

The ACME calculation chain for

Agile

Crop

Model

Ensemble

Michel GINER [(michel.giner@cirad.fr](mailto:michel.giner@cirad.fr) )

François AFFHOLDER [(francois.affholder@cirad.fr](mailto:francois.affholder@cirad.fr) )

Persyst - UPR AIDA  
TA B-115 / 02

Avenue Agropolis - 34398 Montpellier Cedex 5  
France

SUMMARY

Table of contents

[1. Preamble 3](#_Toc106875339)

[2. Presentation of the ACME calculation chain 3](#_Toc106875340)

[3. The models implemented 4](#_Toc106875341)

[4. The components of DATAMILL 5](#_Toc106875342)

[5. Benchmark 10](#_Toc106875343)

[6. Exploitation of the results 11](#_Toc106875344)

[7. Datamill screens 13](#_Toc106875345)

[8. The ModelDictionnaryArise Variables table 14](#_Toc106875346)

[9. How MasterInput is used 16](#_Toc106875347)

[10. Concept note to decompartmentalise Datamill 18](#_Toc106875348)

## Preamble

The ACME calculation chain for Agile Crop Model Ensemble is a set of software (Datamill), two databases (ModelDictionnaryArise and MasterInput) and a notion of tree structure on which its execution is based. It was designed in the framework of the Climate-KIC ARISE research project and is strongly influenced by it, especially at the level of input variables. The purpose of this document is to explain how to implement the simulations using the Datamill software, but not to explain how to collect the necessary data or how to exploit the results.

The software was written in Visual Basic and uses Access as its database manager.

Learning Access is highly recommended by training from <https://www.york.ac.uk/it-services/coursefiles/booklets/Essential%20Access_Book-1.pdf> or any other resource available on the internet or from a competent person.

## Presentation of the ACME calculation chain

The aim of the Climate-KIC ARISE research project is to develop a new service for stakeholders in the agricultural supply chain. It should enable the transformation of agricultural systems for better resilience to climate change and sustainability.



Figure 1: DATAMILL welcome screen

The approach taken is based on the implementation of crop models to evaluate the production capacity of a selection of plants in the area under study. In order to finely define which areas are at risk or conversely the best areas for planting crops, the software platform relies on climatic and soil data from the ERA5Land, Chirps and ISRIC sites.

The aim of the project is to map a country, a geographical area according to risk criteria, which implies evaluating the said area according to the finest grid corresponding to the lowest common denominator of themes such as pedologies or climatology. The system can be used on our PCs as part of feasibility studies or for monitoring experiments.

The plant growth models are rather developed for a WINDOWS environment, focused on experimentation and not very spatialized, even if the geographical information is important at least for a good consideration of the local climate. They are not designed to use geo-referenced data directly and the introduction of data is done on a trial-by-trial basis in the released version of the models. On the other hand, one can use the model's calculation core by providing it with sets of text files as would its interface. This leads us to the construction of an application capable of generating sets of files useful to each model and combining soil, climate, technical itinerary and cultivar inputs. In this way, we will be able to process the set of simulations as if each of our models were spatialized.

A difficulty of the subject lies in the fact that the data accessible by the project sources (grouped in the MasterInput database) do not present an absolute bijection with the set of model parameters. For some, the relationship will be immediate, for others, calculation formulas using several project parameters will provide us with an acceptable proxy, and last but not least, the assignment of default values to a whole group of parameters.

The team's effort was therefore focused on various points,

* Implementation of spatialized data with tools (models) designed for fixed situations,
* Fine-tuned learning of model operation to develop consistent parameter sets without the data verification tools included in the model interfaces or observed data to calibrate certain parameters,
* Agronomic expertise of the regions of the world targeted to set plausible default values,
* Implementation of a file tree that guarantees both the correct individual execution of each simulation and the collection of relevant results without ambiguity.
* Finally, be able to transform the resulting data into a spatial data source to facilitate communication between the different project actors.

## The models implemented

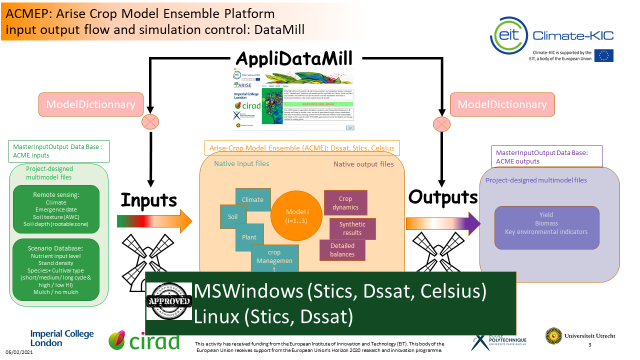
The models selected are STICS V9, DSSAT V4.7 and Celsius Nov17.

For technical reasons, we have chosen to keep the native structure of the models to introduce the varietal parameters. Nevertheless, in MasterInput, the ListCultivars and ListcultOption tables must be filled in to establish the link between DataMill and the models.

**Important constraint**:

* Stics v9.00 : The Modulostics.exe file is the model itself. It will read the files prepared by DataMill. Think to position the cultivar files in the directory dedicated to stics and to fill in the ListCultOption table (field ficplt) so that the good cultivar file is copied under the name of ficplt1.txt
* DSSAT V4.7: The application must be installed on the workstation and the varietal parameters modified in the genotype directory under C:\DSSAT47 and the ListCultOption table filled in (Prcrop and cg fields)
* Celsius: version of 03 November 2017. Cultivar definitions are done directly in the Celsius database

The idea of the platform was thus fixed on the diagram below.



## The components of DATAMILL

Pragmatically, the DATAMILL system for ARISE consists of databases and software code to generate a set of directories containing the necessary and sufficient components to run simulations at a specific geographical point and by a model.

DATAMILL relies on two databases of different use and durability.

The first, called ModelDictionnaryArise, mainly contains a table that manages the bijection and proxies between the project parameter set (Masterinput) and the model parameter sets. It is in ModelDictionnaryArise that the expertise of our agronomists is put down in black and white to directly link a parameter identified by the project to a model parameter, to fill in a calculation formula to evaluate the value of a model parameter as a function of one or more Masterinput parameters and to fill in the remaining parameters in a coherent manner in relation to the study area.

