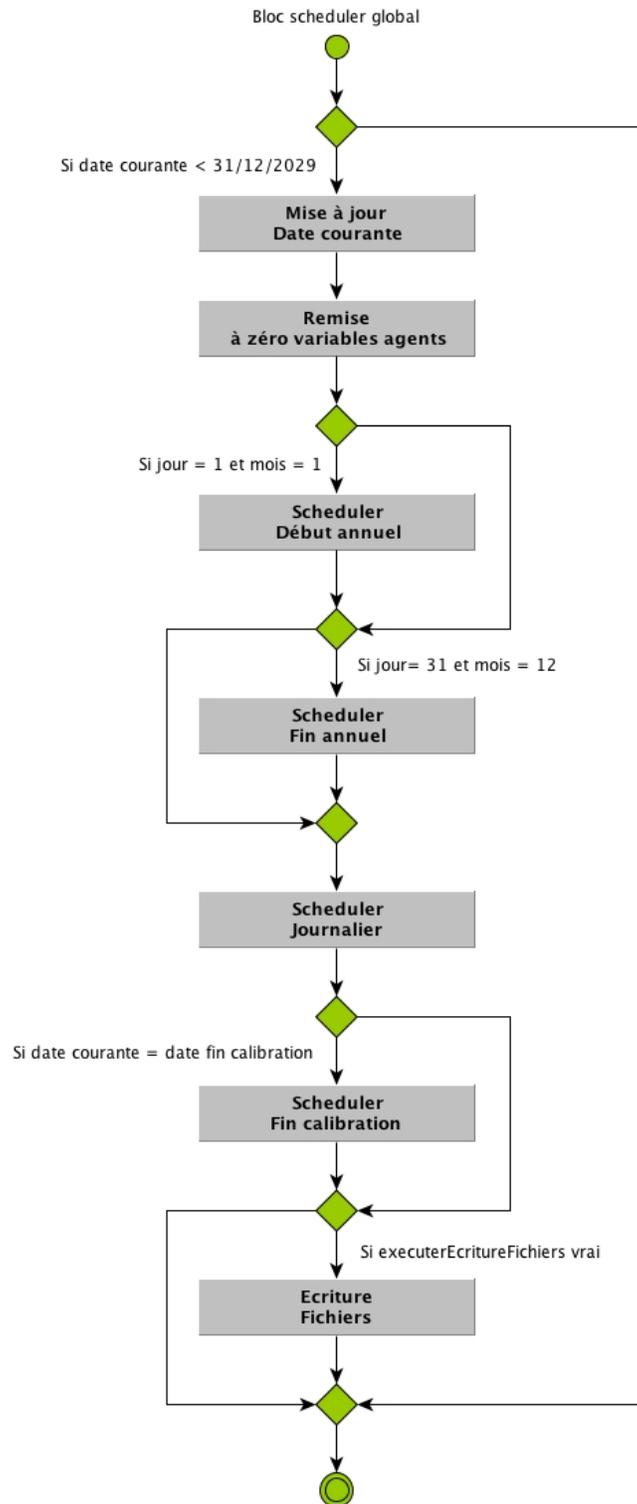


FIGURE 8 – Bloc d'ordonnancement du modèle complet

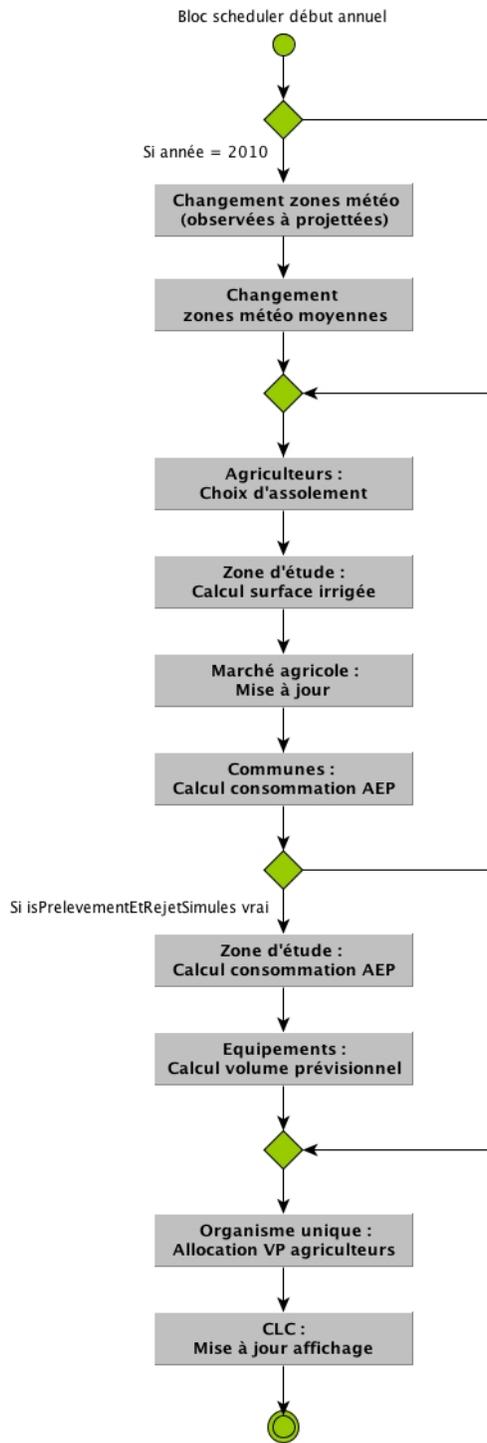


FIGURE 9 – Bloc d'ordonnancement des processus se réalisant en début d'année

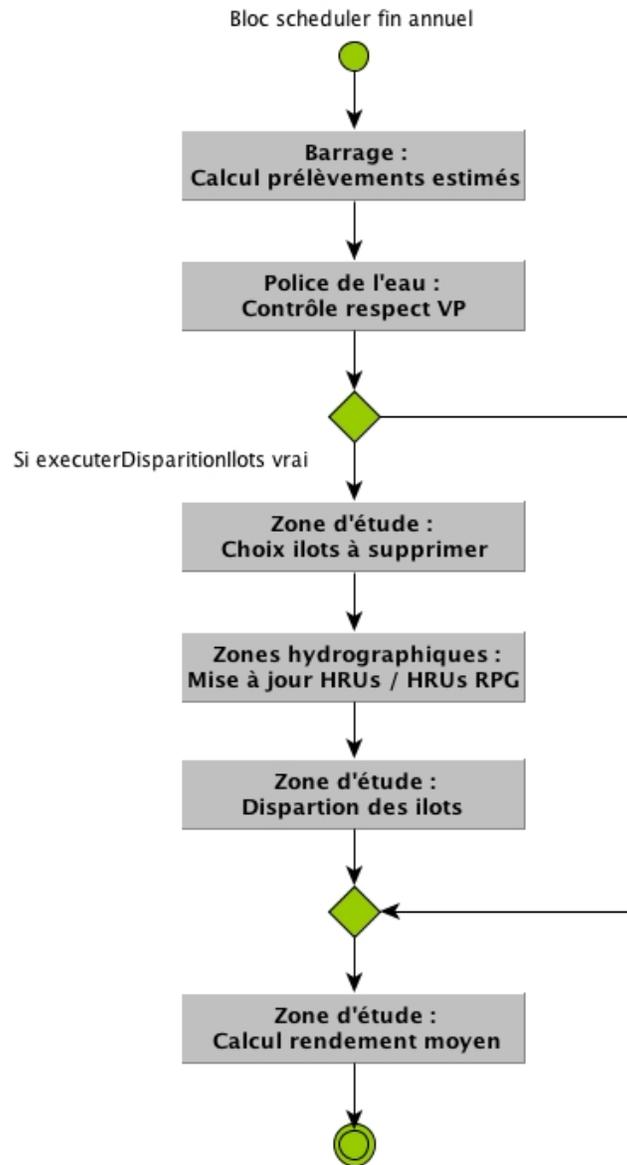


FIGURE 10 – Bloc d'ordonnancement des processus se réalisant en fin d'année

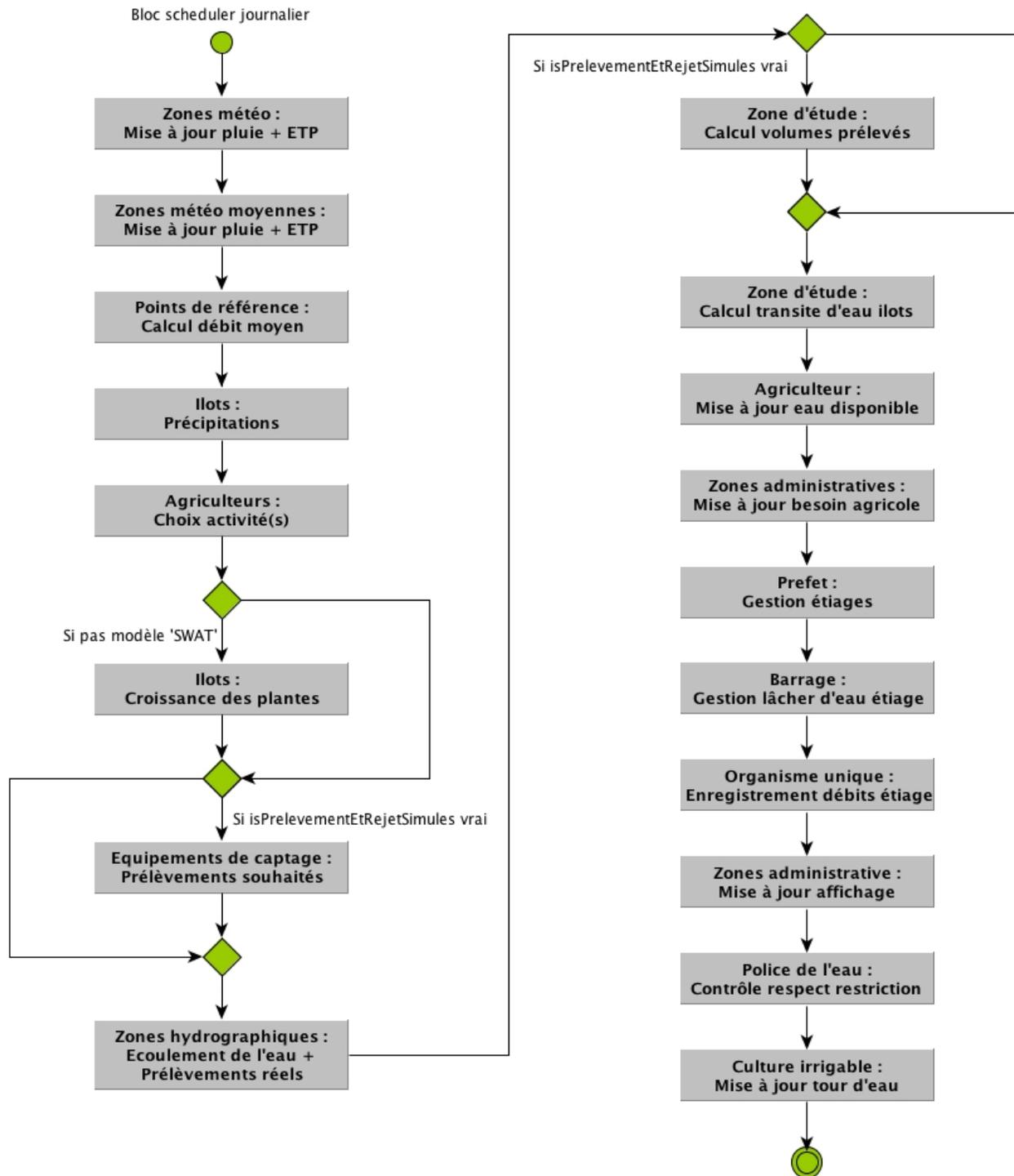


FIGURE 11 – Bloc d'ordonnancement des processus se réalisant chaque jour

## Package « modeleCommun »

- commune.gaml
- contourZoneMaelia.gaml
- dateCourante.gaml
- donneesGlobales.gaml
- ecritureFichiers.gaml
- palettesCouleurs.gaml
- timeStamp.gaml
- typeDeSol.gaml
- zoneMeteo.gaml
- zoneMeteoMoyenne.gaml

## **commune.gaml**

---

### GLOBAL

#### **Attributs**

- communesShape - Chemin vers le fichier SHP
- cheminPopulationCommunes - Chemin vers le fichier CSV
- cheminPrixEauCommunes - Chemin vers le fichier CSV
- cheminSalaireCommunes - Chemin vers le fichier CSV
- cheminResidencePrincipaleCommunes - Chemin vers le fichier CSV
- volumeEauPotableConsommeeZoneMaelia [m3]

#### **constructionCommunes**

Action publique appelée depuis le main à l'initialisation qui crée les agents communes à partir d'un fichier SIG et les initialise à partir de 3 fichiers csv.

#### **initialisationUGCommunes**

Action publique appelée depuis le main qui associe l'uniteDeGestion à la commune. Elle est faite séparément car elle doit être appelée après la création des agents uniteDeGestion.

#### **calculVolumeAEPconsommeZoneMaelia**

Action publique appelée annuellement depuis le main qui calcule le volume d'eau potable consommé par la totalité de la zone MAELIA. Ce volume est ensuite utilisé par les agents pointsDePrelevements pour calculer les volumes prélevés quotidiens.

### AGENT

#### **Attributs**

- couleurCommune
- codeInsee
- populationAnneeCourante
- mapPopulationParAnnee
- listeCultureIrrigeesPasDeTempsCourant [culturesIrrigables]
- isPeutPreleverCommune
- uniteDeGestionAssociee [uniteDeGestion]
- isCommuneEnRestriction
- prixEauPotable
- salaireMoyen
- tauxResidencesPrincipales
- mapConsomationEauPotable
- zoneMeteoAssociee [zoneMeteo]
- zoneAdministrativeAssociee [zoneAdministrative]

#### **comportementAnnuel**

Action appelée annuellement depuis le main.

#### **agmentationDeLaPopulation**

Action appelée depuis l'action comportementAnnuel. Elle met à jour la population de la commune grâce aux données lues à l'initialisation dans un csv.

#### **consomationEauPotableMogire**

Action appelée depuis l'action comportementAnnuel. Elle calcule la consommation annuelle de la commune

à l'aide du modèle Mogire.

***isCommuneEnBesoinAgricoleFort***

Action appelée depuis l'agent secteurAdministratif dont l'agent commune fait parti. Elle met à jour un booléen en fonction du nombre d'agent culture lui étant associé qui ont atteint un stade phénologique avancé.

***applicationRestrictionCommune***

Action appelée depuis l'agent secteurAdministratif dont l'agent commune fait parti. Elle met à jour les jours de la semaine où ses agents culture associés ne pourront pas être irrigués.

***basic***

Action d'affichage SIG.

***toString***

Action d'affichage des variables de l'agent dans la console.

## **contourZoneMaelia.gaml**

---

Cette classe gaml représente le contour de la zone d'étude. Il peut correspondre soit à une seule zoneHydrographique, un agrégat de zoneHydrographique ou la zone MAELIA complète.

### **GLOBAL**

#### ***constructionContourZoneMaelia***

Action publique appelée depuis le main à l'initialisation qui crée les agents contourZoneMaelia à partir d'un fichier SIG.

### **AGENT**

#### ***basic***

Action d'affichage SIG appelée dans la classe gaml Experiment.

## dateCourante.gaml

Cette classe gaml gère tous les aspects liés à la date de la simulation. Les 2 attributs principaux sont : idDate (ou identifiant date, entier de la forme JJMMAAAA) et indiceDate (entier incrémentale pour chaque pas de temps de la simulation, ou le pas de temps est la journée) : il ne commence pas à zéro mais à un entier égale au nombre de jour écoulé depuis la date "zéro" définie en dure (le 01/01/1949).

### GLOBAL

#### **Attributs**

- initDataDate - Chemin vers le fichier CSV
- nbHeureDansUneJournee
- nbSecondesDansUneHeure
- nbJoursDansUneAnneeNonBissextile
- nbAnneesDansCycleIntercalaire
- nbJoursDansUnCycleIntercalaire
- nbJoursDansUneSemaine
- jourZero
- moisZero
- anneeZero
- indiceJourDeLaSemaineDateZero
- indiceColoneAnneeBissextile
- indiceColoneAnneeNonBissextile
- mapJoursDeLaSemaine
- indiceDateDebutEte
- indiceDateFinEte
- indiceDateFinSimulation
- indiceDateDebutSimulation
- indiceDateFinCalibration
- timeStampPremierJourSimulation
- indexDateCultureHiver
- indexDateFinFrein
- date [dateCourante]

#### **constructionDateCourante**

Action publique appelée depuis le main à l'initialisation qui crée l'agent dateCourante.

#### **calculIndiceDatesDivers**

Action publique appelée l'action globale constructionDateCourante. Elle initialise certaines dates clés (par exemple la date de début et de fin de simulation, etc.) dans un format exploitable par la plateforme (sous la forme d'un indice).

### AGENT

#### **Attributs**

- idDate
- jour
- mois
- annee
- indiceDate
- indiceDateInitial
- nbJoursEcoulesDansAnnee

- indiceJourDeLaSemaine
- nombreJourDansAnneeCourante
- isAnneeBissextile
- longueurDuJour

### ***comportementJournalier***

Action appelée en début de chaque pas de temps depuis le main. Elle met à jour la date courante.

### ***initialisationDateCourante***

Action appelée depuis l'action globale constructionDateCourante. Elle initialise entre autre le jour, le moi, l'année et l'indice.

### ***miseAJourDateCourante***

Action appelée depuis l'action interne comportementJournalier. Elle met à jour l'indice de la date en utilisant l'attribut propre à Gama : le time.

### ***majJourDeLaSemaine***

Action appelée depuis l'action interne comportementJournalier. La formule se base sur la date du jour zéro, le 01/01/1952 est un mardi. Cela équivaut a dire que l'indiceDate 0 équivaut a l'indice jour de la semaine 2. Partant de là il suffit de faire un modulo (+1) sur l'indiceDate a chaque pas de temps en soustrayant a l'indiceDate le nombre de jour entre l'indiceDate 0 et l'indiceDate où on est pour la première fois un lundi (pour nous il y a 6 jours). ATTENTION : Ce calcul ne marche donc pas pour les 6 premiers, donc du 01/01/1952 au 06/01/1952

### ***calculNbJourAnneeCourante***

Action appelée dans les méthodes internes initialisationDateCourante et miseAJourDateCourante. Elle met à jour le nombre de jour de l'année courante (2 possibilités selon que l'année est bissextile ou non).

### ***calculNbJourEcouleDansAnnee***

Action appelée chaque fois que l'on souhaite connaître le nombre de jours écoulés entre le début de l'année (le premier janvier) et une date donnée en entrée.

### ***calculNbJourEcouleDansAnneeAlaDateCourante***

Action appelée chaque fois que l'on souhaite connaître le nombre de jours écoulés entre le début de l'année (le premier janvier) et la date courante.

### ***calculNbJourEcouleEntreDeuxDates***

Action appelée chaque fois que l'on souhaite connaître le nombre de jours écoulés entre 2 dates données en entrée.

### ***isAnneeBissextile***

Action appelée chaque fois que l'on souhaite savoir si l'année courante est une année bissextile ou non.

### ***isDateCourantEntreDeuxDates***

Action appelée chaque fois que l'on souhaite savoir si la date courante est comprise entre 2 dates données en entrée.

### ***convertirIndiceEnDate***

Action appelée dans l'action interne miseAJourDateCourante. Elle convertie l'indice de la date en date (jour, mois, année).

***convertirDateEnIndice***

Action qui convertie le jour, le mois et l'année donnés en entrée par l'indice de la date.

***convertirIdDateEnIndice***

Action qui convertie l'identifiant de la date (JJMMAAAA) donné en entrée par l'indice de la date.

***basic***

Action d'affichage SIG appelée dans la classe gaml Experiment.

***toString***

Action d'affichage des variables de l'agent dans la console. Cette action affiche la date complète ainsi que le cycle courant et le temps depuis le début de la simulation.

***toStringLight***

Action d'affichage des variables de l'agent dans la console. Affichage un peu plus épuré (date et temps écoulé).

