**Rename:**

Variable->Cell

Namespace->Plasm (is this change necessary?)

GenTable->LookupTable

MPFTable->RecordTable

Accumulation->PlasmGroup

ReportVar->CellGroup

InputPrototype->BaseTable

Script->CellDefTable

**Model:**

Description: The largest class that creates everything else, including Input, Process and Output

**Input:**

Description: handles all the inputs of the Model, including all the Tables and Scripts

**BaseTable:**

Description: Previously called InputPrototype. It is the base class for all Tables. Scripts are Tables now. It is a class to store values. It contains 3 categories:

1. SysTable
2. DefinitionTable
3. ParameterTable:
4. ModelTable

**ParameterTable:**

Description: Inherited from Table. These ParameterTables contain values necessary for running the model (these values will be looked up and used by the Cells).

1. LookupTable (previously called GenTable)
2. RecordTable (previously called MPFTable)
3. PlasmSpecificTable (contain values specific to each Plasm. i.e. a LookupTable with the key being Plasms’ Name
4. GlobalTable (contain values for global constants)

**DefinitionTable:**

Description: Inherited from Table. These DefinitionTables contain information for different components of the model. However, the Model may not use all these components for calculation.

1. **CellDefGroupTable**

Description: contains information for different groups of Cell definition (not Cell class instances). Notice that the Cells included in a CellDefGroup can be specific to a particular Identifier (using the [] operator). If there is only 1 definition for the Cell, then the identifier is not necessary. Also, a CellDefGroup can include other CellDefGroup. e.g.

<CellDefGroup name="ACC2">

<Cells>

cell1[Identifier1], cell2

</Cells>

<OtherCellDefGroup>

ACC1

</OtherCellDefGroup>

</CellDefGroup>

1. **DimensionTable**

Description: contains information for different Dimensions used. e.g.

<Dimension name=”t” >

<PossibleValues category=”list”>

1-12,15

</PossibleValues>

</Dimension>

<Dimension name=”t2” >

<PossibleValues category=”function”>

def c():

return [i\*3 for i in [1,2,3]]

</PossibleValues>

</Dimension>

<Dimension name=”t3” >

<PossibleValues category=”enum”>

Enum1

</PossibleValues>

</Dimension>

1. **EnumTable**

Description: contains information for all Enum, e.g.

<Enum name=”Enum1” >

<PossibleValues category=”list”>

1-2,5

</PossibleValues>

<Tags>

“first”,”second”,”fifth”

</Tages>

</Enum>

1. **CellDefTable (previously called Script)**

Description: Contains definitions for all Cells. There may be a lot of CellDefTables available, but the Model will not use all these definitions. For the same cell, there may be different definition for different **Identifier**.

e.g. Script 1

<CellDef name=”aa1”>

<DEFINITION id=”first”>

def aa1(t):

return bb(t)+2

</DEFINITION>

<DEFINITION id=”second”>

def aa1(t):

return bb()+3

</DEFINITION>

</ CellDef >

1. **PlasmDefTable**

Description: contains information and definition for Plasms, including which Cell definition it contains, (the struture of plasm will be discussed later). It contains the following components:

1. Cells
2. InputTissue
3. OutputTissue
4. Dimensions

e.g.

<PlasmDef name”P1”>

<CellsDef>

<IndividualCells>

cell1,cell2

</ IndividualCells>

<CellDefGroup>

ACC2

</CellDefGroup>

</CellsDef>

<InputCellDefs>

<PlasmDef name=”P3”>

<CellDefs>

cell2

</CellsDef>

</PlasmDef>

</InputCellDefs>

<OutputCellDefs>

<PlasmDef name=”p2”>

<CellDefs>

cell1

<CellsDef>

</PlasmDeF>

</OutputCellDefs>

<Dimensions>

<Dimension name=t>

<Range>

1,3,5

</Range>

</Dimension>

<Dimension name=t3>

<Range>

1,2,5

</Range>

</Dimension>

</Dimensions>

</PlasmDef>

**ModelTable:**

Description: contains information about the structure of the model.

1. **ModelStructureTable**

Description: contains information about which Plasms will be included in the Model

<Structure name =”M1”>

<Plasms>

p1,p2,p3

</Plasms>

</Struture>

1. **PlasmGroupTable** (previously called Accumulation)

Description: contains information for different groups of Plasm instances (not Plasm definitions), e.g.

<PlasmGroup name="ACC2">

PROD1,PROD2

</PlasmGroup>

1. **TissueTable (CellGroupTable)**

Description: contains information for different groups of Cell instances (not Cell definitions). Tissue will be explained below.