ModelDictionnaryArise also serves as a detailed specification of parameter-level implementations in the Datamill application. By correctly filling in the fields provided for this purpose, it is possible to define how to implement but also in return what has been implemented. Although rather rustic, this approach ensured the link between the computer scientist and the agronomist.

The following slide shows the number of variables to be processed per model and per theme. This describes the importance of the work done.

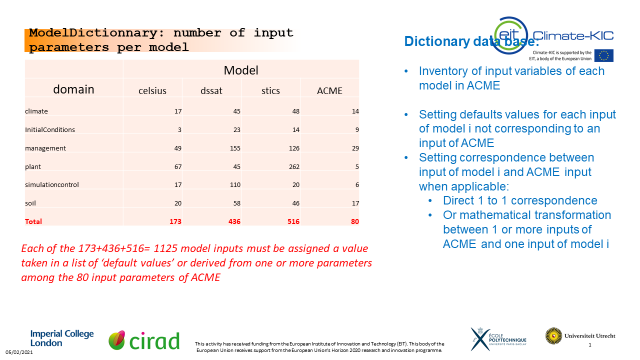


Figure 2: Number of model parameters

This is reflected in the structuring of the second database (MasterInput).

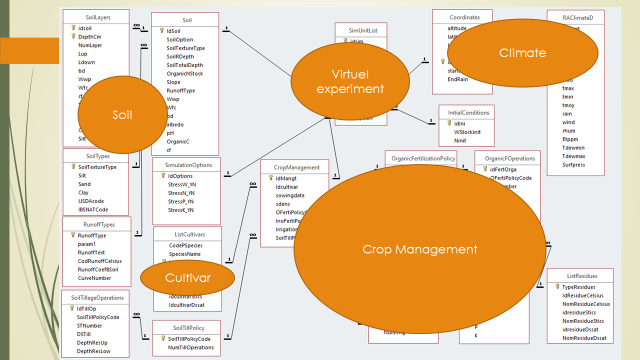


Figure 3: Thematic grouping of MasterInput tables

The latter, named Masterinput, contains all the data collected by and for the project.

It is from the knowledge entered in ModelDictionnaryArise that the DATAMILL software declines the writing of the various files based on the information characterising the bijections, the calculated proxies or the default values to be positioned in the files. The values resulting from the notion of bijection or proxy come from extraction from the Masterinput database and the default values from ModeldictionnaryArise.

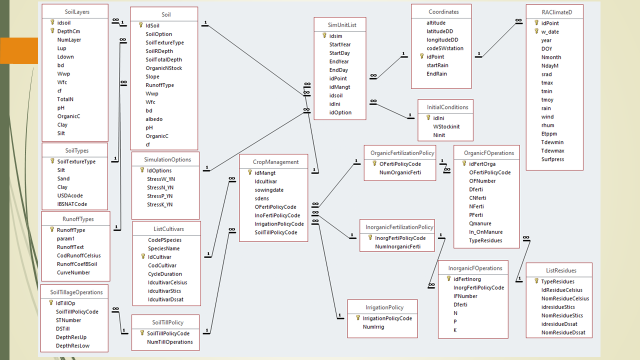


Figure 4: Relational diagram of MasterInput

The execution of the DATAMILL software will create in each of the directories dedicated to the models either an ACCESS database for Celsius or a set of specific directories containing the files adapted to the simulation required for DSSAT or STICS. The highest level directory containing the two databases and one directory per model. Several tree structures with different names but the same architecture can be used to work on several virtual experiments.

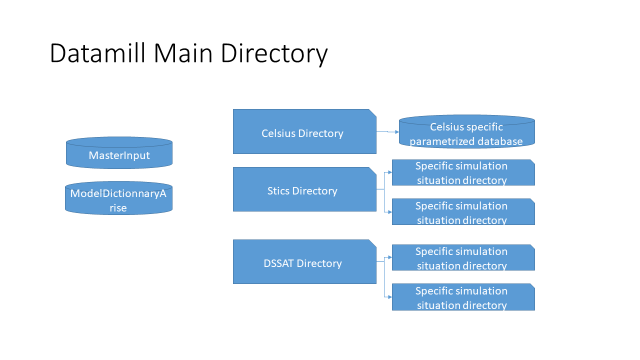


Figure 5: File tree diagram

This set of files is associated with a DOS command file capable of launching the simulation, the whole being headed by another command file summarising all the simulations to be launched for a given model and taking advantage of all the processor cores.

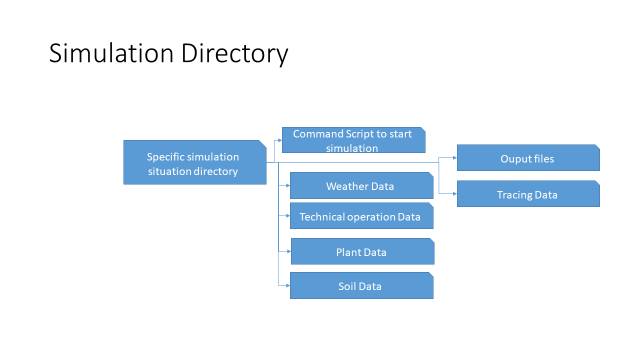


Figure 6: Contents of a simulation situation directory

The following box shows the contents of the scripts generated by Datamill that launch the execution of the simulations (general script; all simulations in parallel and script of an individual simulation) for DSSAT and STICS