## **donneesGlobales.gaml**

---

Il n'y a pas d'action, ni d'agent défini dans cette classe. Elle regroupe tous les paramètres et constantes propres à la simulation (cf. première et deuxième partie du document).

## **écritureFichiers.gaml**

---

Cette classe gaml est dédiée à l'écriture de données simulées dans des fichiers de sorties (principalement des csv pour le moment).

### **GLOBAL**

#### ***initialisationEcritureFichiers***

Action publique appelée depuis le main à l'initialisation qui crée les fichiers (et plus spécifiquement les entêtes de chaque colonne des cvs créés) où l'on souhaite enregistrer des données de sortie.

#### ***écritureFichiers***

Action publique appelée depuis le main chaque pas de temps qui enregistre les données souhaitées dans les fichiers.

## palettesCouleurs.gaml

---

Initialise des map contenant les dégradés de couleur en fonction de la valeur potentielle que va prendre une entités de la simulation grâce à des images d'entrée.

### GLOBAL

#### *constructionPalettesCouleurs*

Action publique appelée depuis le main à l'initialisation qui crée les différentes palettes de couleurs utilisées pour les affichage de certains agents (les débits de l'agent zoneHydrographique par exemple). Ces palettes sont initialisées à partir d'images png.

## **timeStamp.gaml**

---

Classe gaml qui va permettre l'affichage dans la console du temps (en secondes) écoulé depuis le lancement de la simulation.

### GLOBAL

#### ***constructionTimeStamp***

Action publique appelée depuis le main à l'initialisation qui crée l'agent timeStamp.

#### ***ecritureConsolePourDebug***

Action publique appelée depuis le main à chaque création et / ou initialisation d'une espèce d'agent. Elle va écrire dans la console la chaîne de caractères donnée en entrée avec le temps écoulé (en seconde) depuis le début de l'initialisation.

### AGENT

#### ***initialisationtimeStamp***

Action appelée dans constructionTimeStamp qui appelle une méthode java qui va créer le timeStamp.

#### ***getTimeStampLocale***

Action qui met à jour le temps écoulé (en ms) depuis le début de la simulation, temps d'initialisation non compris.

#### ***getTimeStamp***

Action qui renvoie le temps écoulé (en seconde) depuis le début de l'exécution de la simulation, temps d'initialisation compris.

## typeDeSol.gaml

---

Le type de sol est associé par parcelle, pour le modèle agricole et est associé par HRU pour le modèle hydrographique (et plus précisément pour le modèle SWAT).

### GLOBAL

#### *Attributs*

- typeDeSolShape - chemin vers le fichier SHP
- cheminTypesDeSol - chemin vers le fichier CSV

#### *constructionTypeDeSol*

Action publique appelée depuis le main à l'initialisation qui crée et initialise les agents typeDeSol.

### AGENT

#### *Attributs*

- idSHP
- idZhAssociee
- idTypeDeSol
- profondeurMax
- reservePotentielleUtileSHP
- reserveFacilementUtilisableMaximum
- tauxArgile
- profondeurMaxEnMm [mm] - SWAT
- mapProfondeurMinParCouche [mm] - SWAT
- mapProfondeurMaxParCouche [mm] - SWAT
- mapEpaisseurParCouche [mm] - SWAT
- capaciteDuChamp [mm] - SWAT (FC)
- pointFletrissementPermanent [mm] - SWAT (WP)
- capaciteEauDisponible [mm] - SWAT (AWC)
- densiteGravier [mg.m-3] - SWAT
- tauxCapaciteDuChamp [mm] - SWAT (FC)
- tauxPointFletrissementPermanent [mm] - SWAT (WP)
- tauxCapaciteEauDisponible [mm] - SWAT (AWC)
- tauxSaturation [mm] - SWAT (SAT)
- conductiviteHydroliqueSaturee [mm/hr] - SWAT (Ksat)
- stuDominant - SWAT
- reservoirHorizonTravailProfond [mm] - JeuOC (RUw)
- reservePotentielleUtileMax [mm] - JeuOC (RUM)
- doseIrrigationMax [mm] - JeuOC (dosIm)
- coefEfficaciteAzote - JeuOC (efN)
- coefStabiliteCultural - JeuOC (alpha)
- coefCC - JeuOC
- ctr\_m - JeuOC
- tauxGravier - JeuOC
- horizonDeTravailProfond [cm] - JeuOC (W3)
- teneurEnMatiereOrganique - JeuOC (%MO)
- noteQualiteStructureSol - JeuOC (Cstru3)
- permeabiliteSol [mm] - JeuOC (PIRm)
- mineralisationNannuelle [kgN/ha] - JeuOC (minNA)
- coefLimitantDoseIrr - JeuOC

- pente - JeuOC

***initialisationTypeDeSol***

Action appelée dans `constructionTypeDeSol` réalisant l'initialisation de l'agent `typeDeSol`. Elle va, entre autre, créer des couches dans le sol en vue de l'application du modèle SWAT.

***colorationReserveUtile***

Action appelée dans `initialisationTypeDeSol` qui va initialiser une couleur à l'agent en fonction de la valeur de sa réserve utile.

***reserveUtileAffichage***

Action d'affichage SIG appelée dans la classe `gaml Experiment`.

## **zoneMeteo.gaml**

---

Une zone météo se définit comme un polygone autour des points météo données par Météo France (SA-FRAN) ; données observées (12km\* 8km). Ainsi, sur la zone d'étude, les zones météo font une grille recouvrant toute la zone. En 2010, les données observées sont remplacées par les données projetées d'ARPEGE. Dans ce cas les polygones n'ont plus la même taille (8 km \* 8 km).

### GLOBAL

#### ***constructionZoneMeteo***

Action publique appelée depuis le main à l'initialisation qui crée et initialise les agents zoneMeteo.

#### ***changementZonesMeteo***

Action publique appelée depuis le main dès que l'année 2010 est atteinte dans la simulation. Elle va faire le changement des zones météo. Entre 2000 et 2009, ces zones correspondent aux données observées. Passé 2010, les zones météo changent pour être appliquées aux données projetées.

### AGENT

#### ***comportementJournalier***

Action appelée à chaque pas de temps depuis le main.

#### ***miseAJourDonnees***

Action appelée depuis l'action comportementJournalier. Elle va mettre à jour les 4 variables importantes de la météo ; à savoir la température minimale, la température maximale, l'ETP et la hauteur des précipitations. Ces données ont été récupérées par la lecture d'un fichier csv à l'initialisation et sont stockées sous la forme d'une « map » par jour.

#### ***liste\_pluies***

Action appelée principalement depuis les agents strategieSIR. Elle renvoie la listes des hauteurs de pluie tombées depuis le nombre de jour passé en paramètres.

#### ***pluies\_periode***

Action appelée principalement depuis les agents strategieSIR. Elle renvoie la hauteur de pluie tombée entre 2 dates données en entrée.

#### ***ETP\_periode***

Action appelée principalement depuis les agents strategieSIR. Elle renvoie la somme des ETP entre 2 dates données en entrée.

#### ***coloration***

Action appelée depuis l'action comportementJournalier. Elle colore l'agent en fonction de la quantité d'eau qu'il a plu.

#### ***basic***

Action d'affichage SIG appelée dans la classe gaml Experiment.

## zoneMeteoMoyenne.gaml

---

Une zone météo moyenne va avoir la même forme qu'une zoneHydrographique. Elle correspond à la météo d'une ZH. Celle-ci correspond à la somme des valeurs connues des zones Météo, interceptant la ZH, ramenée au prorata de l'interception ZH/Zone Météo. Correspond aux données météo d'une zone hydro moyennées en fonction des taux de surface. Ces valeurs évoluent en fonction du temps. L'agent zoneMeteoMoyenne hérite de l'agent zoneMeteo. Il a donc les mêmes actions.

### GLOBAL

#### ***constructionZoneMeteoMoyenne***

Action publique appelée depuis le main à l'initialisation qui crée et initialise les agents zoneMeteo.

#### ***changementZonesMeteoMoyennes***

Action publique appelée depuis le main dès que l'année 2010 est atteinte dans la simulation. Elle va faire le changement des zones météo. Entre 2000 et 2009, ces zones correspondent aux données observées. Passé 2010, les zones météo changent pour être appliquées aux données projetées.

### AGENT

#### ***comportementJournalier***

Action appelée à chaque pas de temps depuis le main.

#### ***processusMiseAJourMeteoMoyenne***

Action appelée depuis l'action comportementJournalier. Elle va mettre à jour les 4 variables importantes de la météo ; à savoir la température minimale, la température maximale, l'ETP et la hauteur des précipitations. Ces données sont calculées en appliquant un prorata (en fonction de l'intersection des polygones zones météo et ZH) aux données des zones météo appartenant à la zoneHydrographique.

#### ***liste\_pluies***

Voir agent zoneMeteo.

#### ***pluies\_periode***

Voir agent zoneMeteo.

#### ***ETP\_periode***

Voir agent zoneMeteo.

#### ***basic***

Action d'affichage SIG appelée dans la classe gaml Experiment.

#### ***toString***

Action d'affichage des variables de l'agent dans la console.

## Package « modeleAgricole »

- Agriculteurs / agriculteur.gaml
- Agriculteurs / agriculteurComplexe.gaml
- Agriculteurs / agriculteurJeuOC.gaml
- ITKs / itk.gaml
- ITKs / itkComplexe.gaml
- PlansAssolement / planAssolement.gaml
- StrategiesIrrigation / strategieIrrigation.gaml
- StrategiesIrrigation / strategieIrrigationComplexe.gaml
- StrategiesRecolte / strategieRecolte.gaml
- StrategiesRecolte / strategieRecolteComplexe.gaml
- StrategiesSemis / strategieSemi.gaml
- StrategiesSemis / strategieSemiComplexe.gaml
- SystemesDeCultures / systemeDeCulture.gaml
- SystemesDeCultures / systemeDeCultureComplexe.gaml
- clcRPG.gaml
- Cultures / culture.gaml
- Cultures / cultureIrrigable.gaml
- especeCultivee.gaml
- exploitation.gaml
- Ilots / ilot.gaml
- Ilots / ilotHorsZone.gaml
- marcheAgricole.gaml
- Parcelles / parcelle.gaml
- Parcelles / parcelleJeuOC.gaml
- Parcelles / parcelleHorsZone.gaml
- strategieSIR.gaml

## Agriculteurs / agriculteur.gaml

L'agriculteur est le seul acteur du modèle agricole. Il va pouvoir avoir un comportement rationnel (agent agriculteurComplexe) ou bien un comportement basé uniquement sur des données d'entrées (agent agriculteurJeuOC). Ce dernier est à revoir.

### GLOBAL

#### **Attributs**

- imageAgriculteur - Chemin vers l'image pour l'affichage
- listeAgriculteurs [agriculteur]
- listeAgriculteursAyantUneParcelleIrrigableMinimum [agriculteur]
- mapNombreTypeCultureGlobale
- mapSurfaceTypeCultureGlobale
- mapRotationTypeCultureGlobale
- mapRendementMoyenZoneMaelia
- surfaceIrrigeeZoneMaelia [m3]

#### **constructionAgriculteur**

Action publique appelée depuis le main à l'initialisation qui crée et initialise les agents agriculteurComplexe ou agriculteurJeuOC (selon le modèle d'agriculteur choisi dans les paramètres de simulation).

#### **suppressionAgriculteursAnePasExecuter**

Action appelée à l'initialisation depuis le main. Il est possible de lancer des simulations uniquement sur un agriculteur choisi dans donneesGlobales (ceci afin d'alléger les temps de calcul, utile surtout pour le debug). Cette action a pour but de supprimer les agriculteurs non souhaités, ainsi que tous les agents dépendants de celui-ci tels que les agents ilot, ilotHorsZone, exploitation et parcelle.

#### **calculRendementMoyenAnneePasseeZoneMaelia [Indicateur]**

Action appelée à la fin de chaque année depuis le main. Elle calcule le rendement moyen de la zone d'étude par type d'espèce cultivée et par année.

#### **calculNbParcellesIrrigees [Indicateur]**

Action normalement appelée à la fin de chaque année depuis le main. Elle permet de connaître le nombre de parcelle irriguées chaque année sur la zone d'étude.

#### **calculSurfaceIrrigeeZoneMaelia**

Action appelée en début de chaque année depuis le main. Elle calcul la surface irriguée totale sur la zone d'étude. Ce calcul est possible une fois que le choix d'assolement a été fait par les agents agriculteur. En effet à partir de là, la culture de l'année à venir est connue. Cette surface est ensuite utilisée par l'agent barrage, pour estimer le volume total de la zone d'étude prélevé chaque année pour l'agriculture.

### AGENT

#### **Attributs**

- taillePointAgriculteur
- couleurAgriculteurRespectRestriction
- couleurEauDisponible
- sonExploitation [exploitation]
- capital
- volumePrelevableAnnuel [m3]
- eau\_disponible [m3]

- heuresRestantesActivite [hr]
- heuresEffectueesActivite [hr]
- nb\_heures\_travails\_min [hr]
- nb\_heures\_travails\_max [hr]
- derniereParcelleTraitee [parcelle]
- dernier\_ilot\_traite [ilot]
- listeParcelles [parcelles]
- listeParcellesHorsZone [parcellesHorsZone]
- nbParcellesIrriguees
- nbParcelles
- surfaceIrrigee [m2]
- mapIlotsIrrigues
- profile
- ilotPrincipale [ilot]
- communeAssociee [commune]
- isAgriculteurIrrigueContreRestriction
- mapVerbalisationsLiesesRestriction
- mapVerbalisationsLiesesRespectVP
- isAafficher
- mapNombreTypeCulture
- mapRendementParEspece
- mapRendementMoyen
- mapDerniereParcelleIrrigeeDuGroupe
- mapGroupesIrrigation
- mapSurfaceTotaleGoupe
- mapSurfaceJournaliereGoupe
- isDebutPeriodeIrrigationPassee
- mapIlotsIrriguesAssociesParZA
- listeParcellesAsemer [parcelles]
- listeParcellesAirriguer [parcelles]
- listeParcellesArecolter [parcelles]

#### ***comportementJournalier***

Action appelée à chaque pas de temps depuis le main. L'agriculteur va faire le choix de ses activités de la journée. Ainsi il va pouvoir semer, irriguer et / ou récolter dans un même pas de temps si ces activités sont possibles et si son temps disponible lui permet.

#### ***comportementAnnuel***

Action appelée chaque année depuis le main. L'agriculteur va entre autre faire son choix d'assolement ainsi que la création de ses groupes d'irrigation tous les premiers janvier.

#### ***initialisationAgriculteur***

Action appelée depuis l'action globale constructionAgriculteur. Elle initialise entre autre la liste des agents parcelle, la commune ou encore la position de l'agriculteur.

#### ***miseAJourVariables***

Action appelée chaque jour depuis le main. Dans le cas où le modèle hydrographique est activé, un traitement pour déterminer quelle quantité d'eau l'agriculteur va pouvoir prélever est effectué au niveau des agents zoneHydrographique. Si aucun modèle hydrographique n'est lancé, on considère que l'agriculteur que l'agriculteur a un volume prélevable infini. Mais dans tout les cas cette action est appelée pour mettre à jour la quantité d'eau disponible pour l'agriculteur.

### ***miseAJourMapNombreTypesCultures***

Action appelée depuis la méthode comportementAnnuel. Cette action stocke dans l'agent et au niveau de la zone d'étude des informations sur le nombre de type de culture, les rotations ainsi que les surface pour chaque culture. Ces données sont potentiellement des indicateurs de sortie que l'on peut en partie stocker dans des fichiers à l'aide de la classe ecritureFichier.

### ***creationGroupesIrrigation***

Action appelée depuis la méthode interne comportementAnnuel, uniquement si le modèle d'irrigation choisi est le complexe. Dans ce cas, l'agriculteur va pouvoir paralléliser l'irrigation des ses parcelles, par la création de un ou plusieurs groupes d'irrigation. La création de ces groupes d'irrigation se fait dans un premier temps zoneAdministrative. Dans le cas où le modèle normatif est pris en compte, il faudra que chaque parcelle d'un groupe appartienne à la même zone de restriction. Ensuite, les groupes sont composés de parcelles proches les unes des autres.

### ***creationGroupesIrrigationPourListeIlots***

Action appelée depuis la méthode interne creationGroupesIrrigation. Elle choisi quelles parcelles (ou fraction de parcelle) va appartenir quel groupe d'irrigation. Méthode complexe, voir le document « [/Tasks/T4.Processus/T4.3.08 - Irrigation/T4.3.08 - Irrigation.pdf](#) » pour plus d'information.

### ***calculRendementMoyenAnneePassee***

Action appelée depuis l'action globale calculRendementMoyenAnneePasseeZoneMaelia. Elle calcule le rendement moyen par espèce pour chaque agriculteur.

### ***choixAssolement***

Action non définie pour l'agent parent agriculteur.

### ***choixActivite***

Action non définie pour l'agent parent agriculteur.

### ***activiteSemer***

Action non définie pour l'agent parent agriculteur.

### ***activiteRecolter***

Action non définie pour l'agent parent agriculteur.

### ***activiteIrriguer***

Action non définie pour l'agent parent agriculteur.

### ***activiteIrriguerComplexe***

Action appelée depuis l'action interne choixActivite, dans le cas où le modèle d'irrigation choisi est le « complexe ». Elle détermine quelles parcelles, ou du moins fraction de parcelle, peuvent être irriguer le jour courant (de 0 à n), puis elle applique l'activité sur les parcelles. Pour ce faire, l'agriculteur se base sur ses groupes d'irrigation. Chaque parcelle appartient au moins à un groupe. Au sein d'un groupe, l'irrigation se fait de proche en proche. Pour plus d'information voir le document : « [/Tasks/T4.Processus/T4.3.08 - Irrigation/T4.3.08 - Irrigation.pdf](#) ».

### ***calculSurfaceAirriguerParJour***

Action appelée depuis la méthode interne activiteIrriguerComplexe. Elle met à jour la surface qui va pouvoir être irriguée par jour et par groupe. Cette surface est normalement constante et correspond à la surface maximale d'un groupe d'irrigation divisé par la durée d'un tour d'eau. Cependant, dans le cas où des restric-

tions sont en cours il y aura certains jours de la semaine où l'agriculteur ne pourra plus irriguer. La durée d'un tour d'eau étant fixe, il faut alors jouer sur la surface maximale pouvant être irriguée chaque jour. Si le niveau de restriction est de 1 (c'est à dire 1 jour de restriction), alors on ramène la surface irrigable journalière à  $(duréeTourEau * nbJourRestriction) / NbJoursSemaine$ . Pour plus d'information voir le document : « [/Tasks/T4. Processus/T4.3.08 - Irrigation/T4.3.08 - Irrigation.pdf](#) ».

#### ***suppressionCulturesNonRecoltees***

Action non définie pour l'agent parent agriculteur.

#### ***colorationAgriculteur***

Action appelée depuis la méthode interne comportementJournalier. Elle met à jour la couleur définie pour certains attributs de l'agent.

#### ***basic***

Action d'affichage SIG appelée dans la classe gaml Experiement. Elle colore tous les agriculteurs de la même couleur.

#### ***restrictionAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Elle colore l'agriculteur selon son respect de la restriction en cours. En rouge si il y a restriction et que l'agriculteur irrigue le pas de temps courant contre restriction. En vert si il n'y a pas de restriction ou si il n'irrigue pas contre restriction.

#### ***eauDisponibleAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Elle colore l'agriculteur selon son niveau d'eau disponible. En orange si il est très proche ou égale à zéro, en vert sinon.

#### ***imageAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Affiche l'agriculteur sous la forme d'une image représentant un tracteur.

#### ***toString***

Action d'affichage des variables de l'agent dans la console.

## Agriculteurs / agriculteurComplexe.gaml

Les agents agriculteurComplexe sont créés dans le cas où le modèle de comportement de l'agriculteur choisi est le complexe. L'agriculteurComplexe a un comportement rationnel qui se base sur le paradigme BDI (Plan-Désirs-Croyances-Intention). Les croyances de l'agriculteur sont prédéfinies par des fonctions de croyances. Cet agent hérite de la classe agriculteur.