<Tissue name=”t1”>

<Plasm name=”p3”>

<Cells>

cell2

<Cells>

</Plasm>

<Plasm name=”p4”>

<Cells>

cell2,cell3

<Cells>

</Plasm>

<Cells>

p5.cell2, cell5

</Cells>

</Tissue>

1. **PlasmTable**

Description: Create Plasm instances based on Plasm Definitions

<Plasm name=”p”>

<PlasmDef>

P1

</PlasmDef>

<OutputFormat>

<OutputOption>”byRecord”</OutputOption>

<DimensionPoint>

<Dimension name =”t”>

<value mode=”Cell”>Cell1</value>

</Dimension>

</DimensionPoint>

<CellGroup>C1</CellGroup>

</OutputFormat>

</Plasm>

**SysTable:**

Description: contains information about the settings and system

1. **GlobalSettingTable**

Description: contains all the settings set up by the user.

**Process:**

Description: load the model, run the model

**Cell:**

Description: Their definitions are written in Scripts. Cells are contained in Plasms. Each Cell has a Cache. There are 3 timing for Cache reset:

1. Caches can be reset for each record
2. Caches can never be reset
3. Caches can be reset using some user-defined criteria

There are several types of Cell:

1. NormalCell

May contain random numbers.

Their Caches will be reset for each record, or using some user-defined criteria.

* 1. GlobalCell

Contain values for GlobalTable. Each GlobalCell only represents 1 value of GlobalTable. They are basically constants.

* 1. PlasmSpecificCell

contain values for PlasmSpecificTable. Their caches will not be reset after setting of initial values.

* 1. RecordCell

contain values for Record, e.g. there can be a RecordCell for “sex” if there is a column in RecordTable called “sex”.

* 1. DimensionCell

Representing current dimensions. e.g. t for time and s for scenario.

* 1. BooleanCell

Inherited from NormalCell.

Their return type must be boolean. They can be used for Cache reset (point 3 of “Cache reset” section).

1. AggregateCell

Their caches will not be reset. Only AggregateCell can be used as input Tissue and output Tissue transferring information between Plasms.

1. PreprocessCell

They are used to preprocess the input Tissue transferred to the Plasm.

**Tissue:**

Description: Each Tissue represents a group of Cells (not CellDef). Notice that it is “Cells” but not “CellDefinitions”. “p5.cell2” means Cell2 in Plasm p5. “cell5” means all the Cell named “cell5” in different Plasms e,g,

<Tissue name=”t1”>

<Plasm name=”p3”>

<Cells>

cell2

<Cells>

</Plasm>

<Plasm name=”p4”>

<Cells>

cell2,cell3

<Cells>

</Plasm>

<Cells>

p5.cell2, cell5

</Cells>

</Tissue>

**Plasm:**

Description: They are like namespaces. “PlasmDefTable” defines different Plasm definitions. Several Plasm instances can be created using the same Plasm definition. In a Plasm, different Cell definitions will be loaded and Cells are created and later calculated. Plasms can transfer information between each other by packages of AggregateCells (see example XML of PlasmDefTable) called input Tissue and output Tissues. Those input Cells must be preprocessed by PreprocessCell in order to be used by other Cells, e.g.

Plasm 1 contains Cell S. It put Cell S in an Input Tissue and send to Plasm 3.

Plasm 2 also contains Cell S. It put Cell S in an Input Tissue and send to Plasm 3.

For Plasm 3, its input Cells are Plasm1.S and Plasm2.S. It has its own PreprocessCell S to handle Plasm1.S and Plasm2.S, e.g.

definition of “S” of Plasm 3

def S():

sum=0

for each Tissue in AllInputTissue:

if Cell “S” exist in the InputTissue then

sum=sum+ value of InputTissue’s S

All the normal cells can only dependent on

1. LookupTables
2. Other Cells (including PreprocessCells, but not those input Cells transferred from other plasms)
3. RecordTable

Also, different Plasms can have different Dimensions. e.g. Plasm A has one Dimension called “t”, which contain a list [1,2,3,4,5]. Then all the AggregateCells in OutputTissue should only dependent on Cells that have only 1 dimension “t”.

**Organ:**

Description: A collection of Plasms

**Output:**

Description: A class that handles all the outputs. Basically, there are 2 modes of outputs:

1. ByRecord

Under this mode, values will be stored for each Record.

1. Aggregate

This mode uses specified AggregateCells of Plasms as outputs

**Proposed Flow of the model:**

1. **Model.Load()**

Load all Tables.

1. **Model.Run()**
2. Checking phases:
3. Cyclic conditions between Plasms

For each Plasm,

1. check cyclic condition between Cells (regardless of dimensions)
2. check cyclic condition between Cells (take dimensions into account)
3. check Tables’ formats are all correct
4. Transform to LLVM scripts
5. determine the sequence of calculation of Cells
6. Run the model