For DSSAT, the control script (individual simulation) contains

*C:DSSAT47DSCSM047.EXE B DSSBatch.v47*

exit

The call is made by Dssat.bat from ACME\_workspaceDssat

*Start /d C:\ACME\_WorkspaceDssat\BEOU.2014.BEOU1401.1 BEOU.2014.BEOU1401.1.bat*

*Start /d C:\ACME\_WorkspaceDssat\BEOU.2015.BEOU1501.1 BEOU.2015.BEOU1501.1.bat*

*Start /d C:\ACME\_WorkspaceETBA.2013.ETBA1301.1 ETBA.2013.ETBA1301.1.bat*

*Start /d C:\ACME\_WorkspaceDssat\ETBA.2014.ETBA1401.1 ETBA.2014.ETBA1401.1.bat*

*Start /d C:\ACME\_WorkspaceDssat\GHKP.2008.GHKP0801.1 GHKP.2008.GHKP0801.1.bat*

*......*

For Stics, the control script (individual simulation) contains

*copy ..\var.mod /Y*

*copy ..\maiplt.txt ficplt1.txt /Y*

*Del mod\*. sti*

*.. \stics\_modulo*

*exit*

The call is made by Stics.bat of ACME\_espacedetravail\Stics

*Start /d C:\ACME\_WorkspaceBEOU.2014.BEOU1401.1 BEOU.2014.BEOU1401.1.bat*

*Start /d C:\ACME\_WorkspaceBEOU.2015.BEOU1501.1 BEOU.2015.BEOU1501.1.bat*

*Start /d C:\ACME\_Workspace\Stics\ETBA.2013.ETBA1301.1 ETBA.2013.ETBA1301.1.bat*

*Start /d C:\ACME\_Workspace\Stics\ETBA.2014.ETBA1401.1 ETBA.2014.ETBA1401.1.bat*

*Start /d C:\ACME\_Workspace\_SticsGHKP.2008.GHKP0801.1 GHKP.2008.GHKP0801.1.bat*

*......*

Contents of the Datamill tree (the directories useful to the system and an example of a simulation directory)

The volume in the C drive is called System

The serialnumber of the volume is 66C3-5D23

Directory of C:\ACME\_Workspace

13/06/2022 08:56 <DIR> .

13/06/2022 08:56 <DIR> .

30/05/2022 10:03 <DIR> celsius

30/05/2022 10:25 <DIR> Dssat

07/03/2022 15:45 53 870 592 MasterInput.accdb

07/03/2022 11:31 2 572 288 ModelsDictionaryArise.accdb

13/10/2021 16:08 16 617 MZCER047.CUL

13/10/2021 16:08 1 870 MZCER047.ECO

30/05/2022 10:28 <DIR> Stics

10 file(s) 223,846,228 bytes

Directory of C:\ACME\_Workspace

30/05/2022 10:03 <DIR> .

30/05/2022 10:03 <DIR> .

22/11/2021 10:01 66 Celsius.bat

30/05/2022 10:03 41 005 056 CelsiusV3nov17\_dataArise.accdb

8 file(s) 59,624,870 bytes

Directory of C:\ACME\_Workspace

30/05/2022 10:25 <DIR> .

30/05/2022 10:25 <DIR> .

30/05/2022 10:26 <DIR> BEOU.2014.BEOU1401.1

30/05/2022 10:26 <DIR> BEOU.2015.BEOU1501.1

30/05/2022 10:25 860 Dssat.bat

07/03/2022 08:48 <DIR> ETBA.2013.ETBA1301.1

07/03/2022 08:48 <DIR> ETBA.2014.ETBA1401.1

07/03/2022 08:48 <DIR> GHKP.2008.GHKP0801.1

07/03/2022 08:48 <DIR> GHKP.2009.GHKP0901.1

07/03/2022 08:48 <DIR> MANT.2009.MANT0901.1

07/03/2022 08:48 <DIR> MANT.2010.MANT1001.1

07/03/2022 08:48 <DIR> RWBU.2013.RWBU1401.1

07/03/2022 08:48 <DIR> RWBU.2014.RWBU1501.1

3 file(s) 868 bytes

Directory of C:\ACME\_WorkspaceDssat\BEOU.2014.BEOU1401.1

30/05/2022 10:26 <DIR> .

30/05/2022 10:26 <DIR> .

30/05/2022 10:25 44 BEOU.2014.BEOU1401.1.bat

30/05/2022 10:25 22 831 BEOU1401.WTH

30/05/2022 10:25 22 833 BEOU1501.WTH

30/05/2022 10:26 4 129 DSSAT47.INH

30/05/2022 10:26 4 521 DSSAT47.INP

30/05/2022 10:25 260 DSSBatch.v47

30/05/2022 10:26 36 940 ET.OUT

30/05/2022 10:26 931 Evaluate.OUT

30/05/2022 10:26 6 022 INFO.OUT

30/05/2022 10:25 4 844 ITSA1301.SGX

30/05/2022 10:26 685 MON.LST

30/05/2022 10:26 8 111 Mulch.OUT

30/05/2022 10:26 60 487 N2O.OUT

30/05/2022 10:26 1 473 OUTPUT.LST

30/05/2022 10:26 9 277 OVERVIEW.OUT

30/05/2022 10:26 47 650 PlantGro.OUT

30/05/2022 10:26 17 409 PlantN.OUT

30/05/2022 10:26 82 RunList.OUT

30/05/2022 10:26 39 575 SoilNi.OUT

30/05/2022 10:26 1 975 SoilNiBal.OUT

30/05/2022 10:26 1 870 SoilNoBal.OUT

30/05/2022 10:26 10 738 SoilTemp.OUT

30/05/2022 10:26 21 983 SoilWat.OUT

30/05/2022 10:26 1 285 SoilWatBal.OUT

30/05/2022 10:26 1 504 SolNBalSum.OUT

30/05/2022 10:26 71 362 SOMLITN.OUT

30/05/2022 10:26 2 020 Summary.OUT

30/05/2022 10:26 1 201 WARNING.OUT

30/05/2022 10:26 21 721 Weather.OUT

30/05/2022 10:25 590 XX.SOL

30 file(s) 424,353 bytes

Directory of C:\ACME\_WorkspaceStics

30/05/2022 10:28 <DIR> .

30/05/2022 10:28 <DIR> .