### GLOBAL

#### ***constructionAgriculteurComplexes***

Action publique appelée pendant l'initialisation depuis l'action globale constructionAgriculteur de l'agent parent agriculteur. Elle crée les agents agriculteurComplexe et initialise les profils des agriculteurs.

#### ***creationProfilsAgriculteursComplexes***

Action appelée depuis la méthode constructionAgriculteurComplexes. Elle crée les différents profils des agriculteurs par la lecture d'un fichier csv d'entrée.

#### ***creationAgriculteurComplexes***

Action appelée depuis la méthode constructionAgriculteurComplexes. Elle crée les agents agriculteurComplexe à partir des agents exploitation.

#### ***initialisationAgriculteursComplexes***

Action appelée depuis la méthode constructionAgriculteurComplexes. Elle affecte, entre autre, un profil par agriculteur.

### AGENT

#### ***miseAJourDatesActivites***

Mise à jour de la date de début d'irrigation, de semis et de récolte tous les ans (la date est la même tous les ans, seul l'année change).

#### ***choixAssolement***

Action appelée depuis l'action interne comportementAnnuel. Elle va associer à chaque parcelle de l'agriculteur un agent systemeDeCulture. Pour cela elle choisit parmi une liste de planAssolement le meilleur.

#### ***choixPlan***

Action appelée depuis l'action interne choixAssolement. Elle va choisir le meilleur plan parmi la liste définie pour chaque agriculteur, et ce en se basant sur le profil de celui-ci (cf. théorie des fonctions de croyance).

#### ***evaluationCandidat***

Action appelée depuis l'action interne choixPlan (cf. théorie des fonctions de croyance).

#### ***choixActivite***

Action appelée depuis l'action interne comportementJournalier. Elle choisit quelle(s) activité(s) va faire l'agriculteur. En fonction de son temps de travail journalier et du moment de l'année, l'agriculteur va pouvoir semer, irriguer et récolter.

#### ***activiteSemer***

Action appelée depuis l'action interne choixActivite. Elle détermine quelles parcelles peuvent être semées le jour courant (de 0 à n), puis elle applique l'activité sur les parcelles. Sachant que les espèces cultivées ont un ordre de priorité, elle va faire en sorte de semer les parcelles devant accueillir des espèces prioritaires d'abord. À chaque parcelle semée, le temps disponible diminue, l'agriculteur arrêtera de semer le jour courant si il n'a

plus de temps alloué, ou si il n'a plus de parcelle à semer. Dans ce dernier cas, il pourra passer à la 2ième activité possible : l'activité irriguer.

#### ***activiteRecolter***

Action appelée depuis l'action interne choixActivite. Elle détermine quelles parcelles peuvent être récolter le jour courant (de 0 à n), puis elle applique l'activité sur les parcelles. Sachant que les espèces cultivées ont un ordre de priorité, elle va faire en sorte de récolter les parcelles ayant des espèces prioritaires d'abord. A chaque parcelle récoltée, le temps disponible diminue, l'agriculteur arrêtera de récolter le jour courant si il n'a plus de temps alloué, ou si il n'a plus de parcelle à récolter.

#### ***suppressionCulturesNonRecoltees***

Action devant être appelée depuis l'action globale comportementJournalier. Dans le cas où il resterait des cultures non récoltées après la dernière date possible, cette action va les récolter en appliquant un malus au calcul du rendement.

#### ***activiteIrriguer***

Action appelée depuis l'action interne choixActivite, dans le cas où le modèle d'irrigation choisi est le « simple ». Elle détermine quelles parcelles peuvent être irriguer le jour courant (de 0 à n), puis elle applique l'activité sur les parcelles. Sachant que les espèces cultivées ont un ordre de priorité, elle va faire en sorte d'irriguer les parcelles ayant des espèces prioritaires d'abord. A chaque parcelle irriguer, le temps disponible diminue, l'agriculteur arrêtera l'irrigation le jour courant si il n'a plus de temps alloué, ou si il n'a plus de parcelle à irriguer. Dans ce dernier cas, il pourra passer à la dernière activité possible : l'activité récolter. Remarque : dans ce modèle simple, les parcelles ne pourront être irriguées que tous les tours d'eau. Comme le temps alloué pour l'irrigation est très petit, l'agriculteur pourra en fait toujours irriguer toutes ses parcelles le même jour. Ce qui aura pour résultat au niveau des prélèvements pour l'irrigation sur la zone d'étude un énorme pic tous les tours d'eau. Ceci n'est pas très réaliste. Pour plus d'information voir le document : « Tasks/T4. Processus/T4.3.08 - Irrigation/Irrigation.pdf ».

#### ***semer\_parc***

Action appelée depuis la méthode interne activiteSemer. Elle prend en entrée la parcelle à semer et fait appelle à la méthode semer de son agent strategieSemisComplexe associé pour ainsi créer la culture.

#### ***irriguer\_parc***

Action appelée depuis la méthode interne activiteIrriguer ou activiteIrriguerComplexe. Elle prend en entrée la parcelle à irriguer et fait appelle à la méthode irriguer de son agent strategieIrrigationComplexe associé pour ainsi irriguer la culture.

#### ***recolter\_parc***

Action appelée depuis la méthode interne activiteRecolter. Elle prend en entrée la parcelle à récolter et fait appelle à la méthode recolter de son agent strategieRecolteComplexe associé pour ainsi supprimer la culture et calculer son rendement.

## **Agriculteurs / agriculteurJeuOC.gaml**

---

Cet agent va être entièrement refait. Cet agent hérite de la classe agriculteur.

## **ITKs / itk.gaml**

---

L'ITK ou stratégie de culture correspond à l'ensemble des stratégies de l'agriculteur, différente selon l'especeCultivee. L'ITK comprend 3 types de stratégies : strategieSemi, strategieIrrigation et strategieRecolte (SIR).

Cette classe est la classe mère des agents itkComplexe et itkJeuOC (ce dernier est à redéfinir).

### GLOBAL

#### ***constructionITKetStrategie***

Action publique appelée depuis le main à l'initialisation qui crée et initialise les agents itkComplexe ou itkJeuOC selon le modèle d'agriculteur choisi dans les paramètres de simulation (ce dernier type d'agent n'est pas encore implémenté). C'est ici que vont être créés et initialisés les agents strategieSIR.

## ITKs / itkComplexe.gaml

---

Cet agent hérite de la classe itk.

### GLOBAL

#### *constructionITKetStrategieComplexe*

Action publique appelée à l'initialisation depuis la méthode globale `constructionITKetStrategie` de l'agent itk qui crée et initialise les agents `itkComplexe`. C'est ici que vont être créés et initialisés les agents strategieSemiComplexe, strategieIrrigationComplexe et strategieRecolteComplexe.

## ITKs / itkJeuOC.gaml

---

Cet agent hérite de la classe itk.

### GLOBAL

#### *constructionITKetStrategieJeuOC*

Action publique appelée à l'initialisation depuis la méthode globale `constructionITKetStrategie` de l'agent itk qui crée et initialise les agents `itkJeuOC`. C'est ici que vont être créés et initialisés les agents strategieSemiJeuOC, strategieIrrigationJeuOC et strategieRecolteJeuOC.

## PlansAssolement / planAssolement.gaml

---

L'agriculteur est amené à choisir chaque année un plan d'assolement qui correspond à la liste exhaustive de tous ses choix possible d'assolement pour toutes ses parcelles. Ainsi, un plan d'assolement va correspondre à une liste d'affectation des rotations de culture aux parcelles.

### GLOBAL

#### *constructionPlanAssolement*

Action publique appelée depuis le main à l'initialisation qui crée et initialise les agents `planAssolementComplexe` ou `planAssolementSimple` selon le modèle d'agriculteur choisi dans les paramètres de simulation (ce dernier type d'agent n'est pas encore implémenté).

## **StrategiesIrrigation / strategieIrrigation.gaml**

---

Cette classe représente la stratégie de l'agent agriculteur vis à vis de l'activité irrigation. Plusieurs stratégies d'irrigation peuvent être définies, selon le modèle agriculteur choisi. Cette classe est la classe mère de agents strategieIrrigationComplexe et strategieIrrigationJeuOC (ce dernier est à redéfinir).

### **AGENT**

#### ***est\_senable***

Action appelée à chaque pas de temps depuis l'agent agriculteur. Dans sa version générique elle ne fait rien.

#### ***semer***

Action appelée à chaque pas de temps depuis l'agent agriculteur. Dans sa version générique elle ne fait rien.

## StrategiesIrrigation / strategieIrrigationComplexe.gaml

---

Il n'y a pas de section globale. La création et l'initialisation de cet agent se fait dans la classe gaml itkComplexe. Cet agent hérite de la classe strategieIrrigation.

### AGENT

#### *est\_irrigable*

Action appelée à chaque pas de temps depuis l'agent agriculteurComplexe. Elle retourne vrai si la parcelle donnée en entrée peut être irriguée le pas de temps courant. Une parcelle sera irrigable si elle a une cultureIrrigable, si on est dans la bonne fenêtre temporelle, si il n'a pas trop plu les 2 derniers jours, si la cultureIrrigable n'est pas en restriction et enfin si il reste de la surface qui n'a pas été irrigué depuis au moins 1 tour d'eau.

#### *irriguer*

Action appelée à chaque pas de temps depuis l'agent agriculteurComplexe. Elle met à jour l'eau d'irrigation souhaitée dans la parcelle et fait une demande de prélèvement à la zoneHydrographique associée à l'agent ilot. C'est ensuite au sein de l'agent zoneHydrographique que va se calculer la quantité qui pourra effectivement être prélevée pour l'irrigation.

## StrategiesIrrigation / strategieIrrigationJeuOC.gaml

---

Il n'y a pas de section globale. La création et l'initialisation de cet agent se fait dans la classe `gaml itkJeuOC`. Cet agent hérite de la classe `strategieIrrigation`.

### AGENT

#### *est\_irrigable*

Action appelée à chaque pas de temps depuis l'agent `agriculteurJeuOC`. Pour le moment le calcul est même que celui de l'agent `strategieIrrigationComplexe`.

#### *irriguer*

Action appelée à chaque pas de temps depuis l'agent `agriculteurJeuOC`. Pour le moment le calcul est même que celui de l'agent `strategieIrrigationComplexe`.

## **StrategiesRecolte / strategieRecolte.gaml**

---

Cette classe représente la stratégie de l'agent agriculteur vis à vis de l'activité récolte. Plusieurs stratégies de récoltes peuvent être définies, selon le modèle agriculteur choisi. Cette classe est la classe mère de agents strategieRecolteComplexe et strategieRecolteJeuOC (ce dernier est à redéfinir).

### **AGENT**

#### ***est\_recoltable***

Action appelée à chaque pas de temps depuis l'agent agriculteur. Dans sa version générique elle ne fait rien.

#### ***recolter***

Action appelée à chaque pas de temps depuis l'agent agriculteur. Dans sa version générique elle ne fait rien.

## StrategiesRecolte / strategieRecolteComplexe.gaml

---

Il n'y a pas de section globale. La création et l'initialisation de cet agent se fait dans la classe gaml itkComplexe. Cet agent hérite de la classe strategieRecolte.

### AGENT

#### *est\_recoltable [Override]*

Action appelée à chaque pas de temps depuis l'agent agriculteurComplexe. Elle retourne vrai si la parcelle donnée en entrée peut être récoltée le pas de temps courant. Une parcelle sera récoltable si elle a une culture, si on est dans la bonne fenêtre temporelle, et si il n'a pas trop plu les 5 derniers jours.

#### *recolter [Override]*

Action appelée à chaque pas de temps depuis l'agent agriculteurComplexe. Elle calcule le rendement de la culture et supprime l'agent culture de l'agent parcelle.

## StrategiesRecolte / strategieRecolteJeuOC.gaml

---

Il n'y a pas de section globale. La création et l'initialisation de cet agent se fait dans la classe `gaml itkJeuOC`. Cet agent hérite de la classe `strategieRecolte`.

### AGENT

#### *est\_irrigable*

Action appelée à chaque pas de temps depuis l'agent `agriculteurJeuOC`. Pour le moment le calcul est même que celui de l'agent `strategieRecolteComplexe`.

#### *irriguer*

Action appelée à chaque pas de temps depuis l'agent `agriculteurJeuOC`. Pour le moment le calcul est même que celui de l'agent `strategieRecolteComplexe`.

## StrategiesSemis / strategieSemi.gaml

---

Cette classe représente la stratégie de l'agent agriculteur vis à vis de l'activité semi. Plusieurs stratégies de semis peuvent être définies, selon le modèle agriculteur choisi. Cette classe est la classe mère de agents strategieSemiComplexe et strategieSemiJeuOC (ce dernier est à redéfinir).

### AGENT

#### *est\_ semable*

Action appelée à chaque pas de temps depuis l'agent agriculteur. Dans sa version générique elle ne fait rien.

#### *semer*

Action appelée à chaque pas de temps depuis l'agent agriculteur. Dans sa version générique elle ne fait rien.

## StrategiesSemis / strategieSemiComplexe.gaml

---

Il n'y a pas de section globale. La création et l'initialisation de cet agent se fait dans la classe `gaml itkComplexe`. Cet agent hérite de la classe `strategieSemi`.

### AGENT

#### *est\_semable [Override]*

Action appelée à chaque pas de temps depuis l'agent `agriculteurComplexe`. Elle retourne vrai si la `parcelle` donnée en entrée peut être semée le pas de temps courant. Une `parcelle` sera semable si elle n'a pas encore de `culture`, si on est dans la bonne fenêtre temporelle, et si il n'a pas trop plu les 5 derniers jours.

#### *semer [Override]*

Action appelée à chaque pas de temps depuis l'agent `agriculteurComplexe`. Elle crée l'agent `culture` (ou `cultureIrrigable`) dans l'agent `parcelle`.

## **StrategiesSemis / strategieSemiJeuOC.gaml**

---

Il n'y a pas de section globale. La création et l'initialisation de cet agent se fait dans la classe `gaml itkJeuOC`. Cet agent hérite de la classe `strategieSemi`.

### **AGENT**

#### ***est\_irrigable***

Action appelée à chaque pas de temps depuis l'agent `agriculteurJeuOC`. Pour le moment le calcul est même que celui de l'agent `strategieSemiComplexe`.

#### ***irriguer***

Action appelée à chaque pas de temps depuis l'agent `agriculteurJeuOC`. Pour le moment le calcul est même que celui de l'agent `strategieSemiComplexe`.

## SystemesDeCultures / systemeDeCulture.gaml

---

Classe mère. Il est possible de définir un SdC plus ou moins simplement, mais à chaque fois on aura le même schéma. Par exemple, on peut se baser sur des données réelles (systemeDeCultureJeuOC), ou alors sur une liste prédéfinies (systemeDeCultureComplexe). Le système de culture crée va dépendre du modèle agriculteur choisi. Les agents systemeDeCulture sont statiques. Ils n'évoluent pas au cours de la simulation.

### GLOBAL

#### *constructionSystemeDeCulture*

Action publique appelée depuis le main à l'initialisation qui crée et initialise les agents systemeDeCultureComplexe ou systemeDeCultureJeuOC selon le modèle d'agriculteur choisi dans les paramètres de simulation.

## SystemesDeCultures / systemeDeCultureComplexe.gaml

---

Les agents `systemeDeCultureComplexe` sont prédéfinis par une liste fixe en entrée de la simulation. Il y aura ainsi 10 rotations possibles pour les agents `agriculteurComplexe`. Cet agent hérite de la classe `systemeDeCulture`.

### GLOBAL

#### ***constructionSystemeDeCultureComplexe***

Action publique appelée à l'initialisation depuis la méthode globale `constructionSystemeDeCulture` de la classe `systemeDeCulture` qui crée et initialise les agents `systemeDeCultureComplexe`.

## SystemesDeCultures / systemeDeCultureJeuOC.gaml

---

Les agents `systemeDeCultureJeuOC` sont basées sur des données réelles. Les agents `agriculteurJeuOC` auront ainsi toujours la même rotation en cycle. Cet agent hérite de la classe `systemeDeCulture`.

### GLOBAL

#### *constructionSystemeDeCultureComplexe*

Action publique appelée à l'initialisation depuis la méthode globale `constructionSystemeDeCulture` de la classe `systemeDeCulture` qui crée et initialise les agents `systemeDeCultureJeuOC` à partir d'un fichier d'entrée.

## **clcRPG.gaml**

---

Il n'y a pas de section agent. Les agents clcRPG sont statiques. Ils n'évoluent pas au cours de la simulation. Ces agents sont uniquement utiles à l'initialisation des agents zonesHydrographiques au moment où le clc est associé aux ZH.

### **GLOBAL**

#### ***constructionClcRPG***

Action publique appelée depuis le main à l'initialisation qui crée et initialise les agents clcRPG à partir d'un fichier SIG d'entrée.

## Cultures / culture.gaml

---

Il n'y a pas de section globale. La création et l'initialisation de cet agent se fait dans la classe gaml strategieDeSemi. La culture est une entité qui ne va exister qu'entre le jour de son semi et celui de sa récolte. Elle est semée sur une parcelle.

### AGENT

#### **Attributs**

- espece [especeCultivee]
- isCultureIrrigable
- etpCulturePasDeTempsCourant
- temperatureMoyenne
- etm
- kc
- kcPrec
- alpha
- indexDateDeCreation
- nombreJourEcouleDepuisCreationCulture
- couleurCoefficientCultural
- isCultureHiver - JeuOC (pas utilisé)
- frein - JeuOC (pas utilisé)
- coefVigueurVegetative - JeuOC (Cvig)
- coefFermetureStomates - JeuOC (Csto)
- ecartReliquatN - JeuOC (dNif)
- sommeTemperatureAnneePrecedente - JeuOC
- sommeTemperatureSemi - JeuOC
- indiceSatisfactionHydrique - JeuOC (TR\_M)
- transpirationMax - JeuOC
- deltaCoefCultural - JeuOC
- echV - JeuOC
- echVPrec - JeuOC

#### **comportementJournalier**

Action appelée à chaque pas de temps depuis le main UNIQUEMENT dans le cas où le modèle simple de croissance des plantes est choisi.

#### **croissanceCulture**

Action appelée depuis l'action comportementJournalier. Elle correspond au processus simple de croissance des plantes. Elle met à jour le stade phénologique de la plante, dont les données ont été chargées à l'initialisation depuis un fichier externe.

#### **calculReserveAccessibleRacine**

Action appelée depuis l'agent parcelle, dans l'action calculQuantiteEauDeRuissellementJeuOC. Elle est une des 2 actions de l'agent culture utile pour le processus de croissance des plantes utilisant le modèle « Jeu d'OC ». Cette action met à jour la réserve utile accessible aux racines.

#### **calculCoefficientCultural**

Action appelée depuis l'agent parcelle, dans l'action etatDuSol. Elle est une des 2 actions de l'agent culture utile pour le processus de croissance des plantes utilisant le modèle « Jeu d'OC ». Cette action calcule le coefficient cultural (indication sur le stade phénologique de la culture).

***changementCouleurEnFonctionDebit***

Action appelée depuis les actions croissanceCulture ou calculCoefficientCultural selon le modèle de croissance des plantes choisi. Elle met à jour la couleur associée à la culture en fonction de son stade phénologique (valeur du coefficient cultural). La culture n'étant pas une entité géo-référencée, cette couleur est ensuite utilisée par l'agent ilot.

## Cultures / cultureIrrigable.gaml

---

Il n'y a pas de section globale. La création et l'initialisation de cet agent se fait depuis la classe `gaml strategieDeSemi`, elle même appelée depuis l'`agriculteur`. La culture est une entité qui ne va exister qu'entre le jour de son semé et celui de sa récolte. Elle est semée sur une parcelle. Elle hérite de l'agent culture.

### AGENT

#### ***Attributs***

- `isPeutEtreIrrigee`
- `mapHistoriqueIrrigation`
- `mapHistoriqueIrrigationSimplifiee`
- `surfaceNePouvantPasEtreIrrigee [m2]`
- `dernierTourEau`
- `listeJoursDeLaSemaineRestreints`
- `ratio`
- `mapCoeffAbattementRendementParJour`
- `rendementCulture`

#### ***miseAjour***

Action appelée à chaque pas de temps depuis le main, après que l'irrigation ainsi que la gestion d'étiage aient été réalisées. Selon le modèle d'irrigation choisi, elle va mettre à jour le nombre de jours passés depuis la dernière irrigation de la culture (ou de la fraction de culture). Elle va également mettre à jour l'état de restriction de la culture.

