The Crop Model Micro Core System

Reference Manual

Outline:

Introduction:

The definition, function and impact of crop models

Crop model is the computer procedure to simulate the growth of crops under certain environments. It has been widely used in the analysis in the area of agronomy, environment, economy and so on.

The current modular system of the crop models.

It has been over fifty years since the concept of the crop model came out. Hundreds of models have been developed since then. Before the

The year 1990 was a turning point for the development of the crop models. After the discussion of the crop models, modular system become the protagonist of the development and formed the systems that are still popular in nowadays research, such as the DSSAT, APSIM, STICS and so on.

Current improvement on the structure:

Spatial analysis.

CROPSPAL

Multi-model combination and crop module selection.

AgMIP

Disadvantages:

Fortran language of the old modules.

Fortran used to be the start of the scientific programming, while it is also the programing language which is gradually abandoned by the researchers today. There is no doubt that Fortran cannot support many excellent programming features, especially the Object-Oriented Programing diagram and results in the difficulties in the modern design and core re -use. As for the APSIM, it has been completely merged to the .Net Framework. WOFOST also has a version written in Python (Python Crop Simulation Environment, PCSE) for more convenience when integrated with other tools. STICS has a new version called RECORD. STICS used the Java to build the software.

However, refactoring the model using another programming language is of no doubt exhaustive comparing with just wrapping the source codes to a library that could be used directly.

Reuse of the crop model modules, data and paremters

It is obvious that the components of the modules in one framework are not compatible for another one. It is some what because the granularity of the framework.

The granularity

The granularity is also an important factor of the framework.

The burden for the individual developers and the collection of their effort. Community

A platform for more convenience on the developing and publishing of their effort.

We should realize that the improvements were not only achieved by the big teams, but also the diligent individual researchers or the small teams. However, their work were not

DSSAT created a Fortran Simulation Environment

The obstacles between the systems and the granularity.

The gap between the crop models and mathematical tools.

Flexibility (protocol to support model selection)

The researchers using the crop models are gradually evolving to a community. In addition, the background of them are also becoming more and more diverse. We thus wish a platform that could provide convenience for the researchers no matter who have much programming experience or not, no matter whose background is agronomy or not, in easily developing, acquiring and utilizing the crop models.

Structure of the platform.

Programming language

Discussion

If has been over fifty years since the concept of the crop model came out[]. Over these years, hundreds of crop models have been developed, while based on similar theories.

In the environmental area, the improvement usually focus on the soil water movement or the SPAC system.

The community of crop model research and researchers

The researches

It has been over fifty years since the concept of a crop model came out. Over the fifty years, hundreds of models have been developed. Though the theories related with either the growth of crop or the water, heat, solute transportation in the soil have been fully investigated, the structure of the current models still could not satisfied the community of developers.

There is still a critical point for the community to feel exhausted when prompting the current models is that the pillar of the developers are the doctoral or even master students who are of less profession and experience in software engineering and even programming. It is never surprise for them to write the codes that are difficult to maintain and extend because the procedure is only created for some specific functions nut never in a suitable framework.

In fact, the nowadays popular crop models are still developing in dispersed manners.

Therefore, we dedicated on an open platform for all the crop model components which are compliant with some least requirements whichever programming language they are of and whichever theory they used.

In this article, a microkernel system base on fully the “Plug – in Architecture” is proposed and is wished to provide an easy way to prompting the code reuse efficiency and narrow the gap between the developers of different programming languages.

There should be a platform to break the boundary between the nowadays popular crop models and could

Only the interface for the components is unified to give more freedom for tehe developers.

Programming language

The communication among the components is one the most difficult obstacle when developing the system. Passing data with the XML file is never considered because of the computing efficiency. Passing data using the socket convention seems a generic solution when crossing operation systems. However, under the Windows operating system, encapsulating the data into a component object model could easily realize the sharing of the memory to speed up the computing.

Because we are aiming to construct an open platform for all components that are compliant with the require

One critical difference of the system comparing to other model platforms is that the parameters and states are totally dynamic. Unlike the DSSAT packing the states in a Module, the system would never know which parameter or state would be used before the running information loaded. Pre – defining some parameters is thus impossible. Therefore, it is the component’s responsibility to gather all the data it needs from data pools. This character forced us to choose the COM as the final solution even though the .Net Framework would be more popular and more readable. The reasons would be presented later.

The core of the system is constructed on the C++ and the GUI is constructed by the PyQT5. The choice of the programming language is a trade-off. Thought the C# and .Net framework might be a popular choice. However, it is difficult to pass data to the Fortran libraries in a fast speed.

Non invasion architecture

The micro core system

The main body of the software is a micro kernel system for the crop models. The so-called “microkernel system” or the “Plug – in Architecture” is a term in the software architecture. It means a minimum software for

The final structure is settled after exhaustive attempts and tests which considered different programming languages, current popular frameworks, the background of fresh modelers and the future potential usage and extend of the system.

Data Pools

There are two data pool for the system, one is for the parameters and another is for the states. This is similar to the technology used by the DSSAT. However, the advantage in of the pool is the state and parameters are label uniquely by their name. To enhance the search speed, the BKRD hash function whose seed is 131 is used to cast the names into integers. The parameters would not be maintained by the library itself for the more convenient operation when changing the parameters for further analysis. Another obstacle arises as this design, that is, how all the information and value of parameters and states collected before running the simulation. The XML file is used, this technology has been widely used either in the APSIM or the STICS. It is required that a library must have a function named “Description” to generate a XML file that contains the information of the input and output variables including the name, type and size of the variable. A function named “Convert” to convert the original parameter file into XML file is also required. By this way, the parameter file could not be confined in the XML file as the STICS does and we wish this operation would save the energy for the freshman to get familiar with the system.

After these designs, there are only three requirements that modelers must obey:

1. Keep the consistency of the name of states used.
2. Be compliant with the formula of the XML file.
3. Keep the consistency of the interface with the formula designed.

There also some additional work that modelers need do, to form the description function of the component to let the system could know the component. Create an interface that could

Update needed parameters and states before the simulation in a time step begins and write the simulation result to the state pool after the simulation in a time step finished.

Library

But unfortunately, the directly converted COM by MATLAB does not accord with our requirements. Therefore, we created a wrapper for such kind of codes. We thus created a wrapper.

Programming Language

Demonstrations

Discussion