30/05/2022 10:28 <DIR> BEOU.2014.BEOU1401.1

30/05/2022 10:28 <DIR> BEOU.2015.BEOU1501.1

01/09/2021 18:42 5 300 cowplt.txt

30/05/2022 10:28 <DIR> ETBA.2013.ETBA1301.1

30/05/2022 10:28 <DIR> ETBA.2014.ETBA1401.1

07/03/2022 09:02 <DIR> GHKP.2008.GHKP0801.1

07/03/2022 09:02 <DIR> GHKP.2009.GHKP0901.1

08/03/2021 07:35 10 657 maiplt.txt

07/03/2022 09:02 <DIR> MANT.2009.MANT0901.1

07/03/2022 09:02 <DIR> MANT.2010.MANT1001.1

01/09/2021 18:42 4 317 milplt.txt

01/09/2021 18:42 4 320 ricplt.txt

07/03/2022 09:02 <DIR> RWBU.2013.RWBU1401.1

07/03/2022 09:02 <DIR> RWBU.2014.RWBU1501.1

01/09/2021 18:42 4 689 sorplt.txt

30/05/2022 10:28 860 Stics.bat

23/10/2018 10:31 9 754 612 stics\_modulo.exe

01/09/2021 15:14 406 var.mod

14 file(s) 9 803 795 bytes

Directory of C:\ACME\_Workspace\_SticsBEOU.2014.BEOU1401.1

30/05/2022 10:28 <DIR> .

30/05/2022 10:28 <DIR> .

30/05/2022 10:28 90 BEOU.2014.BEOU1401.1.bat

30/05/2022 10:28 56 210 climate.txt

30/05/2022 10:28 232 ficini.txt

08/03/2021 07:35 10 657 ficplt1.txt

30/05/2022 10:28 1 789 fictec1.txt

30/05/2022 10:28 16 387 modhistory.sti

30/05/2022 10:28 9 290 mod\_bmaize.sti

30/05/2022 10:28 13 749 mod\_profile.sti

30/05/2022 10:28 501 mod\_report.sti

30/05/2022 10:28 100 213 mod\_smaize.sti

30/05/2022 10:28 321 new\_work.usm

30/05/2022 10:28 541 param.sol

30/05/2022 10:28 29 prof.mod

30/05/2022 10:28 125 rap.mod

30/05/2022 10:28 68 635 recup.tmp

30/05/2022 10:28 70 snow\_variables.txt

30/05/2022 10:28 433 station.txt

30/05/2022 10:28 573 stics\_errors.log

30/05/2022 10:28 7 044 tempopar.sti

30/05/2022 10:28 1 310 tempoparv6.sti

01/09/2021 15:14 406 var.mod

21 file(s) 288,605 bytes

Here we have a system which ensures the independence of the simulations so that the consolidation of the results can be carried out from independent files which can be differentiated by the simulation identifier which is also the name of the directory. Another advantage is that we also have files for tracing the conditions of the simulations, which are very useful in the system development phases.

The interest of working with databases is that the climate, technical itinerary and soil objects can be treated in separate tables, or even with some refinements, such as for the soil and the different horizons. Thus, each situation can receive a unique identifier that can be mixed between the different objects (soil, climate, ITK, etc.) to compose the simulation tuple to be built.

The database tool, through its SQL query language, allows a Cartesian product of the objects in the database to quickly compose a very rich experimental design.

The result of the Cartesian product is stored in a table (Simunitlist) which will be the starting point of the work of constitution of the sets of files but also a tool for checking the relevance of our virtual experiment.

The composition by the software of the useful files is based on queries selecting the useful fields and filtering all the cases of figures useless to the simulation to be generated. The presentation in tables of each of the objects allows an input as in a spreadsheet and facilitates the verification of the data set. If one had to do this with the native interfaces of the models, the time and the risk of error would have been very big brakes.

It is possible to identify other data sources, as long as we know how to manage the connection to these sources and their interrogation. And consequently the modification of the DataMill source code.

## Benchmark

As soon as we were satisfied with the results on a small dataset (from the Agmip initiative), we started to write an experimental design of 89280 simulations.

The following tables show the disk size and time parameters of a set of 89280 simulations. Stics and DSSAT are of equivalent design but do not use the same number of files nor the same compilers, which explains the differences in performance. On the other hand, Celsius uses MS ACCESS to run and gives the impression of running as in a container and therefore performs very well.

|  |  |  |  |
| --- | --- | --- | --- |
| **Simulation time (redone on 18-20/03/2021)** | | | |
| **Model** | **Step** | **1 simulation (sec)** | **Fullvirtualexperiment-89310 simulations (hour:minutes:seconds)** |
| **Stics** | Buildmodelinputs | 0.935 | 07:44:00 |
|  | Runsimuations | 0.653 | 05:24:00 |
|  | Harmonizeoutputs | 0.0014 | 00:00:43 |
| Total |  | 1.5894 | 13:08:43 |
| **Celsius** | Buildmodelinputs | 0.003 | 00:01:42 |
|  | Runsimuations | 0.165 | 01:22:00 |
|  | Harmonizeoutputs | 0.0009 | 00:00:27 |
| Total |  | 0.1689 | 01:24:09 |
| **Dssat** | Buildmodelinputs | 0.393 | 03:15:00 |
|  | Runsimuations | 0.487 | 04:02:00 |
|  | Harmonizeoutputs | 0.002 | 00:01:10 |
| Total |  | 0.882 | 07:18:10 |
| All models-all steps | | 2.6403 | 21:51:02 |

|  |  |  |
| --- | --- | --- |
| Disk space required | | |
| Model | 1 simulation (KB) | Total virtual experiment (GB) |
| Stics | 614 | 52.3 |
| Celsius | 9 | 0.75 |
| Dssat | 435 | 37 |
| Total | 1058 | 90.05 |

## Exploitation of the results

Once the simulations have been run, the data of interest are collected in the form of a table in the Masterinput database called SummaryOutput. It is from this database that the agronomists were able to assess the consistency of the results. This stage was used to check the relevance of the parameters and file generations. As the entire calculation chain is based on the unique identifier of the simulation, it is possible, thanks to the SQL query language, to compare the parameters of the simulation and the results of the latter in the same data matrix. This validation work, although tedious, could be carried out in an acceptable time. This made it possible to initiate feedback on the software development in order to achieve the primary objective of a first, fairly rough study before introducing more finesse in the parameterizations and complexity in the software in order to come closer to the expectations of our partners.