#### ***miseAjourNbJoursDernierTourEauBasique***

Action appelée depuis l'action `miseAjour`. Elle est utilisée dans le cas où le modèle d'irrigation choisi est le modèle dit « simple ». Et elle correspond à la simple mise à jour du nombre de jours passés depuis la dernière irrigation de la culture.

#### ***miseAjourNbJoursDernierTourEauComplexe***

Action appelée depuis l'action `miseAjour`. Elle est utilisée dans le cas où le modèle d'irrigation choisi est le modèle dit « complexe » (celui avec création de groupe d'irrigation). L'agent `cultureIrrigable` possède une « `map` » qui stocke le nombre de jour depuis la dernière irrigation par surface. En effet, la culture est partitionnée dans le cas où elle appartient à plusieurs groupes d'irrigation. Cette action va augmenter le nombre de jours associés aux différentes surfaces de la culture. Dès que le nombre de jour est supérieur au tour d'eau, la surface est supprimée de la « `map` ».

#### ***isEnRestriction***

Action appelée depuis l'agent parcelle. Elle renvoie vrai si la culture est en restriction le jour de la semaine courant (par exemple, une culture peut être en restriction uniquement le lundi).

#### ***isEnStressHydrique***

Action appelée depuis l'agent parcelle. Elle renvoie vrai si la culture est en stress hydrique. Ceci est déterminé à partir d'un `ratio` qui ne doit pas être inférieur à 1. TODO : pour le modèle Jeu d'OC, cette méthode est à revoir.

#### ***calculCoefficientAbattement***

Action appelée depuis l'agent parcelle. Cette action n'est valide que pour le modèle de croissance des plantes simple. Elle calcule un coefficient d'abattement utile au calcul final du rendement de la culture le jour de sa récolte. TODO : pour le modèle Jeu d'OC, cette méthode est à revoir.

***calculRendement***

Action appelée depuis l'agent strategieRecolteComplexe. Cette action n'est valide que pour le modèle de croissance des plantes simple. Elle renvoie le rendement de la culture. TODO : pour le modèle Jeu d'OC, cette méthode est à revoir.

## especeCultivee.gaml

---

Dans MAELIA, seuls quelques types d'espèces cultivées sont considérés, ceux qui sont le plus représentatifs de la zone d'étude.

### GLOBAL

#### *constructionEspeceCultivees*

Action appelée à l'initialisation depuis le main. Elle crée depuis un fichier d'entrée la dizaine de type de cultures prédéfinies pour MAELIA.

### AGENT

Cet agent n'a pas d'action, seulement de attributs constants au cours de la simulation.

## **exploitation.gaml**

---

Il y a une exploitation par agent agriculteur. Une exploitation regroupe tous les agents ilots.

### **GLOBAL**

#### ***constructionExploitations***

Action appelée à l'initialisation depuis le main. Elle crée les agents exploitations à l'aide des agents ilots, et plus précisément grâce au code de l'exploitation de l'ilot récupéré depuis les données d'entrée.

### **AGENT**

#### ***initialisationExploitation***

Action appelée à l'initialisation depuis la méthode globale `constructionExploitations`. Elle initialise, entre autre, les `uth`.

#### ***basic***

Action d'affichage SIG appelée dans la classe `gaml Experiment`.

#### ***restrictionAspect***

Action d'affichage SIG appelée dans la classe `gaml Experiment`. Elle affiche l'exploitation en fonction de son rendement. TODO : à revoir.

## **Ilots / ilot.gaml**

---

L'ilot est la seule entité géo-référencée du modèle agricole. C'est par regroupement des ilots que l'on peut avoir une visualisation de l'exploitation. Mais on ne peut en revanche avoir aucune représentation de ses parcelles. L'ilot est l'entité qui va faire le lien avec le modèle hydrographique. De même, c'est sur l'ilot que la pluie va tomber.

### **GLOBAL**

#### ***Attributs***

- ilotsAllDepartementsShape - Chemin vers le fichier SHP
- mapAssociationListeIlotsParZH
- mapAssociationZHParIlot
- mapIlots
- listeIlots [ilots]
- listeExploitations

#### ***constructionIlot***

Action appelée à l'initialisation depuis le main. Elle crée les agents ilot à partir des données SIG d'entrées.

#### ***initialisationIlot***

Action appelée à l'initialisation depuis le main. Elle est faite séparément de la construction car il faut d'abord créer les parcelles avant de pouvoir initialiser les ilots.

#### ***creationMapsAssociationZHsIlots [Indicateur]***

Action appelée à l'initialisation depuis l'action constructionIlot. Elle remplit les maps liant les ilots aux ZH. Ces maps sont ensuite utiles pour avoir une et une seule combinaison ilots/ZH possible.

### **AGENT**

#### ***Attributs***

- id
- idZhAssociee
- idTypeDeSol
- isIlotIrrigable
- isPasRemplissageIlotsCulturePrincipale
- isPasRemplissageIlots
- couleurIlot
- couleurIlotParExploitation
- couleurIlotIrrigable
- codeExploitationAssociee
- isIrrigable
- departement
- zoneMeteoMoyenneAssociee [zoneMeteo]
- zoneHydroAssociee [zoneHydrographique]
- surfaceIlotAPrtirDesParcelles [m2]
- listeParcelles [parcelles]
- listeParcellesHydro [parcelles]
- parcellePrincipale [parcelle]
- typeDeSolIlot [typeDeSol]
- listePointDePrelevementIrrAssocie [equipementDeCaptageIRR]
- retenueCollinaireAssociee [retenueCollinaire]

- agriculteurAssocie [agriculteur]
- clcPlusProche [clc]
- idNatureRessourcePrelevee
- isAssociePointPrelevement
- isAssocieRetenueCollinaire
- hruRPGassociee [hru]
- communeAssociee [commune]
- penteAssociee

### ***croissancePlante***

Action appelée depuis 2 endroits différents selon le modèle hydrographique utilisé. Ainsi, elle va être appelée à chaque pas de temps depuis le main, uniquement si le modèle hydrographique pris en compte n'est pas SWAT, ou si il n'y a pas de modèle hydrographique dans la simulation en cours. En effet, dans le cas de SWAT il faut prendre en compte l'écoulement des ilots de manière interne, les 2 processus (écoulement de l'eau et croissance des plantes) sont donc liés. Pour plus d'information, voir les documents « [/Tasks/T4.Processus/T4.3.01 - Ecoulement de l'eau/Ecoulement de l'eau\\_RPG/ T4.3.01 - Lien entre SWAT et les prelevements.pdf](#) et [T4.3.01 - Lien SWAT\\_RPG.pdf](#) ». Si le modèle hydrographique est SWAT, cette action est appelée à chaque pas de temps depuis l'agent HRUrpq.

### ***initialisationIlots***

Action appelée depuis la méthode initialisationIlot. Elle initialise certains attributs de l'agent ilot, comme par exemple le typeDeSol ou la zoneHydrographique associée.

### ***precipitationsEtETP***

Action appelée depuis le main. Elle est appelée au début du pas de temps. Du moins avant que les agriculteurs irriguent et avant la croissance des plantes.

### ***prelevementEau***

Action appelée depuis l'agent strategieIrrigationComplexe. Elle est appelée au début du pas de temps. Du moins avant que les agriculteurs irriguent et avant la croissance des plantes.

### ***ruissellementVersZH***

Action appelée depuis croissancePlante de l'agent ilot. C'est depuis cette méthode que sont calculées les volumes de ruissellement, d'évapotranspiration ainsi que de drain qui s'échappent des parcelles. Ces volumes sont soit renvoyés vers la ZH de l'ilot, soit vers la HRUrpq de l'ilot dans le cas où le modèle hydrographique est SWAT.

### ***basic***

Action d'affichage SIG appelée dans la classe gaml Experiment. Colore tous les ilots de la même manière.

### ***videoAspect***

Action d'affichage SIG appelée dans la classe gaml Experiment. Utilisé pour un experiment particulier pour la création d'une vidéo.

### ***cultureAspect***

Action d'affichage SIG appelée dans la classe gaml Experiment. Colore l'ilot en fonction de l'especeCultivee sur sa parcelle principale..

### ***exploitationAspect***

Action d'affichage SIG appelée dans la classe gaml Experiment. Colore l'ilot en fonction de l'exploitation à laquelle il appartient.

***coefficientCulturalCulturePrincipaleAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore l'ilot en fonction du stade phénologique de la plante (culture) de sa parcelle principale.

***rfuParcelleAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore l'ilot en fonction du remplissage de la réserve facilement utilisable de sa parcelle principale (uniquement si le modèle de croissance des plantes considéré est le simple).

***ilotIrrigableAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore l'ilot en fonction de son caractère irrigable ou non (définie par les données SIG d'entrées). En bleu si il peut contenir des cultureIrrigable en vert sinon.

***ilotIrrigueAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore l'ilot en fonction du caractère irrigable ou non de la culture courante de sa parcelle principale.

***parcellePrincipaleEnRestricton***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore l'ilot en fonction de l'état de restriction de la culture d'une de ses parcelles. En rouge si la culture ne peut pas être irriguée le pas de temps courant, en vert sinon.

***parcellePrincipaleEnFonctionDuNiveauDeRestriction***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore l'ilot en fonction du nombre de jour de restriction de la culture d'une de ses parcelles. Les couleurs vont du jaune au rouge, en passant par le orange. Jaune si l'ilot est restreint 1 jour (niveau 1), rouge si il l'est 7 jours (niveau 4).

***etatIrrigationParcelle***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore l'ilot en fonction de l'état d'irrigation de la culture d'une de ses parcelles. En blanc, il n'y a pas d'irrigation demandée, en orange l'irrigation ne peut pas se faire car il n'y a plus d'eau, en dégradé de bleu il y a irrigation mais pas toute la quantité souhaitée (du bleu clair au bleu foncé il y a de plus en plus d'eau), en vert il y a assez d'eau pour l'irrigation, en gris l'irrigation ne se fait pas car la culture est en restriction et enfin en rouge l'irrigation se fait contre restriction.

***ilotEnStressHydrique***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore l'ilot en fonction de l'état de stress de la culture d'une de ses parcelles. En rouge, la plante est en stress hydrique, en vert non.

***toString***

Action d'affichage des variables de l'agent dans la console.

## **Ilots / ilotHorsZone.gaml**

---

Ces ilots n'appartiennent pas géographiquement à la zone MAELIA. Ils sont pris en compte pour la cohérence de exploitations. Ainsi ces ilots appartiennent forcément à une exploitation agricole de la ZM, ils seront traités plus simplement que les ilots classiques car on ne peut pas connaître les débits simulés hors ZM. Cet agent est le fils de l'agent ilot.

### **GLOBAL**

#### ***constructionIlotHorsZone***

Action appelée à l'initialisation depuis le main. Elle crée les agents ilotHorsZone à partir des données SIG d'entrées.

### **AGENT**

#### ***initialisationIlots [Overwrite]***

Action appelée depuis la méthode initialisationIlot, dans le global de l'agent parent ilot. Elle initialise certains attributs de l'agent ilot, comme par exemple le typeDeSol ou la zoneHydrographique associée.

## marcheAgricole.gaml

---

Cette classe regroupe tous les aspects prix des intrants, pour l'achat des récoltes et la vente de semences.

### GLOBAL

#### *constructionMarcheAgricole*

Action appelée à l'initialisation depuis le main. Elle crée un seul agent marcheAgricole.

### AGENT

#### *comportementAnnuel*

Action appelée chaque année depuis le main.

#### *initialisationMarcheAgricole*

Action d'initialisation appelée depuis l'action globale constructionMarcheAgricole. Elle initialise les prix à partir d'un fichier csv d'entrée.

#### *majMarcheAgricole*

Action appelée depuis comportementAnnuel. Met à jour les prix par lecture d'un csv d'entrée.

## Parcelles / parcelle.gaml

---

La parcelle est une partie d'ilot qui va posséder une culture au cours de l'année. La parcelle n'est pas une entité géo-référencée (mais son ilot oui). Dans la parcelle les quantités d'eau manipulées sont exprimées en hauteur (mètre), en entrée et en sortie de la parcelle ces quantités sont des volumes (mètre cube).

### GLOBAL

#### ***Attributs***

- parcellesShape - Chemin vers le fichier SHP
- listeParcelles [parcelles]
- mapSurfaceParcellesAtraiteesParHydro
- mapParcelleRecolteeParCulture
- volumeEauPluieEntreeParcellesZoneMaelia [m3]
- volumeEauIrrigationEntreeParcellesZoneMaelia [m3]
- volumeEauEntreeParcellesZoneMaelia [m3]
- volumeEauSortieParcellesZoneMaelia [m3]
- volumeEauIrrigationSouhaiteZoneMaelia [m3]
- volumeEauIrrigationReelZoneMaelia [m3]
- mapNombreTypeCultureGlobaleReelle

#### ***constructionParcelles***

Action appelée à l'initialisation depuis le main. Selon le modèle de croissance des plantes en cours, cette action construira un type particulier d'agent parcelle. Pour le moment 2 types sont possibles (pour les 2 modèles de croissances des plantes) : parcelle et parcelleJeuOC.

#### ***creationParcelles***

Action appelée depuis l'action constructionParcelles. Elle appelle une méthode générique prenant en entrée le nom d'un fichier et crée les parcelles à partir de ce fichier.

#### ***initialisationParcelles***

Action appelée depuis l'initialisation du main. Elle se fait séparément de la création car il faut que les agents ilot soient initialisés avant. Elle initialise toutes les parcelles de la simulation (quelles soient dans la zone ou non : parcelle / parcelleJeuOC et parcelleHorsZone).

#### ***lectureFichierParcelle***

Action générique prenant en entrée le nom d'un fichier et créant les parcelles à partir de ce fichier. Si les fichiers d'entrée sont dans un même format, cette action peut créer autant de type d'agents parcelle différents qu'il y a de fichiers disponibles. Pour le moment cette action est utile pour créer les agents parcelle, parcelleJeuOC et parcelleHorsZone.

#### ***remiseAZeroVolumeParcellesZoneMaelia***

Action appelée en début de chaque pas de temps depuis le main. Elle remet simplement à zéro des variables globales propre aux agents parcelle.

#### ***calculVolumeParcelleZoneMaelia [Indicateur]***

Action appelée à chaque pas de temps depuis le main. Elle calcule le volume d'eau total qui arrive sur les parcelles (l'eau de pluie + irrigation).

#### ***comparaisonAssolementReelleVSSimuleParParcelle [Indicateur]***

Action appelée à chaque fin d'année depuis le main. Elle est utile pour la calibration, car elle compare les données réelles et simulées des especeCultivee de chaque parcelle. Les données réelles n'étant disponibles

pour le moment que entre 2006 et 2009, elle compare les especeCultivee de chaque parcelle uniquement sur cet intervalle. Elle renvoie un pourcentage correspondant au nombre de parcelles ayant exactement la même especeCultivee en simulée qu'en réel pour la même année. Par exemple si en 2006 une parcelle a un blet en réel et un maïs en simulé, on aura un pourcentage de 0 pour cette parcelle.

#### ***comparaisonRotationReelleVSSimuleParParcelle [Indicateur]***

Action appelée à la fin de l'année 2009 depuis le main. Elle est utile pour la calibration, car elle compare les données réelles et simulées des especeCultivee de chaque parcelle. Elle compare le nombre de culture de la rotation de chaque parcelle entre 2006 et 2009. Par exemple si une parcelle a une rotation maïs, blet, maïs, blet en réel et blet, maïs, blet, orge en simulé on aura alors un pourcentage pour cette parcelle de  $\hat{A}_i$  de concordance (car il a au moins 2 blet et 1 maïs dans chacune). Elle a donc plus de chance d'avoir de meilleurs résultats que comparaisonAssolementReelleVSSimuleParParcelle qui elle compare à l'année près.

#### ***comparaisonRotationReelleVSSimuleZM [Indicateur]***

Action appelée à chaque fin d'année entre 2006 et 2009 depuis le main. Elle est utile pour la calibration, car elle compare les données réelles et simulées especeCultivee de chaque parcelle. Elle compare le nombre de culture de la rotation de la zone MAELIA entre 2006 et 2009. Dans ce cas on compte le nombre de chaque especeCultivee par année et on compare les données réelles et simulées. Elle a donc plus de chance d'avoir de meilleurs résultats que comparaisonRotationReelleVSSimuleParParcelle et comparaisonAssolementReelleVSSimuleParParcelle qui elle comparent par parcelle ou par 4 ans.

### AGENT

#### ***Attributs***

- idParcelle
- rotationReelle
- surfaceDonneesEntrees [m2]
- surface [m2]
- isParcelleHorsZone
- idSystemeDeCultureParcelle
- systemeDeCultureParcelle [systemeDeCulture]
- ITK\_annee [itk]
- cpt\_annee
- indiceRotationDepart
- mapIndiceDepartRotation
- derniereProduction
- cultureParcelle [culture]
- typeDeSolParcelle [typeDeSol]
- ilot\_app [ilot]
- isParcelleIrrigable
- isParcelleIrrigee
- etatIrrigationParcelle
- mapRotationInit
- mapRotationSimulee
- cpt
- mapSurfaceParcelleGroupeIrrigation
- reserveFacilementUtilisable [m]
- pluie [m]
- irrigationReelle [m]
- irrigationSouhaitee [m]
- pluieEtIrrigation [m]

- etpParcelle [m]
- temperatureMoyenneParcelle
- quantiteEauDeRuissellement [m]
- ES
- rendementParcelle
- penteParcelle
- couleurParcelle
- couleurCulture
- couleurCoefCultural
- couleurParcelleRFU
- couleurIsIrrigable
- couleurIsIrrigee
- couleurIsEnRestriction
- couleurNiveauDeRestriction
- couleurEtatIrrigation
- couleurParcelleSiIrrigeePasDeTempsCourant
- couleurIsStressHydrique
- couleurIsParcelleUtile
- isPasRemplissage

#### ***initialisationParcelle***

Action appelée à l'initialisation dans l'action globale `initialisationParcelles`. Elle initialise la dernière production et le `typeDeSol` de la parcelle.

#### ***initDerniereProd***

Action appelée à l'initialisation dans l'action interne `initialisationParcelle`. Elle initialise la dernière production.

#### ***remiseAzeroParcelle***

Action appelée à chaque pas de temps depuis de l'agent `ilot`. Elle remet à zéro certaines variables le nécessitant en début de tour.

#### ***miseAJourPluieETPetTemperature***

Action appelée en début de chaque pas de temps depuis de l'agent `ilot`. Elle stocke l'eau de pluie (et l'ETP) dans la parcelle sous forme d'une hauteur.

#### ***hauteurEauIrrigationARajouter***

Action appelée à chaque fois que la parcelle reçoit de l'eau d'irrigation, donc depuis l'agent `strategieIrrigationComplexe`. Cela peut donc arriver plusieurs fois dans un pas de temps selon le modèle d'irrigation choisi. En effet, si la parcelle appartient à plusieurs groupes d'irrigation, chaque portion appartenant chacune à un groupe pourront être irrigué en même temps. Cette action met à jour l'irrigation souhaitée pour la parcelle. L'irrigation effectivement prodiguée sera calculée par le modèle hydrographique. Dans le cas où aucun modèle hydrographique n'est pris en compte, l'irrigation réelle sera automatiquement égale à la souhaitée.

#### ***calculEauEntreeReelleSurParcelle***

Action appelée depuis l'action interne `calculQuantiteEauDeRuissellement`. Elle joue 2 rôles, le premier est de mettre à jour la quantité effective d'eau qui est arrivée en entrée de la parcelle pour pouvoir appliquer le processus de croissance des plantes. Son deuxième but est de mettre à jour une variable pour l'affichage de l'état d'irrigation de l'agent `ilot`.

