Crop Model MKS使用说明

绪论

尽管作物模型自面世以来已经历了将近50年的发展。但在这50年间，计算机与软件工程方面的技术已经飞速发展，以面向对象的软件设计方式为代表。在新时期，作物模型的开发又受到了新技术的影响。

同时，作物模型的开发人员一般不是专业的软件开发人员。

现有的流行的作物模型平台各有各的优点。DSSAT平台胜在平台成熟，作物模型种类完善。瓦赫宁根大学的模型系列在于模型数量庞大，模型种类完善。APSIM实际上更偏重于对于田间管理措施的模拟。STICS可以在一定程度上对模块进行替换，但其通过xml文档实现。

尽管对于作物模型的重构也一直在进行，但始终未能真正创造出非侵入式的组件化系统。

Crop Model MKS (Crop Model Micro Kernel System) 是针对上述问题而开发出来的一款作物模型平台。该平台采用微内核架构，主要负责对于作物模型各个模块的组织、管理与调用，使其可以兼容不同编程语言生成的组件，从而减小作物模型各组件开发与组件交互的难度，促进作物模型各开发团队之间的交流与协作。

编写语言 C++ 运行平台 Windows 10

**平台结构：**

CropModelMKS分为底层的代码库与操作界面两个部分。底层的代码均被封装在CropModelMKS.dll中，其中包括IParameter，IState，ISimulator，IAnalyzer，IWrapper等接口及其具体实现。

CropModelMKS包括底层的代码库与操作界面。底层代码库的所有代码均被封装在CropModelMKS.dll中。系统中主要包括Parameter、State、Simulator、Analyzer与Wrapper五个组件。组件采用COM技术制作，其中Parameter组件负责储存作物模型各模块所采用的参数。State负责储存作物模型各模块的输出参数。Simulator负责管理作物模型各模块在运行时的生命周期。Analyzer负责耦合作物模型与一些分析算法，如优化算法、敏感性分析算法等等。Wrapper组件负责包装作物模型各个模块，实现系统对不同编程语言的兼容。

系统的类图如图所示。系统基于组件对象模型（Component Objective Model，COM）技术进行开发，系统中的各接口均为简单ATL控件。

IState组件

参数数据以键值对的数据结构储存。作物模型各个模块的数据则储存在一个链表中。如图所示：

Parameter的类图如图所示：

interface IParameter : IDispatch

{

[id(1)] HRESULT Inquiry([in] BSTR name, [in, out]VARIANT\* value);

[id(2)] HRESULT Change([in] BSTR name, [in] VARIANT value);

[id(3)] HRESULT Push([in] INT hash, [in] BSTR file);

[id(4)] HRESULT Current([in] INT hash, [in] USHORT index);

};

其中，作物模型的各个组件通过Inquiry函数查询其所需要的参数数据。

本系统中，作物模型的参数数据通过XML文档的形式进行储存。Simulator将从XML文件中读取参数，然后通过Push函数将参数数据存入Parameter组件中。作物模型的各个模块则将通过Inquiry函数在Parameter组件中查询自身所需要的参数数据。Change函数为客户提供了改变某一特定参数的功能。由于参数文件涉及的库较多，Current函数是负责实现

interface IState : IDispatch

{

[id(1)] HRESULT Inquiry([in] BSTR name, [in] USHORT index, [in, out] VARIANT\* value);

[id(2)] HRESULT Write([in] BSTR name, [in] VARIANT value);

[id(3)] HRESULT Push([in] BSTR name);

[id(4)] HRESULT Attach([in] IDispatch\* idisp);

[id(5)] HRESULT Align();

};

Parameter

State

**Wrapper**

Wrapper统一了作物模型各模块的接口，使个模块可以方便地被Simulator调用。作物模型各模块符合单例模式。Wrapper将通过加载与卸载的方式调用各个模块。

在加载作物模型的模块时，Wrapper首先会判断模块的编程语言。由于静态语言如C++、Fortran等，只能通过库的形式调用，所以本平台要求静态语言制作的模块应基于COM技术，并且接口要与Wrapper保持一致。

在加载脚本语言时，Wrapper将首先创建一个该语言解释器的进程，然后在该解释器中通过字符串命令调用模块中的函数。

尽管MATLAB可以通过MATLAB Compiler将代码转换为COM组件，但一方面MATLAB Compiler有一定的获取难度，并且其生成的COM运行时依赖最新版本的MATLAB Runtime库，另一方面生成的COM的接口形式也并不符合CRopModelMKS的要求。所以设计时考虑在Wrapper中启动MATLAB进程，并利用字符串命令的形式在该进程中调用MATLAB命令。对于Python而言，尽管可以利用pyinstaller将Python代码包装为COM库，但其操作仍较为繁琐，如此操作失去了使用Python的意义。所以CropModelMKS在识别到Python语言后，Wrapper将初始化Python进程，然后在Python进程中

利用脚本语言开发模块时，也应遵循一定的规定。

以上是对开发者的所有要求。

以下是系统的工作原理

**Simulator**

Simulator组件中由Clock类控制整个模拟的生命周期。

**Analyzer**

**操作界面**

操作界面采用PyQt5开发。

数据

对于作物模型的各个组件而言，数据可以分为参数、状态与依赖项三种不同的类别。参数是在进行模拟时采用的公式所需要的数据；状态是组件操作的数据；依赖项是

进程间通信

为了提高组件之间数据传输的效率，组件通过共享内存的方式进行通信。共享内存通过组件Parameter与State实现。所有的参数数据均被储存在Parameter组件之中；模拟时所涉及到的状态变量均储存在