The SummaryOutput table consists of the following columns, one row per model and per simulation. If the simulation did not produce any results, the "text" field is filled in and the other variables are left blank. Additional information is available in the Variables table.

|  |  |  |
| --- | --- | --- |
| Table | Field | Description |
| SummaryOutPut | Ant | Anthésis Date |
| SummaryOutPut | Biom\_ma | Maturity Date |
| SummaryOutPut | CroN\_ma | Nitrogen in plant |
| SummaryOutPut | CumE | cumulative evaporation over cropping season |
| SummaryOutPut | Emergence | Emergence Date |
| SummaryOutPut | GNumber | Number of grains |
| SummaryOutPut | IdSim | Identifier of simulation |
| SummaryOutPut | Mat | Dry biomass |
| SummaryOutPut | MaxLAI | Lai Maximum |
| SummaryOutPut | Model | Model used |
| SummaryOutPut | Nleac | Nitrogen leached |
| SummaryOutPut | Planting | Planting Date |
| SummaryOutPut | SoilN | Nitrogen in soil |
| SummaryOutPut | Text | Error Message if importing data failed |
| SummaryOutPut | Transp | cumulative transpiration over cropping season |
| SummaryOutPut | Yield | Yield at maturity |

We can now pride ourselves on having set up the complete processing chain, from spatialized climate and soil data to the representation of crop monitoring indicators



Figure 7: ACME data processing diagram

Among the advantages of the system are that it eliminates the need for tedious data entry and can currently be used for three models. Focusing on the core business and expertise of agronomists to establish technical itineraries adapted to the new climatic conditions. Limiting the exercise of the virtual experiment only to the processing capacity of the computer equipment available.

## Datamill screens

89280 Simulations

Executed on Intel I7-8650U 2.11Ghz Turbo Mode

4 cores 8 thread   
16GB RamSSD   
Disk



Figure 8: Datamill welcome screen

This screen consists of 4 tabs and 3 options. The tabs allow access to the front page and the simulation capabilities (commands per model).

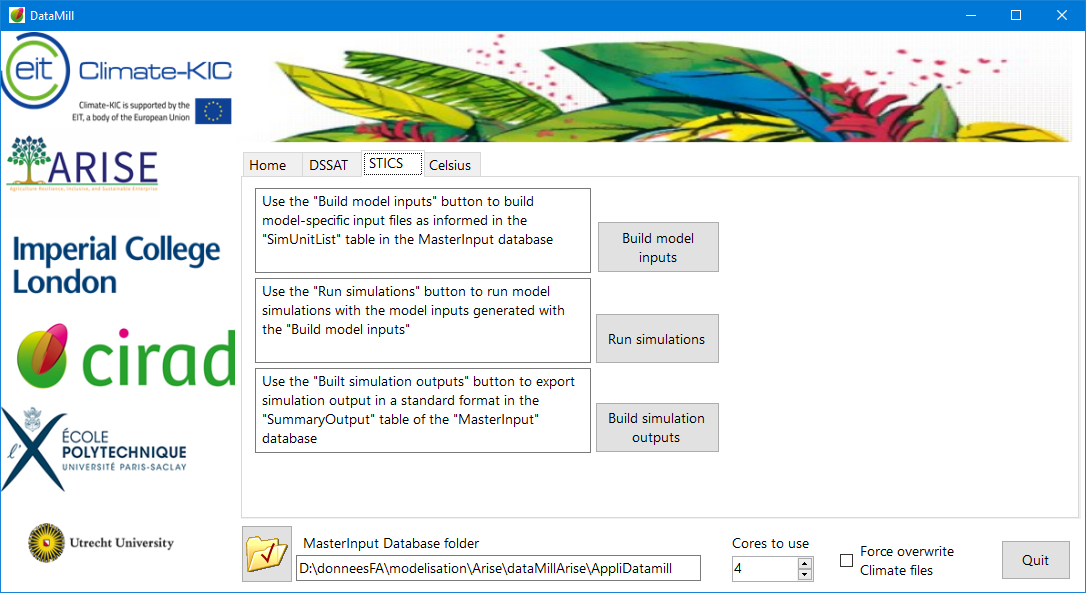


Figure 9: Control screen for the treatment chain

A model tab offers 3 options

* **Build models input** to generate the sets of simulation situations as defined in the SimUnitList table of MasterInput
* **Run simulation** to run the simulations.
* **Build simulation outputs** to populate the SummaryOutput table of Masterinput which collects the results of the simulations

At the bottom of the screen, you have 3 options

* **MasterInput Database folder** which allows you to select the directory containing the tree structure to be executed. This option opens the possibility of having several simulation spaces in order to follow several virtual experiments.
* **Cores to use**: depending on your processor you have a certain number of cores (on an I5: 4 in general, on an I7: 8). This allows you to launch the file generations according to as many process queues as defined, which results in a performance gain. During the execution of the simulations this parameter is not functional but all the capacities of the PC are exploited
* **Force overwrite Climate files** forces to rewrite the climate files because to save time they are only rewritten if the simulation directory does not exist. This means that if your simulation has been generated once, the climate files are not recreated during subsequent generations, even if you have modified the climate content in MasterInput. This option allows you to overcome this behaviour.

## The Variables table of ModelDictionnaryArise

This table contains all the information that was used to write the DataMill program. Most of the variables have been used to ensure a good implementation of the parameter sets and constitute a kind of laboratory notebook for the generation of the simulation spaces.