#### ***calculQuantiteEauDeRuissellement***

Action appelée chaque pas de temps depuis l'agent ilot. Elle calcule la quantité d'eau qui va ruisseler après la croissance de plantes, ce processus est d'ailleurs appelé au sein de cette méthode. Cette action est appelée dans le cas où le modèle de croissance de plantes est le simple.

***calculEvapoTranspiration***

Action non définie pour la croissance des plantes simple.

***calculEcoulementEauSouterraine***

Action non définie pour la croissance des plantes simple.

***isEnStressHydrique***

Action appelée chaque fois que l'on souhaite savoir si la parcelle (ou la culture) est en stress hydrique, donc principalement depuis l'agent strategieIrrigationComplexe.

***isEnRestriction***

Action appelée chaque fois que l'on souhaite savoir si la parcelle (ou la culture) est en restriction, donc principalement depuis l'agent strategieIrrigationComplexe.

***n\_a\_pas\_ete\_recoltee***

Action appelée depuis l'agent agriculteurComplexe. TODO : (pourquoi?) Elle n'est plus utilisée pour le moment.

***colorationParcelle***

Action appelée tous les jours depuis la méthode croissancePlante de l'agent ilot. La parcelle sera colorée en fonction de différents paramètres tels que : si sa culture est en cours d'irrigation, si sa culture est en stress hydrique, etc.

## Parcelles / parcelleJeuOC.gaml

---

Les agents parcelleJeuOC sont les parcelles créées dans le cas où le modèle de croissance des plantes est le modèle dit jeuOC. Cette classe est issue de la classe parcelle. Seules quelques actions spécifiques sont redéfinies.

### GLOBAL

#### *constructionParcellesJeuOC*

Action appelée depuis l'action globale constructionParcelles de l'agent parcelle. Elle appelle une méthode générique prenant en entrée le nom d'un fichier et crée les parcelles à partir de ce fichier.

### AGENT

#### *initialisationParcelle [Overwrite]*

Action appelée à l'initialisation dans l'action globale initialisationParcelles. Elle initialise la dernière production, le typeDeSol ainsi que certaines variables propres au modèle de croissance des plantes JeuOC.

#### *remiseAzeroParcelle [Overwrite]*

Action appelée à chaque pas de temps depuis l'agent ilot. Elle remet à zéro certaines variables le nécessitant en début de tour.

#### *calculQuantiteEauDeRuissellement [Overwrite]*

Action appelée chaque pas de temps depuis l'agent ilot. Elle calcule la quantité d'eau qui va ruisseler avant que l'eau ne s'infiltre dans le sol ou bien ne s'évapore.

#### *calculEvapoTranspiration [Overwrite]*

Action appelée chaque pas de temps depuis l'agent ilot. Elle calcule la quantité d'eau qui va s'évaporer, ou être transpirée par la plante.

#### *calculEcoulementEauSouterraine [Overwrite]*

Action appelée chaque pas de temps depuis l'agent ilot. Elle calcule la quantité d'eau qui va être drainée par le sol et redirigée vers la zoneHydrographique.

## Parcelles / parcelleHorsZone.gaml

---

Les agents parcelleHorsZone sont les parcelles appartenant aux agents ilotHorsZone. Cette classe est issue de la classe parcelle. Les actions redéfinies le sont en faisant de grandes simplifications.

### GLOBAL

#### *constructionParcelleHorsZone*

Action appelée pendant la phase d'initialisation depuis le main. Elle appelle une méthode générique prenant en entrée le nom d'un fichier et crée les parcelles à partir de ce fichier.

### AGENT

#### *initialisationParcelle [Overwrite]*

Action appelée à l'initialisation dans l'action globale initialisationParcelles. Elle initialise la dernière production.

#### *calculQuantiteEauDeRuissellement [Overwrite]*

Action non définie pour cet agent.

#### *isEnStressHydrique [Overwrite]*

Action appelée chaque fois que l'on souhaite savoir si la parcelle (ou la culture) est en stress hydrique, donc principalement depuis l'agent strategieIrrigationComplexe. Elle retourne toujours faux, car il n'y a aucun moyen de connaître la quantité d'eau apportée à ces agents (étant hors de la zone d'étude).

#### *calculRendement*

Action appelée le jour de la récolte de la culture de cette parcelle, depuis l'agent strategieRecolteComplexe. Elle récupère le rendement de la parcelle de la zone d'étude la plus proche, ayant eu le même culture l'année courante et ayant déjà été récoltée.

## **strategieSIR.gaml**

---

Classe mère des espèces d'agent `strategieSemi`, `strategieIrrigation` et `strategieRecolte`.

### **AGENT**

#### ***majDonnees***

Action appelée à l'initialisation depuis l'action globale `constructionITKetStrategieComplexe` de l'agent `itk-Complexe`.

#### ***majDiscontinue***

Action appelée à l'initialisation depuis l'action globale `constructionITKetStrategieComplexe` de l'agent `itk-Complexe`.

#### ***D1estPosterieureAD2***

Action appelée depuis l'action interne `majDiscontinue`.

#### ***fenetreTempOk***

Action appelée à chaque fois que l'on souhaite savoir si on se trouve dans la bonne fenêtre temporelle pour appliquer la stratégie. Elle est donc appelée dans les actions `est_semable`, `est_irrigable`, et `est_recoltable` des agents `strategieSemiComplexe`, `strategieIrrigationComplexe` et `strategieRecolteComplexe`.

## Package « modeleHydrographique »

- barrage.gaml
- clc.gaml
- coursDeau.gaml
- equipement.gaml
- equipementDeCaptage.gaml
- equipementDeCaptageAEP.gaml
- equipementDeCaptageIND.gaml
- equipementDeCaptageIRR.gaml
- equipementDeRejet.gaml
- equipementDeRejetAEP.gaml
- equipementDeRejetIND.gaml
- grosseIndustrie.gaml
- hru.gaml
- hruRPG.gaml
- lac.gaml
- mnt.gaml
- nappePhreatique.gaml
- noeudHydrographique.gaml
- retenueCollinaire.gaml
- zoneHydrographique.gaml
- zoneHydrographiqueSWAT.gaml

## **barrage.gaml**

---

Les agents barrages sont intimement liés aux agents gestionnaireDeBarrage. Le barrage est l'entité qui va libérer de l'eau en vue de limiter les restrictions pour les agriculteurs. Pour le moment, c'est au niveau du barrage que l'algorithme est codé, mais peut-être faudrait-il mettre une partie plutôt au niveau du gestionnaire de barrage.

### GLOBAL

#### ***constructionBarrage***

Action appelée pendant la phase d'initialisation depuis le main. Elle crée, pour le moment, 1 seul agent barrage, correspondant au barrage du lac d'Oô.

### AGENT

#### ***comportementJournalier***

Action appelée chaque pas de temps depuis le main.

#### ***comportementAnnuel***

Action appelée chaque année depuis le main.

#### ***initialisationBarrage***

Action appelée dans l'action globale constructionBarrage.

#### ***estimationPrelevementsJournalier***

Action appelée une fois par an depuis la méthode globale comportementAnnuel. Elle calcule le volume estimé pour l'irrigation de l'année à venir.

#### ***remplissageBarrage***

Action appelée une fois par an depuis la méthode globale comportementAnnuel. Pour le moment, il n'y a aucun processus remplissant le barrage. On considère donc que chaque début d'année le barrage est plein (du moins la réserve pour la période d'étiage).

#### ***isLachePourEtiagePossible***

Action appelée à chaque fois que l'on veut savoir si un lâcher de barrage pour le soutien d'étiage est possible, donc uniquement dans les agents zoneAdministrative. Le lâcher de barrage n'est pas possible avant le 15 août.

#### ***isLachePourEtiageEnCours***

Action appelée à chaque fois que l'on veut savoir si un lâcher de barrage pour le soutien d'étiage est en cours, donc uniquement dans les agents zoneAdministrative.

#### ***majDebitPourEtiageAFournir***

Action appelée depuis l'agent zoneAdministrative. Elle met à jour le débit à fournir.

#### ***lacheDeauPourEtiage***

Action appelée depuis la méthode interne comportementJournalier. Une fois que les agents zoneAdministrative ont déterminé quel débit lâcher, on applique cette action pour réaliser le lâché et envoyer d'eau dans la zoneHydrographique associée.

#### ***arretLacheDeauPourEtiage***

Action appelée au début de la méthode interne comportementJournalier. Si un lâché dure depuis plus de 7

jours (et qu'il n'a pas augmenté depuis), alors on le stoppe. TODO : à revoir, pour le faire au jour le jour.

***colorationBarrage***

Action appelée depuis la méthode interne comportementJournalier. Elle a pour rôle de définir la couleur à donner au barrage en fonction de son débit de lâcher.

***evolutionDebitLache***

Action d'affichage SIG appelée dans la classe gaml Experiement. Le barrage est coloré en fonction de son débit de lâcher.

***toString***

Action d'affichage des variables de l'agent dans la console.

## **clc.gaml**

---

CLC, ou Corinne Land Cover, est une base de données européenne d'occupation biophysique des sols. Elle s'organise autour de 5 grandes classes (elles mêmes subdivisées en n classes) : Territoires artificialisés, Territoires agricoles, Forêts et milieux semi-naturels, Zones humides, Surfaces en eau. Pour Maelia, les classes du CLC sont remaniées pour mieux coller à notre problématique.

### GLOBAL

#### ***constructionCLC***

Action appelée pendant la phase d'initialisation depuis le main. Elle crée les agents clc à partir d'un fichier SIG d'entrée.

#### ***initialisationMapDisparitionIlots***

Action appelée pendant la phase d'initialisation depuis le main. Par la lecture d'un fichier csv d'entrée, elle initialise dans une map les information utiles pour le processus de disparition des ilots. Ce fichier d'entrée est à refaire, mais pour le moment on récupère le nombre d'ilot à faire disparaître par zoneHydrographique.

#### ***choixIlotsAsupprimer***

Action appelée à la fin de chaque année depuis le main. Elle va choisir quels ilots vont disparaître et au profit de quel couvert (forêt ou bâti) chaque année. TODO : cette action doit être supprimée pour la mettre sous forme d'un prétraitement.

#### ***choixIlotsAsupprimerLocal***

Action appelée dans la méthode globale choixIlotsAsupprimer. Elle renvoie les ilots qui vont être supprimés en fonction du nombre donné en entrée.

#### ***disparitionDesIlots***

Action appelée à la fin de chaque année depuis le main, et ce après que le choix des ilots à supprimer ai été fait à l'aide de l'action globale choixIlotsAsupprimer. Cette action supprime les ilots, ainsi que toutes les traces que l'on pourrait garder d'eux une fois supprimer (comme dans des listes par exemple). De plus, elle met à jour la forme des agents clc en agrandissant soit la forêt soit le bâti à la place de l'ilot.

### AGENT

#### ***initialisationClc***

Action appelée dans l'action globale constructionCLC.

#### ***comportementAnnuel***

Action appelée chaque année depuis le main.

#### ***colorationClc***

Action appelée depuis le l'action interne comportementAnnuel. Elle va mettre à jour la couleur de l'agent clc. Avec l'augmentation des forêts ou du bâti, il faut mettre à jour la couleur au même moment.

#### ***basic***

Action d'affichage SIG appelée dans la classe gaml Experiment. Le clc est coloré en fonction de sa classe de couvert.

#### ***foretEtBatiAspect***

Action d'affichage SIG appelée dans la classe gaml Experiment. Seul la forêt et le bâti est coloré. Utile pour voir l'effet de la disparition des ilots.

***toString***

Action d'affichage des variables de l'agent dans la console.

## coursDeau.gaml

---

La classe cours d'eau correspond au cours d'eau principal de la zoneHydrographique, sans les ramifications.

### GLOBAL

#### *constructionCoursDeau*

Action appelée pendant la phase d'initialisation depuis le main. Elle crée les agents coursDeau à partir d'un fichier SIG d'entrée.

### AGENT

#### *basic*

Action d'affichage SIG appelée dans la classe gaml Experiment. Le cours d'eau est coloré en de la même couleur que tous les autres, et sa couleur n'évolue pas au cours du temps.

#### *evolutionDebitCoursDeauAspect*

Action d'affichage SIG appelée dans la classe gaml Experiment. La couleur de cet aspect est mise à jour depuis l'agent zoneHydrographique. Cette action colore le tronçon en fonction du débit de sa zoneHydrographique.

## **equipement.gaml**

---

Cette classe est la classe mère de tous les équipements de MAELIA, à savoir des équipements de captage (AEP, IND et IRR) et de rejet (AEP et IND). D'une manière générale, un volume annuel va être défini à partir de données d'entrée (pour l'AEP il y aura aussi la prise en compte de l'augmentation de la population, depuis les agents commune). Ce volume va ensuite être journalisé.

### **GLOBAL**

#### ***constructionEquipements***

Action appelée pendant la phase d'initialisation depuis le main. Elle fait appelle aux différentes méthodes de construction globales des agents equipementDeCaptageAEP, equipementDeCaptageIND, equipementDeCaptageIRR, equipementDeRejetAEP et equipementDeRejetIND.

#### ***calculVolumesEquipementsZoneMaelia [Indicateur]***

Action appelée tous les pas de temps depuis le main. Elle calcule la somme des volumes prélevés, rejetés et consommés au niveau de la zone d'étude.

#### ***calculVolumePreleveZoneMaelia [Indicateur]***

Action appelée depuis la méthode globale calculVolumesEquipementsZoneMaelia. Elle calcule la somme des volumes prélevés - séparément pour l'AEP, l'IND et l'IRR - au niveau de la zone d'étude. Pour ceci elle somme les volumes des différents équipements de captage.

#### ***calculVolumeRejetZoneMaelia [Indicateur]***

Action appelée depuis la méthode globale calculVolumesEquipementsZoneMaelia. Elle calcule la somme des volumes rejetés - séparément par l'AEP et l'IND - au niveau de la zone d'étude. Pour ceci elle somme les volumes des différents équipements de rejet.

#### ***calculVolumeConsommeZoneMaelia [Indicateur]***

Action appelée depuis la méthode globale calculVolumesEquipementsZoneMaelia. Elle calcule les volumes consommés - séparément par l'AEP, l'IND et l'IRR - au niveau de la zone d'étude. Pour ceci elle soustrait les rejets aux prélèvements.

### **AGENT**

#### ***initialisationEquipement***

Action générique appelée pendant l'initialisation à chaque création d'un nouveau type d'équipement.

#### ***comportementJournalier***

Action générique appelée tous les pas de temps depuis le main (pour les agents equipementDeCaptage) ou depuis les zoneHydrographique (pour les agents equipementDeRejet). Elle effectue une journalisation d'un volume annuel dans certains cas (pas pour l'IRR car les volumes prélevés sont décidés au niveau de chaque parcelle par l'agriculteur).

#### ***comportementAnnuel***

Action appelée tous les ans depuis le main. Elle met à jour le volume prévu annuel.

#### ***algoDeJournalisation***

Action appelée depuis la méthode interne comportementJournalier. Dans sa version générique, elle ramène le volume annuel prévu à un volume journalier en faisant une simple division par le nombre de jours dans l'année courante.

#### ***miseAJourVolumeZH***

Action appelée depuis la méthode interne `comportementJournalier`. Dans sa définition générique elle ne fait rien.

***calculVolumeSimuleAnnuel***

Action appelée depuis la méthode interne `comportementJournalier`. Elle met à jour une map contenant le volume simulé annuel.

***calculVolumePrevuAnnuel***

Action appelée depuis la méthode interne `comportementAnnuel`. Elle met à jour une map contenant le volume prévu annuel. Dans sa version générique elle stocke la même valeur chaque année correspondant à un volume de référence récupéré par les données d'entrée pour l'année 2010.

***journalisationAEP***

Action appelée depuis les équipements AEP (captage et rejet). Il journalise un volume annuel à l'aide d'un algorithme détaillé dans le document « [/Tasks/T2. Donnees/T2.2.98 - Algorithme de journalisation des series AEP/ Journalisation des series.pdf](#) ».

***basic***

Action d'affichage SIG appelée dans la classe `gaml Experiement`. Cet affichage est statique.

***toString***

Action d'affichage des variables de l'agent dans la console.

## équipementDeCaptage.gaml

---

Cette classe est la classe mère de tous les équipements de captage de MAELIA, que se soient pour l'AEP, l'IND ou l'IRR. Ainsi, il y a 3 types classes d'équipement possible : équipementDeCaptageAEP, équipementDeCaptageIND et équipementDeCaptageIRR. Elle étend la classe équipement.

### AGENT

#### ***miseAJourVolumeZH***

Action appelée depuis la méthode interne comportementJournalier. Dans sa définition générique elle ne fait rien.

#### ***imageAspect***

Action d'affichage SIG appelée dans la classe gaml Experiment. Cet affichage est statique, il met une image à l'endroit de chaque point de prélèvement.

## equipementDeCaptageAEP.gaml

Cette classe est la classe fille de l'agent equipementDeCaptage. Elle représente les équipements de captage pour l'eau potable (AEP).

### GLOBAL

#### ***constructionEquipementsDeCaptageAEP***

Action appelée depuis la méthode globale constructionEquipements de l'agent equipement. Elle crée et initialise les agents de type equipementDeCaptageAEP à l'aide de données SIG d'entrée.

#### ***initialisationChroniquesPointsDePrelevementAEP***

Action appelée depuis l'action globale constructionEquipementsDeCaptageAEP. Elle initialise les chroniques de prélèvements AEP par lecture d'un fichier csv. Le fichier d'entrée ne contient que les volumes pour une année de référence (2010). Ces volumes ne sont pas utilisés pour la dynamique du modèle (contrairement aux agents equipementDeCaptageIND) mais sont utilisés pour calculer l'importance de chaque point de prélèvement AEP au niveau de la zone d'étude.

#### ***initialisationPointsDePrelevementAEP***

Action appelée depuis l'action globale constructionEquipementsDeCaptageAEP. Elle détermine l'importance (taux) de chaque point de prélèvement AEP au niveau de la zone d'étude. Par exemple, si un point a un volume de référence de 1L, et si la somme des volumes de référence de tous les équipements de captage AEP vaut 10L. Alors le point de prélèvement aura un taux d'importance de 0,1 au niveau de la zone d'étude. Ce taux va être utile pour connaître le volume annuel prévu pour chaque point de prélèvement AEP.

### AGENT

#### ***calculVolumePrevuAnnuel [Overwrite]***

Action appelée depuis la méthode interne comportementJournalier. Elle calcule le volume prévu à prélever pour l'année à venir. Ce calcul se base sur le volume consommé au niveau de la zone d'étude calculé dans la section globale de l'agent commune. Ce volume est ramené au point de prélèvement AEP en appliquant le taux calculer dans initialisationPointsDePrelevementAEP. Et il est ramené à un volume prélevé en appliquant un coefficient prédéfini (avec MOGIRE). Pour plus d'information voir le document « [/Tasks/T4.Processus/T4.3.06.b - Consommation eau potable \(AEP-MOGIRE\)/ consommation AEP MOGIRE.pdf](#) ».

#### ***algoDeJournalisation [Overwrite]***

Action appelée depuis la méthode interne comportementJournalier. Elle journalise le volume annuel prévu en appliquant l'action journalisationAEP, définie dans equipement.

#### ***miseAJourVolumeZH [Overwrite]***

Action appelée depuis la méthode interne comportementJournalier. Elle informe la zoneHydrographique associée quel volume l'agent souhaite prélever ce jour-ci. La valeur réelle sera alors calculée et mise à jour au moment du processus « écoulement de l'eau » par la zoneHydrographique.

#### ***toString [Overwrite]***

Action d'affichage des variables de l'agent dans la console.

## **equipementDeCaptageIND.gaml**

---

Cette classe est la classe fille de l'agent equipementDeCaptage. Elle représente les équipements de captage pour les industries (IND).

### **GLOBAL**

#### ***constructionEquipementsDeCaptageIND***

Action appelée depuis la méthode globale constructionEquipements de l'agent equipement. Elle crée et initialise les agents de type equipementDeCaptageIND à l'aide de données SIG d'entrée.

