Overview of the “*SiriusQuality-BioMa-WheatLAI-Component”* folder

1. Content of the “SiriusQuality-BioMa-WheatLAI-Component” folder

   The folder “SiriusQuality-BioMa-WheatLAI-Component” contains:

   - The source code of the *SQ-WheatLAI BioMa* Component (*SiriusQuality-WheatLAI* folder), see *SQ2_WheatLAI_component* document
   - The BioMa dlls which are mandatory to use the component (*BioMa-DLL* folder)
   - A console application which provides an example for the use of the component (*SiriusQuality-WheatLAIConsole* folder)
   - Unit tests for the component (*UnitTestWheatLAI* folder)
   - A visual studio solution which allows to run the console application and the unit tests (*SiriusQuality-WheatLAI.sln*)
   - A detailed documentation about both the calculation scheme and the equations of the component (*Documentation* folder)

2. How to use the component:

   The *WheatLAI* component can be added to a *BioMa* solution with the *CLIC* tool or be plugged to your model via a wrapper. Here the composition with *CLIC* will not be presented (see instead *BioMa* solution documentation). We will first make a quick overview of the component, then we will present the wrapper and the leaf layer classes of *SiriusQuality*. Finally, we will explain how they are used via a console application.

   a. Overview of the component

      The component (*SiriusQuality-PhenologyComponent)* contains six folders:

      - The *domainClass* folder. Here can be found:
        - The getter and setter of the general states (*WheatLAIState.cs*) and those for the states

specific to the leaf layers (*WheatLeafState.cs*). The state of the leaf layers are *Lists.*
- The metadata on the states (*WheatLAIStateVarInfo.cs and WheatLeafStateVarInfo.cs*)

- The *strategies* folder where the simple strategies and the composite one (*WheatLAI.cs*) can be found. The composite strategy allows calling sequentially the simple strategies via its *Estimate* function or the *UpdateArea* simple strategy via the composite strategy function called by the same name
- The *API* folder containing the classes for the Application Programming Interface
- An *XML* folder where the xml files used to generate strategies and domain classes with the BioMa tools can be found
- The *obj/Debug* folder containing the dll of the component after having built it and the *BioMA* dlls which are mandatory for the project
- The *bin* folder for binaries

b. Phenology wrapper

The wrapper (*SiriusQuality-PhenologyConsole/WheatLAIWrapper.cs*) makes possible:

- The initial loading of the parameters
- The connection of the *SiriusQuality* leaf layer objects to the leaf layers of the component.
- The day by day valorization of the inputs
- The daily call of the component (the *Estimate* function of the composite class and the *UpdateAreas* function are called separately in the core of *SiriusQuality*)
- And the daily export of the outputs

For these purposes a *WheatLAIState* object along with two *WheatLeafState* (*WheatLeafstate_* and *WheatLeafstate1_*) ones are instantiated. Most of the time, *WheatLeafstate_* contains the information of the current day, but when used with *WheatLeafstate1* it contains the information of the day before while *WheatLeafstate1_* is used for the current day. These objects allow valorizing inputs and export outputs (via getter). In addition, an object of the composite class is instantiated (*wheatLAI_*). It helps to valorize the parameters and call the *Estimate* or the *UpdateAreas* function of the composite class.

The valorization of the parameters is done in the constructor via the *loadParametersWheat* function. The *Estimate* and *UpdateArea* function can be called everywhere in the code. their arguments are the values of the input for the current day. Four steps are achieved in these functions:

- The current day inputs are valorized
- The information on the leaf layer object in *SiriusQuality* are loaded in the leaf layer lists of the component via the *FillIntputLayersWheat* function
- The *Estimate* function or the *UpdateAreas* function of the composite class is called
- The information on the leaf layer object of *SiriusQuality* are updated from the new values of the component leaf layer lists which are now outputs (*FillOutputLayersWheat*)

A last function has to be mentioned: *CreateLeafLayerLAIComponentWheat* is used to Add an element to the component layer list each time a leaf layer is created in *SiriusQuality*.

c. Leaf layer classes of *SiriusQuality*

Two classes are used to model leaf layers in *SiriusQuality.* They were copied in the folder *SiriusQuality-WheatLAIConsole*:

- *LeafState.cs* contains only a public object of type *enum* which is use to enumerate and store the possible states of a leaf layer (Growing, Mature, Senescing, Dead)

- ➢ *LeafLayer.cs* allows creating as many leaf layer objects as contained in the canopy via a constructor. Here can also be found a copy constructor. The class proprieties (*LayerPhyllochron*, *GAI*, *sheathAI*, *LaminaAI*, *State*, *DeltaDM*, *DeltaAI*…) characterize each individual leaf layer. The *LeafLayer* objects can be seen as containers for the information on single leaf layers.

d. Console application

The *SiriusQuality-WheatLAIConsole/Program.cs* class can be divided in five parts:

- ➢ Inputs that cannot be calculated are grouped in tables. The table counts correspond to the number of days in the simulation
- ➢ An object *WheatLAIWrapper* is instantiated to be able to call the *wrapper's Estimate* and *UpdateAreas* functions and export outputs
- ➢ When entering the loop over the days of the plant cycle several steps are achieved daily:
    - Other inputs are calculated
    - Leaf layer are created both at the console application level (*LeafLayer.cs* class) and in the *WheatLAI* component (*CreateLeafLayerLAIComponentWheat*)
    - The *WheatLAI* component is called via the *Estimate* function of the wrapper
    - The leaf layer's area indexes are updated via the *UpdateAreas* function of the wrapper (which calls the *UpdateAreas* function of the component)
    - The total Green Area Index of the canopy is updated (increase for growing leaves, decrease for senescing ones)
    - The biomass of individual leaves is updated accordingly and that of the whole canopy, as well.
- ➢ Outputs of the component are called from the wrapper and printed
- ➢

3. List of the provided libraries

   Six libraries are mandatory to be able to run the component. Five of them are part of the *BioMa* framework and are loaded both in the component and in the console application projects, while *CRA.AgroManagement2014.dll*, *CRA.AgroManagement2014.Impacts.dll* and *CRA.AgroManagement2014.dll* are used for the management of agronomic events (irrigation, fertilization…), *CRA.Core.Preconditions.dll* is used for the test of parameter, input and output values and *CRA.ModelLayer.dll* is dedicated to the generation of domain classes and strategies.

   When building the *SiriusQuality-WheatLAI* project a library called *SiriusQuality-WheatLAI.dll* is created. It is loaded in the console application project (and the corresponding include is added on top of the wrapper class).

   These six libraries have to be loaded in any project aiming at working with the phenology component.

4. List of inputs and outputs of the component:

Inputs		
Name in the code	units	Definition
newLeafHasAppeared	-	1: a new leaf has appeared the day before, 0: if not
roundedFinalLeafNumber	leaf	Rounded final leaf number (at maturity) on mainstem
finalLeafNumber	leaf	Decimal final leaf number on mainstem
leafNumber	leaf	Leaf number on mainstem the current day
FPAW	-	Fraction of available water
isPotentialLAI	-	1: to apply drought stress, 0: if not
cumulTTShoot	°Cd	Thermal time accumulated by the canopy
deltaTTShoot	°Cd	Daily growth thermal time increase of the day
deltaTTSenescence	°Cd	Daily senescence thermal time of the day
VPDairCanopy	hPa	Air-Canopy Vapour Pressure Deficit
phytonum	phytomer	Number of phytomers on the plant
index	-	Index of the last created phytomer
previousIndex	-	Index of the previous created phytomer
tilleringProfile	shoot/m²	List which stores the density of new tiller (main-stem +tillers) created at each time a new tiller appears
leafTillerNumberArray	shoot	List which stores the number of tillers (main-stem +tillers) for each leaf layer
GAI	m²(leaf)/m²(ground)	List which stores the Green Area Index (Lamina Area Index + Exposed Sheath Area Index) of the day before for each leaf layer
Phyllochron	°Cd/leaf	List which stores the phyllochron of each leaf layer
laminaSpecificN	g(N)/m²(leaf)	List which stores the photoshyntetycally active nitrogen content of each leaf layer

Outputs		
Name in the code	units	Definition
State	-	List of the state of each leaf layer as an *integer* (0: Growing, 1:Mature, 2:Senescing, 3:Dead). The WheatLAIWrapper's function getLeafStateList() convert the list of integers in a list of *LeafState*.
isPrematurelyDying	-	List of *integers*. An *integer* (1: if the leaf is prematurelyDying, 0: if not) is assigned at each leaf layer of the list.
potentialIncDeltaArea	m²(leaf)/m²(ground)	Total daily potential expansion rate. It corresponds to the sum of the potential expansion rate of each layer.
incDeltaAreaLimitSF	m²(leaf)/m²(ground)	Total daily expansion rate under drought stress.
incDeltaArea	m²(leaf)/m²(ground)	Total daily expansion rate under drought and nitrogen stress.
WaterLimitedPotDeltaAIList	m²(leaf)/m²(ground)	List which stores the expansion or senescence rate under drought stress for each leaf
DSF	-	Drought stress factor for senescence acceleration
DEF	-	Drought stress factor for expansion

TTsen	°Cd	List which stores the thermal time duration of senescence for each leaf layer
TTmat	°Cd	List which stores the thermal time duration of the maturation phase for each leaf layer
TTgroLamina	°Cd	List which stores the thermal time duration of expansion for each leaf layer
IsSmallPhytomer	-	List which stores an *integer* (1: for juvenile phytomers, 0: for others) for each leaf layer
MaxAI	°Cd	List which stores the maximum Green Area Index attained the current day for each leaf layer
DeltaAI	m^2(leaf)/m^2(ground)	List which stores the daily expansion rate under drought and nitrogen stress for each leaf layer
laminaAI	m^2(leaf)/m^2(ground)	List which stores the lamina area index under drought and nitrogen stress for each leaf layer the current day
sheathAI	m^2(leaf)/m^2(ground)	List which stores the exposed sheath area index under drought and nitrogen stress for each leaf layer the current day

A list of the parameters dedicated to the calculations of the component is given in Table A2 of */Documentation/SQ3_WheatLAI_component.pdf*.