Thanks to this information, the DataMill code was written taking into account three possible sources for the parameter

* Default value if the **DefaultValueYN** field is true
* Calculated value if **FunctionACME** contains a formula
* Value reproduced from a MasterInput field

The main variables to be monitored are

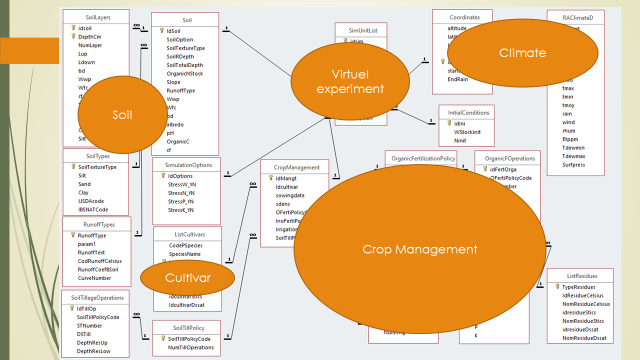
* **DefaultValueYN** which determines if the value for this model parameter is a default value filled in by an expert and assigned in one of the two fields Default\_Value\_Datamill or defaultValueOtherSource knowing that the Datamill program will use defaultValueOtherSource when it is filled in otherwise Default\_Value\_Datamill will be taken into account
* **FunctionACME** which is implemented in the program code to calculate a proxy obtained from several parameters collected in MasterInput. If you wish to modify this formula, the DataMill code must be modified accordingly
* **Follow-upModify** allows you to note when the last modification was made
* **ACMEinputField** which determines the field to be copied into the model parameter in the case of a replicated value
* **Field** that determines the name of the model parameter

|  |  |
| --- | --- |
| Field | Description |
| model | Stics or Dssat or Celsius |
| Table | Table or file |
| Field | Name of the field |
| Description |  |
| unit |  |
| Type |  |
| domain | localization, management, climate, soil, simulationcontrol, plant, initial conditions... |
| TableUsedInACMEYN | is the set of parameters belonging to this table used in ACME ? yes/no (apply to parameters that are optional in a model) |
| TableListYN | is the set of parameters belonging to this table a predefined list that should not be modified in ACME simulations ? Yes / no |
| defaultvalueYN | is the variable to be set at default value Yes/No (or respectively True/False) |
| status | identify, how, standard ( input...) |
| ACMEinputEquivalentYN | is there an exact equivalent in ACMEinput table (with values stored in MasterInputs.accdb Yes/No |
| ACMEinputField | equivalent field in ACMEinput if ACMEinputEquivalentYN is True. Null otherwise |
| FunctionACME | Math Function to use for getting field value from ACME inputs in MasterInputs.accdb when ACMEinputEquivalentYN is False and defaultvalueYN is False |
| Default\_Value\_Datamill | default value taken from Data\_Arise database (datacomponent of datamill) |
| FileNativeF | Name of file in model s native format |
| defaultValueOtherSource | default value taken from other source |
| SourceDV | Source of default value |
| Minnval | minimal value for the parameter |
| Maxval | maximal value for the parameter |
| comment |  |
| Follow-upModif |  |

Figure 10: List of fields in the Variables table

The Celsius model derogates from the rule of reading default values in ModelDictionnaryArise and requires a series of tables suffixed with the acronym DV (for default value in the CelsiusV3nov17\_dataArise.accdb database) to be filled in with default value data. Attention should be paid to the tables ListPAnnexesDV, ParamIniDV, SoilDV, Soil\_layersDV, Tech\_CommunDV, Tech\_perCropDV for Coordinates, InitialConditions, Soil, SoilLayers and Management (? ) respectively.

## How MasterInput is used



The MasterInput database is built around a SimUnitlist table that links different identifiers that themselves designate the different occurrences of the objects that make up the crop evolution system. In extenso: Climate, Soil, Technical Route and Crop. Each of these objects is composed of one or more tables describing in as much detail as possible all the data useful for setting up the simulations, such as the values reproduced or the proxies calculated by formula.

|  |  |  |  |
| --- | --- | --- | --- |
| model | domain | Table | Description |
| master | climate | RAClimateD | Climate Data |
| master | InitialConditions | InitialConditions | Initial conditions |
| master | localization | Coordinates | Site contact details |
| master | management | CropManagement | Technical routes |
| master | management | InorganicFertilizationPolicy | Mineral fertilisation |
| master | management | InorganicFOperations | Mineral fertilisation operations |
| master | management | ListResidues | Residue |
| master | management | OrganicFertilizationPolicy | Organic fertilisation |
| master | management | OrganicFOperations | Organic fertilisation operations |
| master | plant | ListCultivars | Cultivars |
| master | plant | ListCultOption | Model option for cultivars |
| master | SimulationControl | SimUnitList | List of simulations |
| master | soil | RunOffTypes | Runoff |
| master | soil | Soil | Soil parameter |
| master | soil | SoilTypes | Soil classification |

Figure 11: Linking tables to themes

The definition of the contents of these tables can be found in the Variables table of ModelDictionnaryArise. You must fill the different tables with the data corresponding to the experiments you wish to evaluate with the models.

The SimUnitList table can be populated by manually filling in

* The identifier of the simulation that will be used to name the directory that will host the simulation (ids)
* The start and end dates of the simulation
* The climate identifier (idpoint)
* The technical route identifier (idmangt)
* The soil identifier (idsoil)
* The initial condition identifier (idini)
* The stress option identifier (idoption)

Or by composing queries such as, for example, selecting a number of tables that allow the fields of the Simunitlist table to be filled in by displaying the fields idoption, idpoint, idmangt, idsoil, startyear, endyear, startday, endday, idini and the calculation of the field idsim by the formula ids: [coordinates].[idpoint] & ". & [year] & ". & [idmangt] & ". & [idsoil] & "." & [idoptions]

The fact of not putting a link between the tables allows SQL to constitute a data matrix resulting from the crossing of all the values retained in the display. In this case, if we have 3 cropmanagement, 5 soils, 2 simulation options and 200 climatic situations (here we are filtering the sites with data for the year 2000), we will obtain 5\*2\*3\*200=6000 simulations in the virtual experiment. The use of the Distinct clause makes it possible to eliminate duplicate records, otherwise the results are multiplied by the number of lines contained in RAClimateD (see the SQL code).

Finally, this query adds its result to the existing Simunitlist table. This makes it possible to create something finely adapted according to the relevant compositions and filtering that you know how to do.