#### ***initialisationChroniquesPointsDePrelevementIND***

Action appelée depuis l'action globale constructionEquipementsDeCaptageIND. Elle initialise les chroniques de prélèvements IND par lecture d'un fichier csv. Le fichier d'entrée ne contient que les volumes pour une année de référence (2010). Ces volumes vont servir de base pour la dynamique du modèle de prélèvement IND. On suppose donc que le volume annuel prélevé pour l'industrie est constant.

### **AGENT**

#### ***miseAJourVolumeZH [Overwrite]***

Action appelée depuis la méthode interne comportementJournalier. Elle informe la zoneHydrographique associée quel volume l'agent souhaite prélever ce jour-ci. La valeur réelle sera alors calculée et mise à jour au moment du processus « écoulement de l'eau » par la zoneHydrographique.

## **equipementDeCaptageIRR.gaml**

---

Cette classe est la classe fille de l'agent equipementDeCaptage. Elle représente les équipements de captage pour les agriculteurs (IRR).

### **GLOBAL**

#### ***constructionEquipementsDeCaptageIRR***

Action appelée depuis la méthode globale constructionEquipements de l'agent equipement. Elle crée et initialise les agents de type equipementDeCaptageIRR à l'aide de données SIG d'entrée.

#### ***initialisationChroniquesPointsDePrelevementIRR***

Action appelée depuis l'action globale constructionEquipementsDeCaptageIRR. Elle initialise les chroniques de prélèvements IND par lecture d'un fichier csv. Le fichier d'entrée ne contient que les volumes pour une année de référence (2010). Ces volumes vont servir de base pour la dynamique du modèle de prélèvement IND. On suppose donc que le volume annuel prélevé pour l'industrie est constant.

### **AGENT**

#### ***comportementJournalier [Overwrite]***

Action appelée tous les pas de temps depuis le main. Etant donné que le volume prévu est calculé par les agriculteurs, la seule raison d'être de cette action est de mettre à jour la map du volume simulé annuel.

#### ***miseAJourVolumeZH [Overwrite]***

Action appelée depuis la méthode interne comportementJournalier. Elle informe la zoneHydrographique associée quel volume l'agent souhaite prélever ce jour-ci. La valeur réelle sera alors calculée et mise à jour au moment du processus « écoulement de l'eau » par la zoneHydrographique.

#### ***miseAzeroVolumeJournalier***

Action appelée en début de chaque pas de temps depuis le main. Elle remet à zéro certaines variables.

## equipementDeRejet.gaml

---

Cette classe est la classe mère de tous les équipements de rejet de MAELIA, que se soient pour l'AEP ou l'IND (il n'y a pas de rejet IRR). Ainsi, il y a 3 types classes d'équipement possible : equipementDeRejetAEP et equipementDeRejetIND. Elle étend la classe equipement.

### AGENT

#### ***miseAJourVolumeZH***

Action appelée depuis la méthode interne comportementJournalier. Chaque agent de type equipementDeRejet va donner l'eau rejetée à sa zoneHydrographique associée.

## **equipementDeRejetAEP.gaml**

---

Cette classe est la classe fille de l'agent equipementDeRejet. Elle représente les équipements de rejet pour l'eau potable (AEP).

### GLOBAL

#### ***constructionEquipementsDeRejetAEP***

Action appelée depuis la méthode globale constructionEquipements de l'agent equipement. Elle crée et initialise les agents de type equipementDeRejetAEP à l'aide de données SIG d'entrée.

#### ***initialisationChroniquesPointsDeRejetAEP***

Action appelée depuis l'action globale constructionEquipementsDeRejetAEP. Elle initialise les chroniques de rejet AEP par lecture d'un fichier csv. Le fichier d'entrée ne contient que les volumes pour une année de référence (2010). Ces volumes ne sont pas utilisés pour la dynamique du modèle (contrairement aux agents equipementDeRejetIND) mais sont utilisés pour calculer l'importance de chaque point de rejet AEP au niveau de la zone d'étude.

#### ***initialisationPointsDeRejetAEP***

Action appelée depuis l'action globale constructionEquipementsDeRejetAEP. Elle détermine l'importance (taux) de chaque point de rejet AEP au niveau de la zone d'étude. Par exemple, si un point a un volume de référence de 1L, et si la somme des volumes de référence de tous les équipements de rejet AEP vaut 10L. Alors le point de rejet aura un taux d'importance de 0,1 au niveau de la zone d'étude. Ce taux va être utile pour connaître le volume annuel à rejeter prévu pour chaque point de rejet AEP.

### AGENT

#### ***calculVolumePrevuAnnuel [Override]***

Action appelée depuis la méthode interne comportementJournalier. Elle calcule le volume prévu à rejeter pour l'année à venir. Ce calcul se base sur le volume consommé au niveau de la zone d'étude calculé dans la section globale de l'agent commune. Ce volume est ramené au point de rejet AEP en appliquant le taux calculé dans initialisationPointsDeRejetAEP. Et il est ramené à un volume à rejeter en appliquant un coefficient prédéfini (avec MOGIRE). Pour plus d'information voir le document « [/Tasks/T4. Processus/T4.3.06.b - Consommation eau potable \(AEP-MOGIRE\)/ consommation AEP MOGIRE.pdf](#) ».

***algoDeJournalisation [Override]*** Action appelée depuis la méthode interne comportementJournalier. Elle journalise le volume annuel prévu en appliquant l'action journalisationAEP, définie dans equipement.

## **equipementDeRejetIND.gaml**

---

Cette classe est la classe fille de l'agent equipementDeRejet. Elle représente les équipements de rejet pour les industries (IND).

### **GLOBAL**

#### ***constructionEquipementsDeRejetIND***

Action appelée depuis la méthode globale constructionEquipements de l'agent equipement. Elle crée et initialise les agents de type equipementDeRejetIND à l'aide de données SIG d'entrée.

#### ***initialisationPointsDeRejetIND***

Action appelée depuis l'action globale constructionEquipementsDeRejetIND. Elle initialise les agents de type equipementDeRejetIND, pour ce faire elle lit un fichier csv donnant le lien entre les points de prélèvements IND et ceux de rejet. Les agents de type equipementDeRejetIND ont donc une information supplémentaire qui est leur equipementDeCaptage associé.

### **AGENT**

#### ***calculVolumePrevuAnnuel [Override]***

Action appelée depuis la méthode interne comportementJournalier. Elle calcule le volume prévu à rejeter pour l'année à venir. Pour se faire elle applique un coefficient prédéterminé au volume de son point de prélèvement associé.

***algoDeJournalisation [Override]*** Action appelée depuis la méthode interne comportementJournalier. Elle journalise le volume annuel prévu en divisant le volume prévu par le nombre de jour dans l'année. Ici l'action est redéfinie (alors qu'elle avait été définie de manière générique dans equipement) car on doit s'assurer que la quantité rejetée est en cohérence avec la quantité prélevée. Pour se faire on applique un coefficient calculé dans la zoneHydrographique au moment du processus « écoulement de l'eau ».

**grosseIndustrie.gaml**

---

Pas encore utilisé.

## hru.gaml

Une HRU (ou Hydrographic Response Unit) est un découpage de zoneHydrographique en fractions non géo-référencées. Dans une zoneHydrographique il y aura autant de HRUs qu'il y a de combinaisons pente, typeDeSol, clc possibles et dont l'intersection existe. Pour des raisons de conformité au modèle, tout est exprimé en millimètre au sein des HRU, la conversion en mètre se fait ensuite au niveau des zoneHydrographique. Remarque : Le clc (ou couvert) pour créer les agents hru n'inclue pas dans son clc agricole les surfaces de ilots du RPG. Celles-ci seront traitées à part dans les agents hruRPG. Agents qui étendent la classe hru. Les HRUs ne sont pas des entités géo-référencées, mais si on fait la somme de toutes les surfaces des HRUs et des hruRPG au sein d'une zoneHydrographique, le résultat sera égal à la surface de cette dernière.

### GLOBAL

#### **Attributs**

- hruShape - Chemin vers le fichier SHP
- hruSansIlotsShape - Chemin vers le fichier SHP (dans le cas où on exécute aussi le modèle agricole)
- listeHrus [hrus]

#### **constructionHRUs**

Action appelée à l'initialisation depuis le main. Elle construit les agents hru pour TOUTES les combinaisons de pente, typeDeSol, clc possible, et ce pour tous les agents zoneHydrographique. Pour chacune de ces combinaisons, une surface (ou une fraction de ZH) est associée. Il est tout à fait possible que certaines des HRUs créées aient une surface nulle (ceci étant du au fait que la combinaison ne correspond à aucune intersection possible dans la ZH). Ces HRUs ne seront alors pas utiles la première année. Cependant, le processus « disparition des ilots » va agir sur le clc des zoneHydrographique, cela signifie que les HRUs peuvent évoluer au cours du temps. Cette évolution se fait généralement par une simple augmentation de sa surface, mais elle peut aussi amener à l'apparition d'une nouvelle HRUs. Le fait de les créer toutes à l'initialisation permet de ne pas avoir le faire en cours de simulation. Cela n'augmente pas les temps de calcul, car les HRUs ne correspondant à aucune intersection dans la ZH est inactive jusqu'à ce qu'elle soit utile (ce qui peut ne jamais arriver). Remarque : pour le moment la variable « pente » n'est pas prise en compte dans la création des HRUs par manque de données.

### AGENT

#### **Attributs**

- couleurHRU
- idHRU
- idZhAssociee
- zoneHydrographiqueAssociee [zoneHydrographique]
- idTypeDeSolAssocie
- typeDeSolAssocie [typeDeSol]
- clcAssocie [clc]
- idClasseClcAssocie
- penteAssociee
- fractionDansZH
- surface [m2]
- surfaceKm2 [km2]
- isHRUimpermeable
- mapTeneurEnEauSolParCouche [mm] - (SW,ly)
- mapTeneurEnEauTotalParJour [mm] - (SW,tot)
- quantiteEauTotalPhaseSol [mm] - (Qhru)

- perteParTransmission [mm] - (tlss,surq)
- ruissellementDeSurfaceHRU [mm] - (Qday)
- ruissellementDeSurfaceHRUtotal [mm] - (Qsurf)
- ruissellementDeSurfaceHRUStockeeJourPrecedent [mm] - (Qstor(j-1))
- parametreDeRetention [mm] - (S)
- curveNumber - (CN)
- ecoulementLateral [mm] - (latDay)
- ecoulementLateralStockeeJourPrecedent [mm] - (latStor(j-1))
- mapEcoulementLateralParCouche [mm] - (Qlat,ly)
- mapPercolationParCouche - (omegaPerc,ly)
- lai - (LAI)
- evapoTranspirationReelle [mm] - (Ea)
- ecoulementEauSouterraine [mm] - (Qgw)
- eauEntreeAquiferes [mm] - (omegaRchrg)
- eauDerniereCouche [mm] - (omegaSeep)
- eauStockeeAquiferePeuProfond [mm] - (aqSh)
- hauteurNiveauHydrostatique [m] - (hWtbl)
- eauPompeeAquiferePeuProfond [mm] - (omegaPumps,sh)
- longueurCoursEauTributaireMaxHRU [km] - (LChTri)
- largeurMoyenneCoursEauTributaireHRU [km] - (WchTri)
- ratioStockageHRU - (KchTri)
- tempsDeConcentrationHRU - (tConc)
- fractionRuissellement - (brt)
- fractionEcoulementDeSurface - (FrTtlag)
- parametreDeRetentionMax [mm] - (Smx)
- coefDeFormePourRetention1 - (omega1)
- coefDeFormePourRetention2 - (omega2)
- retardEntreSortiSolEtEntreeAquifere [jr] - (dioptriGw)
- coefRevapEauSouterraine - (betaRev)
- coefPercolationVersAquifereProfond - (betaDeep)
- recessionEcoulementEauxSouterraines [mm] - (alphaGw)
- seuilEcoulementAquiferePeuProfond [mm] - (aqShthr)
- rendementAquiferePeuProfond [m3/m3] - (mu)

### ***initialisationHRU***

Action appelée depuis la méthode globale constructionHRUs. Elle initialise différentes variables propres au modèle SWAT.

### ***calculRuissellementDeSurface***

Action appelée depuis l'agent `zoneHydrographiqueSWAT`. Elle calcule la quantité d'eau de ruissellement que produit l'agent HRU et qui part vers le `coursDeau` principale de sa `zoneHydrographiqueSWAT` associée. Ce calcul se fait pendant la phase sol du modèle SWAT. Pour plus d'information se référer au document « [/Tasks/T4. Processus/T4.3.01 - Ecoulement de l'eau/Module\\_Hydro\\_MAELIA.pdf](#) ».

### ***calculPerteParTransmissionDuRuissellementDeSurface***

Action appelée depuis la méthode interne calculRuissellementDeSurface. Elle calcule la perte par transmission du ruissellement de surface.

### ***calculEcoulementLateral***

Action appelée depuis l'agent `zoneHydrographiqueSWAT`. Elle calcule la quantité d'eau d'écoulement latéral qui va s'infiltrer dans le sol de l'agent HRU et dont une partie va être redirigé vers le `coursDeau`

principale de sa zoneHydrographiqueSWAT associée. Ce calcul se fait pendant la phase sol du modèle SWAT. Pour plus d'information se référer au document « [/Tasks/T4. Processus/T4.3.01 - Ecoulement de l'eau/Module\\_Hydro\\_MAELIA.pdf](#) ».

#### ***calculEvapotranspirationReelle***

Action appelée depuis l'agent zoneHydrographiqueSWAT. Elle calcule la quantité d'eau d'évapotranspiration qui va être perdu par l'agent HRU. Cette perte se fait par l'évaporation de la canopée, et par la transpiration, sublimation et évaporation du sol. Ce calcul se fait pendant la phase sol du modèle SWAT. Pour plus d'information se référer au document « [/Tasks/T4. Processus/T4.3.01 - Ecoulement de l'eau/Module\\_Hydro\\_MAELIA.pdf](#) ».

#### ***evaporationZ***

Action appelée depuis la méthode interne `calculEvapotranspirationReelle`. Elle calcule l'évaporation selon z.

#### ***transpirationZ***

Action appelée depuis la méthode interne `calculEvapotranspirationReelle`. Elle calcule la transpiration selon z.

#### ***miseAJourEauEntreePourCalculEcoulementSouterrain***

Action appelée depuis la méthode interne `calculEcoulementEauSouterraine`. Elle calcule l'eau qui sera disponible en entrée de l'aquifère peu profond, en vue du calcul de l'écoulement souterrain.

#### ***calculEcoulementEauSouterraine***

Action appelée depuis l'agent zoneHydrographiqueSWAT. Elle calcule la quantité d'eau d'écoulement souterrain que va rejeter l'agent HRU vers le coursDeau principale de sa zoneHydrographiqueSWAT associée. Ce calcul se fait pendant la phase sol du modèle SWAT. Pour plus d'information se référer au document « [/Tasks/T4. Processus/T4.3.01 - Ecoulement de l'eau/Module\\_Hydro\\_MAELIA.pdf](#) ».

#### ***calculTotalPhaseSol***

Action appelée nul part. Elle fait le bilan de la phase sol.

#### ***verificationBalanceEau***

Action appelée depuis l'agent zoneHydrographiqueSWAT. Elle calcule la balance d'eau. Elle renvoie vrai si l'eau en entrée de la HRU correspond à l'eau en sortie ajoutée à l'eau stockée.

#### ***remiseAzero***

Action appelée depuis l'agent zoneHydrographiqueSWAT. Dans sa version générique elle ne fait rien.

#### ***toString***

Action d'affichage des variables de l'agent dans la console.

## **hruRPG.gaml**

---

La classe `hruRPG` étend la classe `hru`. Une `hruRPG` (ou Hydrographic Response Unit) est un découpage de `zoneHydrographique` en fractions non géo-référencées. Dans une `zoneHydrographique` il y aura autant de `hruRPGs` qu'il y a de combinaisons pente, `typeDeSol` possibles et dont l'intersection existe. Pour les `hruRPG`, seul le `clc` agricole occupé par les agents `ilot` est pris en compte, ramenant le nombre de `clc` possible à 1. Pour plus d'information sur ce sujet se référer au document « [/Tasks/T4. Processus/T4.3.01 - Ecoulement de l'eau/Ecoulement de l'eau\\_RPG/ T4.3.01 - Lien SWAT\\_RPG.pdf](#) ».

### **GLOBAL**

#### ***constructionHRUsRPG***

Action appelée à l'initialisation depuis le `main`. Elle crée les agents `hruRPG` en se basant sur les intersections entre les ilots, les types de sol et les pentes au sein d'une même `zoneHydrographique`. Contrairement à la création des agents `hru`, seules sont créées les `hruRPG` existantes. En effet, le processus de « disparition des ilots » va avoir pour seul effet la diminution des surfaces des `hruRPG`, et au pire leur suppression. Mais il ne pourra pas y avoir de nouvelles `hruRPG` créées au cours de la simulation. Chaque agent `hruRPG` connaît ses agents `ilot`.

### **AGENT**

#### ***initialisationHRU [Overwrite]***

Action appelée depuis la méthode globale `constructionHRUsRPG`. Elle initialise le lien entre la `hruRPG` et les agents `hru` qui vont potentiellement grandir avec le processus de « disparition des ilots ». Ainsi, la `hruRPG` est liée au plus à 2 agents `hru` : les 2 auront le même `typeDeSol` et la même pente, mais l'une aura un `clc` de type bâti alors que l'autre aura un `clc` de type forêt.

#### ***miseAJourFractionHRUs***

Action appelée depuis sa `zoneHydrographiqueSWAT` associée. Elle met à jour la surface (ou fraction) de l'agent dans le cas où un de ses ilots venait à disparaître. Par exemple, si l'un des ilots de la `hruRPG` disparaît au profit de la forêt, la surface de la `hruRPG` sera diminuée de la surface de l'`ilot` alors que celle de la HRU associée de type forêt verra sa surface augmentée de la surface de l'`ilot` disparu. Remarque : cette action doit être appelée après le choix des ilots à disparaître, mais avant leur suppression, car si l'`ilot` n'existe plus il n'est alors plus possible de mettre à jour la `hruRPG` (on a plus accès à la valeur de sa surface)..

#### ***croissancePlanteHRU***

Action appelée depuis sa `zoneHydrographiqueSWAT` associée. Elle déclenche le processus de « croissance des plantes » des ilots lui appartenant. Ce processus est appelé avant le calcul de la phase sol des agents `hruRPG` et après celui des agents `hru` car les ilots ont besoin de connaître la quantité d'eau réelle que l'agriculteur a utilisé pour l'irrigation des cultures. Pour plus d'information sur ce sujet se référer au document « [/Tasks/T4. Processus/T4.3.01 - Ecoulement de l'eau/Ecoulement de l'eau\\_RPG/ T4.3.01 - Lien entre SWAT et les prelevements.pdf](#) ». Remarque : le processus « croissance des plantes » est appelé depuis la `zoneHydrographique` uniquement dans le cas où le modèle hydrographique est SWAT, ce qui est forcément le cas si des agents `hru` et `hruRPG` sont créés.

#### ***calculRuissellementDeSurface [Overwrite]***

Action appelée depuis l'agent `zoneHydrographiqueSWAT`. Elle renvoie la quantité d'eau de ruissellement que produisent les ilots de la `hruRPG` et qui part vers le `coursDeau` principale de sa `zoneHydrographiqueSWAT` associée. Cette quantité d'eau est calculée par les agents `parcelle` des ilots. C'est l'`ilot` qui va mettre à jour la variable « ruissellement de surface » de la `hruRPG` au moment du processus de « croissance des plantes ». Cette action fait juste une conversion de volume (que renvoient les ilots) à une hauteur (utile pour SWAT).

***calculEcoulementLateral [Override]***

Action appelée depuis l'agent zoneHydrographiqueSWAT. Dans sa version redéfinie pour les agents hruRPG elle ne fait rien. Les ilots ne retournent pas de quantité d'eau sous forme d'un écoulement latéral. On suppose que l'eau qui s'infiltré dans le sol est entièrement utilisée pour le calcul de l'écoulement souterrain.

***calculEvapotranspirationReelle [Override]***

Action appelée depuis l'agent zoneHydrographiqueSWAT. Elle renvoie la quantité d'eau d'évapotranspiration que les ilots de la hruRPG génèrent. Cette quantité d'eau est calculée par les agents parcelle des ilots. C'est l'ilote qui va mettre à jour la variable « évapotranspiration réelle » de la hruRPG au moment du processus de « croissance des plantes ». Cette action fait juste une conversion de volume (que renvoient les ilots) à une hauteur (utile pour SWAT).

***miseAJourEauEntreePourCalculEcoulementSouterrain [Override]***

Action appelée depuis la méthode interne calculEcoulementEauSouterrain. Elle calcule l'eau qui sera disponible en entrée de l'aquifère peu profond, en vue du calcul de l'écoulement souterrain. Pour les agents hruRPG, on suppose que la quantité d'eau « drain » renvoyé par les ilots correspond à ce qui rentre dans l'aquifère peu profond. Le calcul de l'écoulement souterrain est alors identique à celui pour les agents hru classiques.

***remiseAzero [Override]***

Action appelée depuis l'agent zoneHydrographiqueSWAT. Elle remet à zéro certaines variables.

***toString [Override]***

Action d'affichage des variables de l'agent dans la console.

## **lac.gaml**

---

Pas encore utilisé.

## mnt.gaml

---

Pas encore utilisé.

## nappePhreatique.gaml

---

Pas encore utilisé.

## noeudHydrographique.gaml

---

Cette classe ne fait pas parti du DAR, mais elle représente néanmoins une entité indispensable du modèle hydrographique. C'est en effet via les agents noeudHydrographique que le lien d'amont en aval des zoneHydrographique est réalisé.

### GLOBAL

#### *constructionNoeudHydrographique*

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise les agents de type noeudHydrographique à partir d'un fichier SIG d'entrée.

### AGENT

#### *basic*

Action d'affichage SIG appelée dans la classe gaml Experiment. Cet affichage est statique.

## **ressourceEnEau.gaml**

---

Cette classe mère représente englobe tous les types de ressources en eau, à savoir : les cours d'eau, les nappes et les retenues collinaires.

### **GLOBAL**

#### ***Attributs***

- mapRessourcesEnEau - Liste globale des différents types de ressources (cours d'eau, nappe ou retenue) avec en clef le type de la ressource.

### **AGENT**

#### ***Attributs***

- id
- natureRessourcePrelevee
- zhAssociee [zoneHydrographique]
- mapEquipementsCaptageAssocies [equipementDeCaptage par acteur AEP, INID ou IRR]
- volumeUtileAvantPrelevementEtRejet [m3]
- mapVolumePreleveSouhaite [volume par acteur AEP, INID ou IRR]
- mapVolumePreleveReel [volume par acteur AEP, INID ou IRR]
- couleurRessource

#### ***initialisation***

Action générique appelée pendant la phase d'initialisation depuis les méthodes globales de création des agents fils de type ressourceEnEau.

#### ***miseAzero***

Action appelée en début de chaque pas de temps depuis la méthode globale remiseAzeroGlobale de la classe main.

#### ***get VolumePreleve***

Accesseur qui renvoie le volume prélevé souhaité ou réel selon l'argument spécifié en entré.

#### ***get VolumePreleveSouhaite***

Accesseur qui renvoie le volume prélevé souhaité sur la ressource en début de chaque pas de temps.

#### ***get VolumePreleveReel***

Accesseur qui renvoie le volume prélevé réel sur la ressource, si la zoneHydrographique a un débit insuffisant, le débit réel sera inférieur au débit souhaité (il ne peut logiquement pas être supérieur).

#### ***get VolumeUtileAprèsPrelevementEtRejet***

Accesseur qui renvoie le volume restant sur la ressource une fois tous les prélèvements effectués au niveau de la zoneHydrographique associée à celle-ci.

#### ***get VolumeUtileAvantPrelevementEtRejet***

Accesseur qui renvoie le volume présent sur la ressource avant les prélèvements du jour courant.

#### ***setMap VolumePreleveSouhaite***

*ajouterEquipementDeCaptage*

*calculVolumePrelevesReels*

*basic*

Action d'affichage SIG appelée dans la classe gaml Experiment. Cet affichage est statique.

*toString*

Action d'affichage des variables de l'agent dans la console.

**retenueCollinaire.gaml**

---

Pas encore utilisé.

## **zoneHydrographique.gaml**

---

Cette classe est l'entité de référence du modèle hydrographique, elle représente l'entité géo-référencée de base, la résolution du réseau hydrographique. Les valeurs de débit ne seront pas calculées au dessous de cette référence spatiale.

### **GLOBAL**

#### **Attributs**

- zoneHydrographiqueShape - Chemin vers le fichier SHP
- mapZH
- mapArbreZH
- listeZonesHydrographiques [zonesHydrographiques]
- listeZonesHydrographiquesHierarchisees [zonesHydrographiques]
- volumeUtilePourPrelevementsCoursEauZM [m3]
- volumePrelevementSouhaitesZoneMaelia [m3]
- volumePrelevementReelsZoneMaelia [m3]
- mapVolumePrelevementSouhaitesZoneMaelia
- mapVolumePrelevementReelsZoneMaelia

#### **creationZoneHydrographique**

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise les agents de type zoneHydrographique si le modèle hydrographique choisi est le simple. Dans le cas où le modèle hydrographique choisi est SWAT, cette action va alors créer les agents de type zoneHydrographiqueSWAT. Enfin, dans le cas où le modèle agricole choisi est JeuOC, cette action va créer les agents de type zoneHydrographiqueJeuOC. Dans tous les cas, ces agents sont créés à partir d'un fichier SIG d'entrée. Les agents zone hydrographique sont créés très tôt car ils représentent la référence spatiale, c'est donc eux qui dictent la forme de la zone d'étude dont tous les autres agents (géo-référencés) dépendent.

#### **initialisationZonesHydrographiques**

Action appelée pendant la phase d'initialisation depuis le main. Elle se fait séparément de la création car certains agents tels que les agents nÅŞudHydrographique ou encore typeDeSol doivent déjà avoir été construits. Cette action initialise également le lien amont / aval entre les agents zoneHydrographique eux mêmes.

#### **initialisationZonesHydrographiquesLienRPG**

Action appelée pendant la phase d'initialisation depuis le main. Elle se fait séparément de la première initialisation car les agents ilot doivent déjà avoir été construits (or ils le sont après l'initialisation du modèle hydrographique). Cette action initialise entre autre le lien entre la zone hydrographique et les agents ilot.

#### **creationMapArbreZHEtNiveau**

Action appelée depuis l'action globale initialisationZonesHydrographiques. Elle associe à chaque zone hydrographique un niveau hiérarchique correspondant à sa place dans le réseau hydrographique. Plus le niveau est faible, plus la zone hydrographique se rapproche de l'exutoire de la zone d'étude. Au final, cette action crée une liste de zone hydrographique triées selon leur niveau hiérarchique. Cette liste indique l'ordre d'exécution des zones hydrographiques, pour que celles-ci s'exécutent d'amont en aval. Ainsi, les zones hydrographiques ayant le plus haut niveau hiérarchique seront exécutées en premier, et l'exutoire en dernier.

#### **affichageDebitMax [Indicateur]**

Action appelée depuis le toStringGlobal du main. Elle affiche uniquement dans la console le débit maximum, toutes zones hydrographiques confondues.

#### **calculSurfaceTotaleZHs [Indicateur]**

Action appelée nul part. Elle calcule uniquement la surface totale de la zone d'étude.

### ***calculVolumesPrelevementsSouhaitesVSreels [Indicateur]***

Action appelée à chaque pas de temps depuis le main. Elle fait une synthèse au niveau de la zone d'étude des prélèvements AEP, IND et IRR souhaités par rapport aux réels. Ces valeurs sont stockées et peuvent être affichées sous forme de graphiques dans l'interface graphique.

## AGENT

### ***Attributs***

- idZoneHydrographique
- idExutoire
- listeIlotsAssocies [ilots]
- listeTypeDeSolAssocies [typesDeSol]
- listePentesAssocies [pentes]
- landCoverAssocie [clcs]
- listeZonesHydrographiquesAmonts [zonesHydrographiques]
- zoneHydrographiqueAval [zoneHydrographique]
- exutoire [noeudHydrographique]
- pointDentree [noeudHydrographique]
- zoneMeteoMoyenneAssociee [zoneMeteo]
- ressourceEnEauAssociees [ressourceEnEau]
- surfaceIlotsTotale [m2]
- precipitationsTotalesSurZH [m]
- etpTotalSurZH [m]
- debitCourant [m/s]
- volumeEntree [m3]
- volumeSorti [m3]
- volumeEntrantDesZHsAmonts [m3]
- isDebitNegatifPasDeTempsCourant
- niveauHierarchiqueArbreZH
- volumeLacherBarrage [m3]
- volumeRuissellementDeSurfaceRPG [m3]
- volumeEcoulementLateralRPG [m3]
- volumeEvapotranspirationRPG [m3]
- volumeEcoulementEauSouterraineRPG [m3]
- couleurZoneHydrographique
- couleurEvolutionTemperature
- couleurCoursDeauZoneHydrographique
- couleurDebitZoneHydrographique
- couleurEvolutionPrecipitation

### ***comportementJournalier***

Action appelée chaque pas de temps depuis le main. Elle fait appelle à différentes sous méthodes. Elle va ainsi mettre à jour les données liées à la pluie, faire le processus « d'écoulement de l'eau », calculer les prélèvements possibles ainsi que les rejets et enfin mettre à jour les différents affichages possibles pour l'agent géo-référencé zoneHydrographique (température, débit, etc).

### ***comportementAnnuel***

Action appelée chaque année depuis le main. Elle fait appelle à un méthode qui va mettre à jour la liste des hru actives de la zone hydrographique.

### ***initialisationZoneHydrographique***

Action appelée depuis l'action globale `initialisationZonesHydrographiques`. Elle fait appelle à une méthode générique d'initialisation. Il a fallut user de ce subterfuge (créer 2 méthodes d'initialisation alors qu'en apparence une seule aurait suffit) car cela évite d'avoir une duplication de code et permet de garder les mêmes interfaces pour la classe fille `zoneHydrographiqueSWAT`.

### ***initialisation***

Action appelée depuis l'action interne `initialisationZoneHydrographique`. Elle initialise les nÂŞuds et les types de sol de l'agent `zoneHydrographique`.

### ***initialisationLienEntreZonesHydro***

Action appelée depuis l'action interne `initialisationZoneHydrographique`. Elle initialise LA zone hydrographique aval et la ou les zones hydrographiques amont(s) de l'agent courant.

### ***initialisationLienRPG***

Action appelée depuis l'action globale `initialisationZonesHydrographiquesLienRPG`. Cette action initialise entre autre le lien entre la zone hydrographique et les agents `ilot`, elle initialise également ses équipements et son `clc`. Pour ce dernier, chaque agent `zoneHydrographique` a des agents `clc` dont on a retranché la forme des ilots appartenant à l'agent.

### ***remplissageReserveDeauZH***

Action appelée à chaque fois que de l'eau arrive d'un agent extérieur vers la zone hydrographique courante. Elle est ainsi appelée par les agents `equipementDeRejet`, et par les agents `ilot` dans le cas où le modèle hydrographique n'est pas SWAT (car alors les ilots ruissellent vers leur `hruRPG` respective).

### ***precipitationsEtETP***

Action appelée depuis la méthode interne `comportementJournalier`. Elle met à jour les données météo de l'agent (précipitation et ETP).

### ***phaseSol***

Action appelée depuis la méthode interne `comportementJournalier`. Dans sa version générique (donc si le modèle hydrographique est le modèle « simple »), elle considère que l'eau générée pendant la phase sol provient uniquement de la pluie, sans prise en compte d'une quelconque évaporation.

### ***phaseSolRPG***

Action appelée depuis la méthode interne `comportementJournalier`. Dans sa version générique (donc si le modèle hydrographique est le modèle « simple »), elle ne fait rien.

### ***miseAzeroVolume***

Action appelée en début de chaque pas de temps depuis le `main`. Elle remet à zéro certaines variables le nécessitant.

### ***calculVolumeUtileCoursEauReel***

Action appelée depuis la méthode interne `calculVolumePrelevsReels`. Elle calcule l'eau disponible dans le `coursDeau` principal pour les prélèvements en eau de surface. Ce volume disponible est calculé en fonction de l'eau qui arrive de la (ou des) zone hydrographique en amont ainsi que de l'eau générée par la phase sol des agents `hru` classiques (avant la phase sol des agents `hruRPG`). Pour plus d'information sur ce sujet se référer au document « /Tasks/T4. Processus/T4.3.01 - Ecoulement de l'eau/Ecoulement de l'eau\_RPG/ T4.3.01 - Lien entre SWAT et les prelevements.pdf ».

### ***calculVolumeUtileNappesReel***

Action appelée depuis la méthode interne calculVolumePrelevésReels. Elle calcule l'eau disponible dans les nappes pour les prélèvements en nappes. Dans sa version générique (donc si le modèle hydrographique est le modèle « simple »), il est supposé qu'il n'y a pas d'eau disponible.

### ***calculVolumeUtileRetenuesReel***

Action appelée depuis la méthode interne calculVolumePrelevésReels. Elle calcule l'eau disponible dans les retenues pour les prélèvements en retenues. Dans sa version générique (donc si le modèle hydrographique est le modèle « simple »), il est supposé qu'il n'y a pas d'eau disponible. Remarque : les prélèvements en retenues ne sont pas encore modélisés.

### ***calculVolumePrelevésReels***

Action appelée depuis la méthode interne comportementJournalier. Elle fait appelle aux différentes méthodes permettant de calculer les volumes disponibles en eau de surface, en nappes, ainsi que dans les retenues de la zone hydrographique courante. Elle fait ensuite appelle à une méthode qui va calculer les volumes effectivement prélevés par chaque agent equipementDeCaptage lui appartenant.

### ***calculVolumePrelevésReelsEnFonctionRessource***

Action appelée depuis la méthode interne calculVolumePrelevésReels. En fonction du type de prélèvement indiqué en paramètre d'entrée (SURF, NAPP ou RET), elle va calculer pour chaque type d'equipementDeCaptage (AEP, IND et IRR) la quantité d'eau qu'ils pourront prélevés. Il faut savoir que les équipements de captage pour l'eau potable sont prioritaires sur les équipements des industries qui eux même sont prioritaires sur ceux de l'irrigation dans le cas où il n'y aurait pas suffisamment d'eau disponible pour tous. Pour plus d'information sur ce sujet se référer au document « [/Tasks/T4. Processus/T4.3.01 - Ecoulement de l'eau/Ecoulement de l'eau\\_RPG/ T4.3.01 - Lien entre SWAT et les prelevements.pdf](#) ».

### ***miseAJourPrelevementsReels***

Action appelée depuis la méthode interne calculVolumePrelevésReels. En fonction du type de prélèvement indiqué en paramètre d'entrée (SURF, NAPP ou RET), elle va calculer pour chaque type d'equipementDeCaptage (AEP, IND et IRR) la somme des prélèvements ?

### ***calculRejets***

Action appelée depuis la méthode interne comportementJournalier. Une fois que les prélèvements humains ont été calculés, cette action calcule les rejets effectifs de chaque type d'equipementDeRejet (AEP et IND). Ces rejets sont normalement calculés à partir d'une valeur annuelle qui est journalisée. Ils ne prennent donc pas en compte les prélèvements réels et sont calculés dans le cas où il y a toujours assez d'eau dans la zone hydrographique. Cette action va appliquer un pourcentage à ces rejets pour les rendre cohérents (au niveau de la ZH) aux prélèvements. En effet, le lien entre équipements de captage et de rejet étant inconnu, la cohérence ne se fait pas par équipement mais par zone hydrographique.

### ***miseAJourVolumeNappe***

Action appelée depuis la méthode interne comportementJournalier. Dans sa version générique (donc si le modèle hydrographique est le modèle « simple »), elle ne fait rien.

### ***calculVolumeEntreePhaseRoutage***

Action appelée depuis la méthode interne comportementJournalier. Elle est appelée après le calcul des prélèvements et rejets réels. Elle calcule le volume d'eau disponible en entrée de la zone hydrographique pour la phase routage. Elle prend donc en compte l'eau qui arrive des zones hydrographiques amont + l'eau qui arrive de la phase sol (hru + hruRPG) + l'eau qui est rejetée - l'eau qui est prélevée.

### ***phaseRoutage***

Action appelée depuis la méthode interne comportementJournalier. Dans sa version générique (donc si le modèle hydrographique est le modèle « simple »), elle suppose que l'eau qui sort de la zone hydrographique est égale à celle qui entre (calculée dans le méthode calculVolumeEntreePhaseRoutage) sans perte durant le trajet.

***calculDebitSorti***

Action appelée depuis la méthode interne comportementJournalier. Elle converti le volume de sorti de l'agent en un débit.

***miseAJourHRUactive***

Action appelée depuis la méthode interne comportementJournalier. Dans sa version générique (donc si le modèle hydrographique est le modèle « simple »), elle ne fait rien.

***changementCouleurEnFonctionTemperatureMoyenne***

Action appelée depuis la méthode interne comportementJournalier. Elle met à jour un des affichages possibles de la zone hydrographique. Ici, la couleur est mise à jour en fonction de la température moyenne de l'agent.

***changementCouleurEnFonctionPrecipitationEtEtpMoyen***

Action appelée depuis la méthode interne comportementJournalier. Elle met à jour un des affichages possibles de la zone hydrographique. Ici, la couleur est mise à jour en fonction des précipitations qui tombent sur l'agent.

***changementCouleurEnFonctionDebit***

Action appelée depuis la méthode interne comportementJournalier. Elle met à jour un des affichages possibles de la zone hydrographique. Ici, la couleur est mise à jour en fonction du débit du coursDeau principal de l'agent.

***basic***

Action d'affichage SIG appelée dans la classe gaml Experiement. Cet affichage est statique.

***evolutionTemperatureAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore la zone hydrographique en fonction de la température.

***evolutionDebitAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore la zone hydrographique en fonction du débit.

***evolutionPrecipitationAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Colore la zone hydrographique en fonction des précipitations.

***toString***

Action d'affichage des variables de l'agent dans la console.

## zoneHydrographiqueSWAT.gaml

Cette classe est la classe fille de l'agent zoneHydrographique. Ce sont ces agents qui vont être créés si le modèle hydrographique choisi est le « modèle SWAT ».

### GLOBAL

#### *creationZoneHydrographiqueSWAT*

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise les agents de type zoneHydrographiqueSWAT à partir d'un fichier SIG d'entrée. Les agents zone hydrographiqueSWAT sont créés très tôt car ils représentent la référence spatiale, c'est donc eux qui dictent la forme de la zone d'étude dont tous les autres agents (géo-référencés) dépendent.

### AGENT

#### *initialisationZoneHydrographique [Overwrite]*

Action appelée depuis l'action globale initialisationZonesHydrographiques. Elle initialise certaines variables propres à SWAT.

#### *phaseSol [Overwrite]*

Action appelée depuis la méthode interne comportementJournalier. Elle récupère l'eau qui provient de chaque HRU sous forme de ruissellement de surface, d'évapotranspiration, d'écoulement latéral et d'écoulement souterrain. La somme de ces quantités sera ce qui est appelée le volume de la phase sol des agents hru classiques.

#### *phaseSolRPG [Overwrite]*

Action appelée depuis la méthode interne comportementJournalier. Elle récupère l'eau qui provient de chaque hruRPG sous forme de ruissellement de surface, d'évapotranspiration, d'écoulement latéral et d'écoulement souterrain. La somme de ces quantités sera ce qui est appelée le volume de la phase sol des agents hruRPG. Cette action est appelée séparément de la précédente pour des que les prélèvements puissent se faire.

#### *phaseRoutage [Overwrite]*

Action appelée depuis la méthode interne comportementJournalier. Elle est appelée après la phase sol et une fois que les prélèvements et les rejets aient été pris en compte. Elle calcule la quantité d'eau qui va sortir de la zoneHydrographique compte tenu des pertes par transmission ou encore par évaporation qui ont lieu tout au long du trajet de l'eau entre l'entrée et l'exutoire de la zone.