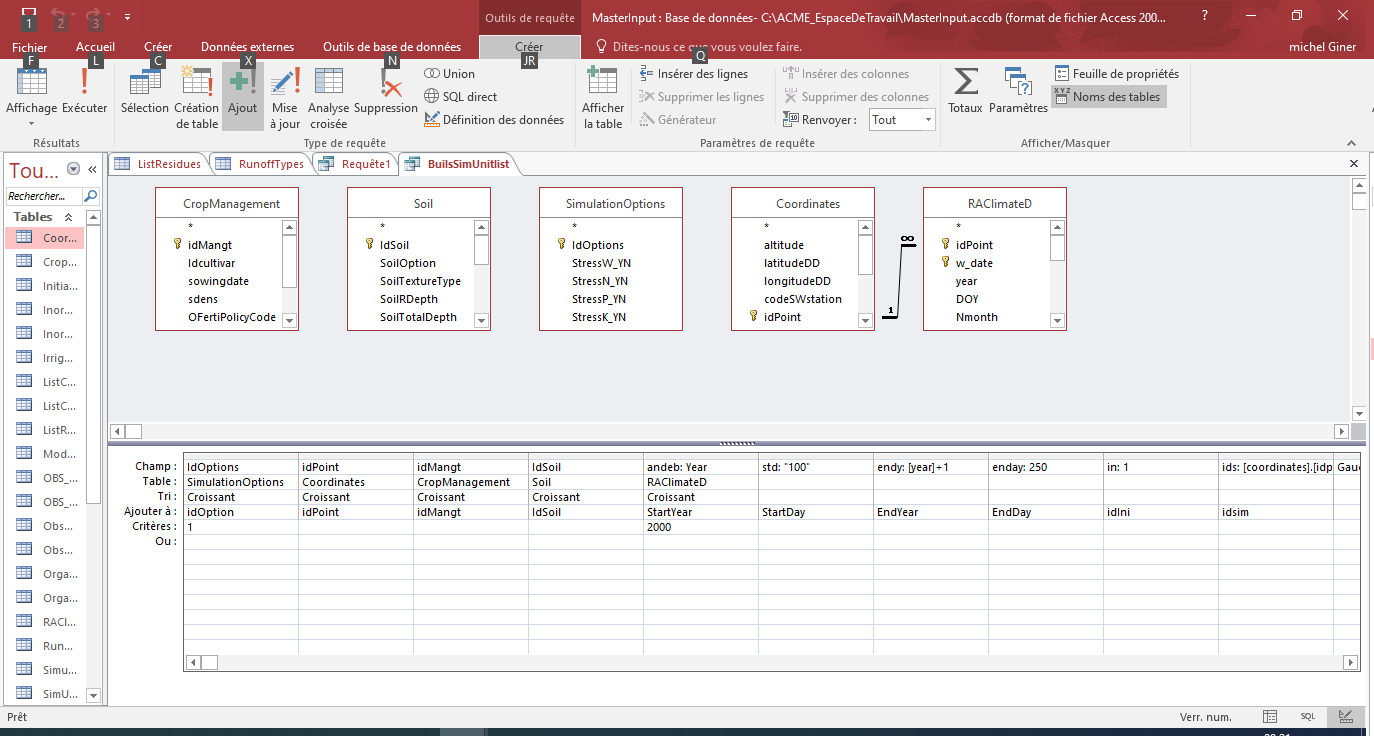


Figure 12: Presentation of the query in the ACCESS interface (incomplete image, see SQL)

Presentation of the SQL text of this query

INSERT INTO SimUnitList ( idOption, idPoint, idMangt, IdSoil, StartYear, StartDay, EndYear, EndDay, idIni, idsim )

SELECT Distinct SimulationOptions.IdOptions, Coordinates.idPoint, CropManagement.idMangt, Soil.IdSoil, RAClimateD.Year AS andeb, "100" AS std, [year]+1 AS endy, 250 AS enday, 1 AS [in], [coordinates].[idpoint] & ". & [year] & "." & [idmangt] & ". & [idsoil] & "." & [idoptions] AS ids

FROM CropManagement, Soil, SimulationOptions, Coordinates INNER JOIN RAClimateD ON Coordinates.idPoint = RAClimateD.idPoint

WHERE (((SimulationOptions.IdOptions)=1) AND ((RAClimateD.Year)=2000))

ORDER BY SimulationOptions.IdOptions, Coordinates.idPoint, CropManagement.idMangt, Soil.IdSoil, RAClimateD.Year;

Translated into good French, this means :

Inserts (INSERT INTO ) in the Simuntlist table the fields taken in the display order ( idOption, idPoint, idMangt, IdSoil, StartYear, StartDay, EndYear, EndDay, idIni, idsim ) which I select by taking the following display order, including calculations. Taking care to keep only the distinct records (DISTINCT clause) from the tables (FROM clause) filtered according to the WHERE clause and sorted according to the ORDER BY clause

Note that the name of a field can be changed by the AS clause, which is in square brackets and refers to tables or fields in the database

The INNER JOIN clause allows two tables to be linked by equality of the quoted fields. If no link clause is quoted between tables then the Cartesian product of the fields selected for display is used.

## Concept note to decompartmentalise Datamill

When reading this document, we realise that if our data or its structure does not correspond to the definition of MasterInput, a lot of work needs to be done to conform. Furthermore, the datasets whose value is taken by default and the one that belongs to the project dataset are defined and limited to this definition.

It is interesting to note, when reading the code, that for each file to be created, only a limited number of queries are implemented, almost all of which have the same global form and the same way of applying filters to the data. In fact, for the software, a query is nothing more than a text that can be parameterised by concatenating strings of characters.

This approach has already been implemented in the SISTER data management system for cotton technology and has been working for a number of years, even when the laboratory's data structure changes without rewriting the supporting software.

The first task is to create a table (Req\_Selection) in ModelDictionnaryArise to identify, name and store the different queries.

The second step is to go through the code to isolate the queries in their general form (without the filter part) in order to fill the Req\_Selection table.