#### *calculDesPertesParTransmission*

Action appelée depuis la méthode interne phaseRoutage. Elle calcule les pertes par transmission.

#### *calculBankStorage*

Action appelée depuis la méthode interne phaseRoutage. Elle calcule le bank storage.

#### *calculPertesParEvaporation*

Action appelée depuis la méthode interne phaseRoutage. Elle calcule les pertes par évaporation.

#### *miseAJourHRUactive [Overwrite]*

Action appelée depuis la méthode interne comportementAnnuel. Elle met à jour la liste des HRUs et des hruRPG de l'agent si cela est nécessaire. Les surfaces de ces agents sont en effet amenées à évoluer dans le cas où un ou plusieurs îlots de la zoneHydrographique venait à disparaître.

#### *calculVolumeUtileNappesReel [Overwrite]*

Action appelée depuis la méthode interne calculVolumePrelevésReels. Elle calcule l'eau disponible dans les

nappes pour les prélèvements en nappes. Ce volume est calculé à partir des quantités stockées dans les aquifère peu profond de chacune de ses hru classiques. Remarque : Les volumes stockés dans les hruRPG ne seront donc jamais pris en compte étant donné que les prélèvements sont calculés avant la phase sol des agents hruRPG.

***miseAJourVolumeNappe [Overwrite]***

Action appelée depuis la méthode interne comportementJournalier. Une fois que les prélèvements humains et les rejets ont été calculés, la quantité présente dans les nappes de chaque HRU de l'agent sont mises à jour.

## zoneHydrographiqueJeuOC.gaml

---

Cette classe est la classe fille de l'agent zoneHydrographique. Ce sont ces agents qui vont être créés si le modèle agricole choisi est le « modèle JeuOC ».

### GLOBAL

#### *creationZoneHydrographiqueJeuOC*

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise les agents de type zoneHydrographiqueJeuOC à partir d'un fichier SIG d'entrée. Les agents zoneHydrographiqueJeuOC sont créés très tôt car ils représentent la référence spatiale, c'est donc eux qui dictent la forme de la zone d'étude dont tous les autres agents (géo-référencés) dépendent.

## Package « modeleNormatif »

- distributeurDeauBrute.gaml
- distributeurDeauPotable.gaml
- gestionnaire.gaml
- gestionnaireDeBarrage.gaml
- organismeUnique.gaml
- pointDeReference.gaml
- pointDeReferenceCalibration.gaml
- policeDeLeau.gaml
- prefet.gaml
- secteurAdministratif.gaml
- uniteDeDefinitionDuVP.gaml
- uniteDeGestion.gaml
- zoneAdministrative.gaml

## **distributeurDeauBrute.gaml**

---

Pas encore utilisé.

## **distributeurDeauPotable.gaml**

---

Pas encore utilisé.

## **gestionnaire.gaml**

---

Pas encore utilisé.

## **gestionnaireDeBarrage.gaml**

---

Pas encore utilisé (cf. la classe barrage). Remarque : il faudrait peut être mettre une partie du code de barrage ici.

## organismeUnique.gaml

---

L'organisme unique a pour rôle d'édicter chaque année le volume prélevable à allouer à chaque agriculteur.

### GLOBAL

#### *constructionOrganismeUnique*

Action appelée pendant la phase d'initialisation depuis le main. Elle crée l'unique agent de type organismeUnique.

### AGENT

#### *comportementJournalier*

Action appelée à chaque pas de temps depuis le main. Elle fait appel aux agents uniteDeDefinitionDuVP pour enregistrer les débits sous le DOE.

#### *comportementAnnuel*

Action appelée chaque année depuis le main. Elle fait appel aux agents uniteDeDefinitionDuVP pour allouer le volume prélevable aux agents agriculteur.

#### *toString*

Action d'affichage des variables de l'agent dans la console.

## pointDeReference.gaml

Cette classe représente les points où sont mesurés les débits le long des coursDeau. Il existe 3 types de point de référence :

- Station hydrographique : station géo-référencée où sont mesurés les débits
- Point Nodal (ou DOE) : point de contrôle situés sur une station hydrographique
- Point ROCCA : principalement pour les zones non réalimentées, les points ROCCA représentent les lieux (non référencés) surveillés par des agents de l'eau de manière informelle.

Remarque : Cette classe regroupe ces 3 types de points. Pour le moment seul les points nodaux sont pris en compte.

### GLOBAL

#### ***Attributs***

- pointDOEShape - Chemin vers le fichier SHP des points nodaux
- stationsHydrographiqueShape - Chemin vers le fichier SHP des stations
- nbJoursMoyenPourCalculDebit [jr]
- mapBesoinAgricole
- mapPalierDebit

#### ***creationPointsDeReference***

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise les agents de type pointDeReference à partir d'un fichier SIG d'entrée.

#### ***initialisationPointsDeReference***

Action appelée pendant la phase d'initialisation depuis le main. Elle est faite séparément de la création car il faut que les agents zoneAdministrative aient été créés.

### AGENT

#### ***Attributs***

- idPointDeReference
- idZHassociee
- taillePointRef
- couleurTypePoint
- couleurPalierDebit
- isNodal
- isStationHydrographique
- doe [m/s]
- qa [m/s]
- qi [m/s]
- qar [m/s]
- dcr [m/s]
- qmj3 [m/s]
- debitJournalier
- seuilDeGestion
- mapDebit
- mapQMj3
- mapDebitMaxParNiveauDeRestriction
- tendance
- zoneHydrographiqueAssociee [zoneHydrographique]
- isRealimente

– mapDebitReel

***comportementJournalier***

Action appelée à chaque pas de temps depuis le main. Elle fait appel à des actions qui vont mettre à jour son débit mesuré et son affichage.

***initialisationPointDeReference***

Action appelée depuis la méthode globale creationPointsDeReference. Elle initialise certaines variables tel que les seuils DOE, DCR, QA, QAR et QI.

***miseAJourDebit***

Action appelée depuis la méthode interne comportementJournalier. Elle stocke le débit de sa zoneHydrographique associée et en calcule une moyenne sur les 3 derniers jours.

***surveillancePointDeReference***

Action appelée depuis l'agent zoneAdministrative. Elle retourne vrai dès lors que le débit moyen devient inférieur à un la valeur seuil DOE.

***colorationDifferentiationTypePoint***

Action appelée dans la méthode interne initialisationPointDeReference. Elle initialise un des affichages possibles de l'agent. Ici elle colore le point en fonction de sont type.

***colorationEnFonctionDebitMoyen***

Action appelée depuis la méthode interne comportementJournalier. Elle met à jour l'affichage de l'agent en fonction du débit moyen par rapport aux différents seuils.

***basic***

Action d'affichage SIG appelée dans la classe gaml Experiement. Cet affichage est statique.

***palierDebitAspect***

Action d'affichage SIG appelée dans la classe gaml Experiement. Elle colore le point en fonction de sont débit moyen. Si il est inférieur au débit DOE il sera coloré en jaune, plus il sera sous un seuil faible plus sa couleur sera rouge.

***toString***

Action d'affichage des variables de l'agent dans la console.

## pointDeReferenceCalibration.gaml

---

Cette classe charge en entrée des données réelles en vue de la calibration du modèle hydrographique.

### GLOBAL

#### *constructionPointDeReferenceCalibration*

Action appelée pendant la phase d'initialisation depuis le main. Elle récupère d'un fichier csv d'entrée les débits réels des points de références. Ces débits sont stockés dans les agents pointDeReference concernés. Ces débits réels seront utiles pour la phase de calibration.

## policeDeLeau.gaml

---

Son rôle est de contrôler et potentiellement verbaliser les agriculteurs qui ne respectent pas une restriction pendant la période d'étiage. De plus, chaque année elle va contrôler quels agriculteurs ont utilisé plus d'eau que leur VP alloué et les verbalise si besoin.

### GLOBAL

#### ***constructionPoliceDeLeau***

Action appelée pendant la phase d'initialisation depuis le main. Elle crée l'unique agent de type policeDeLeau.

### AGENT

#### ***comportementJournalier***

Action appelée à chaque pas de temps depuis le main. Elle fait appel à une méthode interne qui contrôle le respect des restrictions.

#### ***comportementAnnuel***

Action appelée à la fin de chaque année depuis le main. Elle fait appel à différentes méthodes internes qui vont contrôler le respect du volume prélevable, et qui vont faire un récapitulatif de fin d'année.

#### ***controleRespectRestriction***

Action appelée depuis la méthode interne comportementJournalier. Elle contrôle un jour par semaine un certain nombre d'agriculteurs choisi aléatoirement en période d'étiage. Afin d'éviter de contrôler toujours les mêmes agriculteurs (les agriculteurs étant toujours associés aux mêmes jours de restrictions), la police de l'eau va contrôler cycliquement du lundi au dimanche. Ce contrôle se fait une fois par semaine maximum. Chaque agriculteur enregistre le nombre de fois qu'ils ont été vu en train d'irriguer contre restriction.

#### ***controleRespectVolumePrelevable***

Action appelée depuis la méthode interne comportementAnnuel. Elle contrôle et verbalise si besoin les agriculteurs qui ont utilisé plus que leur volume prélevable. Chaque agent agriculteur va garder en mémoire une indication sur la quantité d'eau qu'ils ont utilisé en plus de ce qu'il leur été alloué.

#### ***recapitulatifAnnuel***

Action appelée depuis l'action interne comportementAnnuel. Elle fait un récapitulatif du nombre d'agriculteurs qui ont été enregistré en train d'irriguer contre restriction et du nombre d'agriculteurs qui ont utilisé plus que leur volume prélevable alloué.

#### ***toString***

Action d'affichage des variables de l'agent dans la console.

## **prefet.gaml**

---

Le principal acteur du modèle normatif est le préfet. Il n'y en a qu'un pour toute la zone d'étude. C'est lui qui décide de faire un arrêté ou de faire des lâchers de barrage pour l'étiage. Les lâchers pour l'électricité sont simulés à travers l'agent barrage (ou gestionnaireDeBarrage).

### GLOBAL

#### ***constructionPrefet***

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise l'unique agent de type préfet.

### AGENT

#### ***comportementJournalier***

Action appelée à chaque pas de temps depuis le main. Elle fait appel à des méthodes internes qui surveille et applique l'algorithme de gestion d'étiage si besoin.

#### ***initialisationPrefet***

Action appelée depuis la méthode globale `constructionPrefet`. Elle initialise l'agent.

#### ***surveillance***

Action appelée depuis la méthode interne `comportementJournalier`. Elle demande à tous les agents zoneAdministrative de surveiller les débits.

#### ***gestionEtiage***

Action appelée depuis la méthode interne `comportementJournalier`. Dans le cas où la campagne d'étiage serait ouverte, elle demande à tous es agents zoneAdministrative d'appliquer l'algorithme de gestion d'étiage.

#### ***toString***

Action d'affichage des variables de l'agent dans la console.

## secteurAdministratif.gaml

---

Une zoneAdministrative peut être divisée en 1 ou plusieurs secteurs administratifs. Un secteur est une agrégation de communes. On applique alternativement les jours de restriction sur les différents secteurs la zoneAdministrative.

### GLOBAL

#### ***constructionSecteursAdministratifs***

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise les agents de type secteurAdministratif à partir d'un fichier SIG d'entrée. Remarque : pour le moment ces données ne sont pas disponibles.

### AGENT

#### ***initialisationSecteurAdministratif***

Action appelée depuis l'action globale constructionSecteursAdministratifs. Elle initialise les agents commune appartenant à l'agent courant.

#### ***isSecteurEnBesoinAgricoleFort***

Action appelée depuis l'agent zoneAdministrative. Elle renvoi vrai si ses communes l'une de ses communes sont en besoin agricole fort.

#### ***interdictionPompage***

Action appelée depuis l'agent zoneAdministrative. Elle parcourt la liste de ses agents commune et met à jour leurs jours de restriction (si il y en a).

#### ***toString***

Action d'affichage des variables de l'agent dans la console.

## uniteDeDefinitionDuVP.gaml

---

Une des problématiques de MAELIA est d'avoir différents scenarios, avec entre autre comme variable la définition du VP. Par exemple, chaque année on peut calculer le VP comme le volume annuel de l'année la pire sur 5 années (année quinquennale sèche). Ce calcul est fonction du DOE. Une unité va calculer un VP d'une uniteDeGestion. Car une UG est associée à un point DOE.

### GLOBAL

#### ***constructionUniteDeDefinitionDuVP***

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise un agents du type uniteDeDefinitionDuVP pour agent uniteDeGestion existant.

### AGENT

#### ***initialisationUniteDeDefinitionDuVP***

Action appelée depuis l'action globale constructionSecteursAdministratifs. Elle affecte les agents agriculteur ayant au moins une parcelle irrigable dans leur exploitation à l'agent courant.

#### ***enregistrementDebitEtiage***

Action appelée chaque pas de temps depuis l'agent organismeUnique. Elle enregistre la somme des débits au dessus du DOE pendant la période d'étiage en vue de calculer le volume prélevable de l'uniteDeGestion associée à la fin de l'année.

#### ***allocationVPauxAgriculteurs***

Action appelée chaque année depuis l'agent organismeUnique. Elle alloue à chaque agent agriculteur associé son volume prélevable pour l'année à venir. Ce dernier est calculé en égale au niveau de l'UG à la valeur minimale des 5 dernières années.

#### ***toString***

Action d'affichage des variables de l'agent dans la console.

## uniteDeGestion.gaml

---

Une unité de gestion est une entité géo-référencée proposant un découpage de la zone d'étude en grande unité de gestion. Une UG est associée à un point DOE.

### GLOBAL

#### *constructionUniteDeGestion*

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise les agents uniteDeGestion à partir d'un fichier SIG d'entrée.

### AGENT

#### *initialisationUniteDeGestion*

Action appelée depuis l'action globale constructionSecteursAdministratifs. Elle affecte un point nodal à l'agent courant. Ce point est ensuite utilisé pour connaître la valeur définie du seuil DOE pour le calcul du volume prélevable.

#### *basic*

Action d'affichage SIG appelée dans la classe gaml Experiment. Cet affichage est statique.

#### *toString*

Action d'affichage des variables de l'agent dans la console.

## zoneAdministrative.gaml

Une zone administrative est composée d'un ensemble d'agents secteurAdministratif. C'est sur chacune de ces zones que vont se décider et s'appliquer les arrêtés de restriction. Une zone est associée à un et un seul agent pointDeReference (nodal ou station hydrographique). Remarque : Pour le moment, seules les ZA associées aux points nodaux sont prises en compte, car on ne connaît pas les seuils pour les stations hydrographiques.

### GLOBAL

#### *Attributs*

- zonePtNodauxChemin - Chemin vers le fichier SHP
- listeZAParOrdreAvalVersAmont [zoneAdministrative]
- mapNiveauDeRestriction

#### *constructionZonesAdministratives*

Action appelée pendant la phase d'initialisation depuis le main. Elle crée et initialise les agents zoneAdministrative à partir d'un fichier SIG d'entrée.

#### *determinationZoneAdministrativeAval*

Action appelée depuis l'action globale constructionZonesAdministratives. Elle crée une liste de zones administratives triées d'aval en amont. Elle sera ensuite utile pour la gestion des périodes d'étiage.

### AGENT

#### *Attributs*

- idZoneAdministrative
- idPointDeReferenceAssocie
- idZHassociee
- secteursAdministratifsAssocies [secteursAdministratifs]
- couleurTypeZone
- couleurPointRefPalierDebit
- couleurEnFonctionNbJoursRestriction
- pointDeReferenceAssocie [pointDeReference]
- zhAssociee [zoneHydrographique]
- zoneAdministrativeAval [zoneAdministrative]
- niveauDeRestriction
- barrageAssocie [barrage]
- historiqueDesArretes
- isEnCampagneEtiage
- isRealimente
- isBesoinAgricoleFort
- nbJoursMinPourNouvelleRestriction [jr]
- nbJoursMinPourBaisserRestriction [jr]
- nbJoursMemeNiveauRestriction [jr]

#### *comportementJournalier*

Action appelée à chaque pas de temps depuis le main. Elle fait appel à de actions qui vont mettre à jour le besoin agricole ainsi que l'affichage de l'agent.

#### *initialisationZoneAdministrative*

Action appelée depuis l'action globale `constructionZonesAdministratives`. Elle initialise le lien entre la zone administrative et les agents `pointDeReference`, `barrage` ou encore `secteurAdministratif`.

#### ***surveillanceZoneAdministrative***

Action appelée depuis une méthode interne `surveillance` de l'agent `prefet`. Elle met à jour un attribut indiquant si la campagne d'étiage doit commencer. Celle-ci commence si l'agent `pointDeReference` associé constate que son débit moyen est passé sous la valeur seuil du DOE.

#### ***miseAJourBesoinAgricole***

Action appelée depuis la méthode interne `comportementJournalier`. Elle parcourt ses agents `secteursAdministratifs`, dans le cas où au moins un des ses secteur est en besoin fort, alors toute la zone administrative le sera aussi.

#### ***interdictionPompage***

Action appelée depuis la méthode interne `gestionEtiage`. Cette action va mettre à jour les jours de restriction (ou d'interdiction de pompage) de chaque `agriculteur` de chaque `secteur administratifs` de la zone, dans le cas bien sûr où la campagne d'étiage l'exige. Si le niveau de restriction est inférieur à 4 (soit pas 7 jours), chaque secteur verra ses agriculteurs restreint des jours de la semaine différents.

#### ***isLacheBarrageEnCours***

Action appelée depuis la méthode interne `gestionEtiage`. Cette action demande à l'agent `barrage` associé si un lâcher est en cours le jour courant.

#### ***lacheBarrage***

Action appelée depuis la méthode interne `gestionEtiage`. Cette action demande à l'agent `barrage` associé de commencer, ou d'augmenter le débit de lâchure.

#### ***gestionEtiage***

Action appelée depuis la méthode interne `gestionEtiage` de l'agent `prefet`. Elle est appelée dans le cas où le `prefet` décide de commencer la campagne d'étiage. Elle décide quelle action faire selon le seuil sous lequel se trouve le débit moyen de son `pointDeReference`. Ainsi, lorsque le débit est encore assez proche du DOE, un lâcher de `barrage` est demandé pour se ramener au seuil (si bien un lâcher est possible le jour courant). Si le débit passe sous un autre seuil les restrictions de niveau 1 ou 2 vont être mises en place. Si le débit passe sous le seuil critique le niveau de restriction va alors être maximale et un lâcher de `barrage` va être demandé. Il y a quelques autre règles à respecter entre les zones administratives elles-mêmes. Ainsi, une zone en amont ne pourra pas avoir plus d'un niveau d'écart avec sa zone en aval. Il n'y a pas de contrainte pour les zones en aval de la zone restreinte. Pour plus d'information sur la gestion d'étiage se reporter au document « [/Tasks/T4. Processus/T4.3.25 - Ediction arrete de restriction/T4.3.25 - Ediction arrete de restriction.pdf](#) ».

#### ***colorationEnFonctionPointRef***

Action appelée depuis la méthode interne `comportementJournalier`. Elle met à jour l'affichage de l'agent en fonction du débit moyen de son `pointDeReference` associé.

#### ***colorationEnFonctionNbJoursRestriction***

Action appelée depuis la méthode interne `comportementJournalier`. Elle met à jour l'affichage de l'agent en fonction du nombre de jour de restriction.

#### ***basic***

Action d'affichage SIG appelée dans la classe `gaml Experiement`. Cet affichage est statique.

#### ***pointRefPalierDebitAspect***

Action d'affichage SIG appelée dans la classe gaml Experiment. Elle colore la zone selon le débit moyen de son pointDeReference.

***nbJoursRestrictionAspect***

Action d'affichage SIG appelée dans la classe gaml Experiment. Elle colore la zone selon son nombre de jour de restriction.

***toString***

Action d'affichage des variables de l'agent dans la console.

## Package « testsUnitaires »

- launcher.gaml
- testModeleAgricole.gaml

## **launcher.gaml**

---

C'est depuis cette classe que sont lancés les différents tests unitaires des tous les modules confondus.

## **testModeleAgricole.gaml**

---

C'est dans cette classe que sont définis les tests unitaires liés aux actions globales et internes des agents du modèle agricole.