Once this is done, the text of a query can be selected at the place of execution from a query of the following form in order to make the code independent of the data access queries:

Access to the Stics model queries for the climate module

Dim T2 As String = "Select \* from Req\_selection where Modele='stics' and [Module]='climat' Order By N"

Dim DSql As New DataTable

Dim Cmd2 As New OleDb. OleDbDataAdapter(T2, DI\_CS)

the query set is stored in the DSQL object

Cmd2.Fill(DSql)

the string fetchAllquery calls line 0 of DSql by selecting the 'SQL field (text of interest) and concatenating the filters and sorting depending on the 'module

fetchAllQuery = DSql(0)("SQL") & " where idPoint='" + Site + "' And (Year=" & Year & " or Year=" & Year + 1 & ") Order by w\_date;"

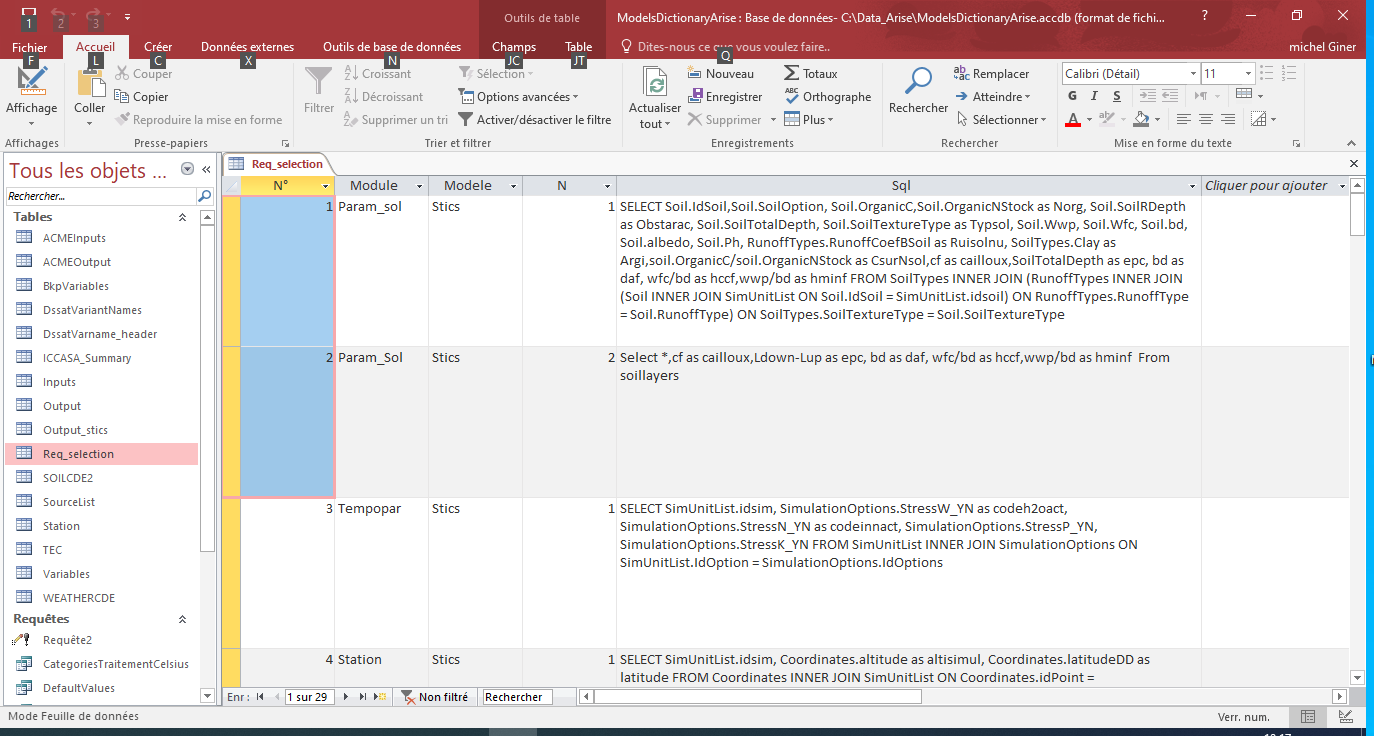


Figure 13: Contents of the req\_selection table

As the query text is an interpreted element at runtime, we can dispense with a predefined and unique structure of our Datamill system at the agronomic database level (MasterInput)

The software text is thus fixed on an access to agronomic data through the Req\_Selection query collection, which is to be considered as fixed. For example, if two queries (global level and layer contextualisation) are required for stics soils as Datamill expects, this requirement must be met as well as the requirement to present the result of this query as Datamill expects. For the record, SQL allows you to rename a field and perform some simple calculations in the query text. However, if your field names do not match the software's expectations, this is resolved by using the AS clause. You are then free to access the database defined for another purpose if it contains what you need for the simulations.

In the wake of this, the first version of this new Datamill has endeavoured to allow a field to be switch from the set of default values to the set of informed parameters and vice versa.

To do this, any field in the Variables table of ModelDictionnaryArise has a boolean field qualifying whether it is taken as a default value or not. In the negative case, it consider that one of the Req\_Selection queries will fill it. The reading of the data has been standardised and systematised in datamill according to a procedure defined as follows:

If (the field belongs to the default fields) then

Move up the default value corresponding to the field

Otherwise

Bring up the value from the selected query result in Req\_Selection

FinSi

This openness makes it possible to generalise the approach taken during the KIC Arise climate project by not freezing the sets of variables taken by default or filled in only at the initial determination.

This requires a little more work, but not necessarily more than defining the contents of the "variables" table. The advantage is that it allows the user greater flexibility in data management.

This approach transforms DATAMILL into a kind of data access engine in order of writing all the files of a virtual experiment. The parameterisation of the engine is done through the Variables and Req\_Selection tables of ModelDictionnaryArise. Apart from knowledge of models, the only prerequisites for using this version of Datamill are familiarity with the Microsoft ACCESS interface and the handling of SQL queries.

For the time being, Stics and DSSAT are upgrade in this way, using the project's data source.