E-cell项目

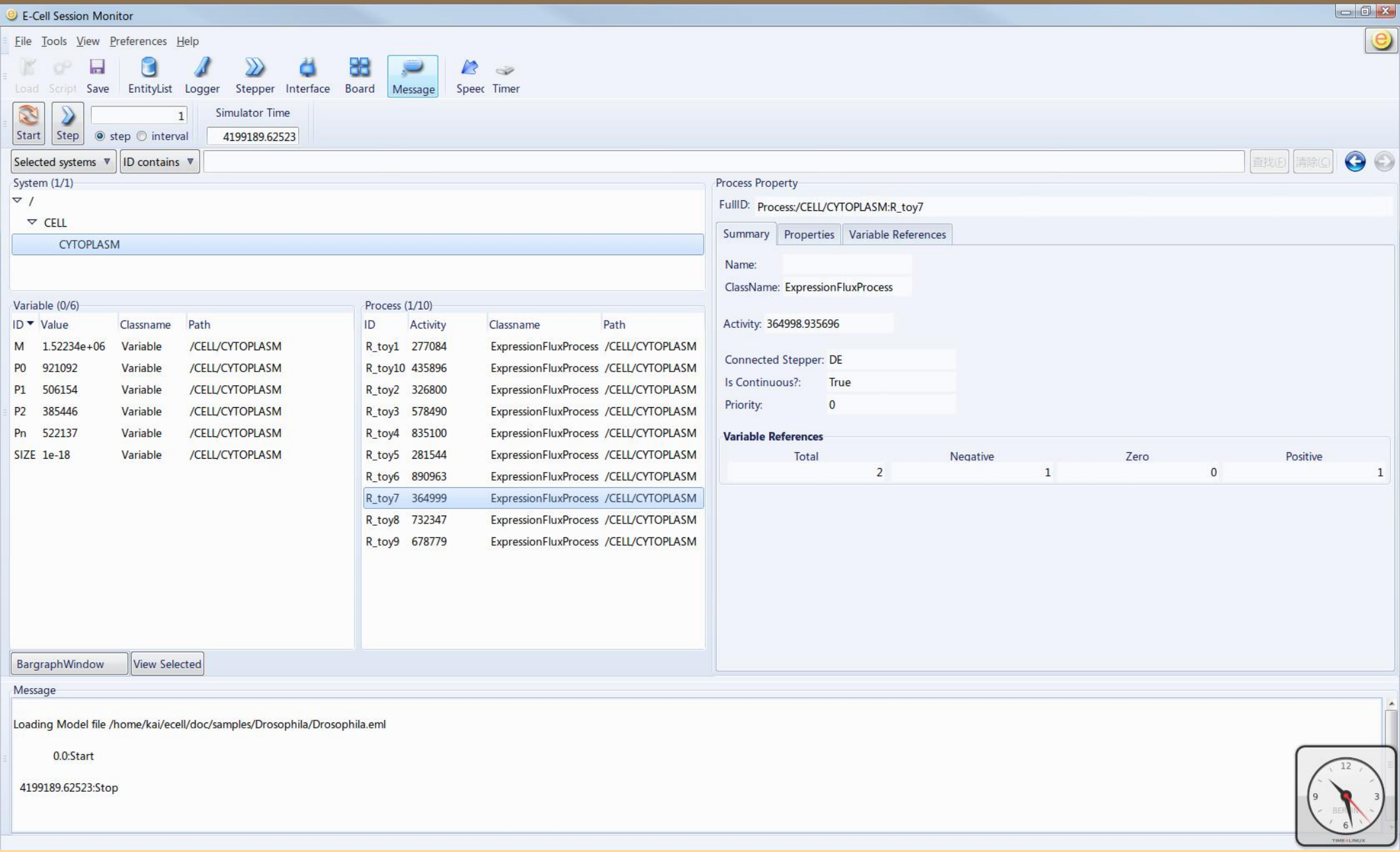
- E-Cell项目是一个目标是模拟和在计算机上重建生物现象，和开发能够精细地模拟全细胞所必要的理论、技术和软件平台支持国际研究项目 。
- 虽然这个项目成员是日本Keio 大学的先进生物科学的研究所的教员和学生，但他们的工作是与其他的大学和研究所协同共同完成的。软件开发和建模环境，模拟软件，和数学分析是与各种不同建模例如红血球新陈代谢，细菌的信号转换和线粒体新陈代谢项目协作完成的。

E-cell系统

- E-Cell系统是一套建模,模拟和分析大规模复杂系统例如生物细胞的面向对象的软件。第一版E-Cell模拟环境(E-Cell SE)在1999发布,随后开始Windows版E-Cell ver2。软件开发现在是ver3, 重建系统目标是为细胞模拟界提供一个公共的高度灵活性的和高性能的软件环境。
- Ver3 瞄向一个统一的模拟平台，可以综合任何不同的模拟算法，包括基于微分的模型， 扩散-反应, Gillespie算法, 细胞的自动控制和GMA/S-系统。它被设计成能在一个有子系统组成的细胞模型上有效地操作不同时间范围和不同空间分辨率的模拟。

E-cell系统

- E-Cell 3 核心模拟软件是一组Python 语言解释器的扩展模块，用c++/c/python写的。利用E-Cell模型描述语言EML (Model description Language), XML的子集,描述模型。为了能够跨平台交换模型，也正在开发对系统生物学标签语言SBML（Systems Biology Markup Language)的支持。
- E-Cell是一个开源项目（OpenSource）假若不侵犯它的GPL (GNU General Public License)许可，任何人可以修改和再分发这个软件。软件环境正在由E-Cell协会的成员开发，包括三井知识产业有限公司（Mitsui），国际Web和基因组信息协会，三菱空间软件有限公司等。


E-Cell Session Monitor
File Tools View Preferences Help
Load Script Save EntityList Logger Stepper Interface Board Message Speed Timer
Start Step 1 step interval Simulator Time 4199189.62523
Selected systems ID contains
System (1/1)
/
CELL
CYTOPLASM
Variable (0/6)
ID Value Classname Path
M 1.52234e+06 Variable /CELL/CYTOPLASM
P0 921092 Variable /CELL/CYTOPLASM
P1 506154 Variable /CELL/CYTOPLASM
P2 385446 Variable /CELL/CYTOPLASM
Pn 522137 Variable /CELL/CYTOPLASM
SIZE 1e-18 Variable /CELL/CYTOPLASM
Process (1/10)
ID Activity Classname Path
R_toy1 277084 ExpressionFluxProcess /CELL/CYTOPLASM
R_toy10 435896 ExpressionFluxProcess /CELL/CYTOPLASM
R_toy2 326800 ExpressionFluxProcess /CELL/CYTOPLASM
R_toy3 578490 ExpressionFluxProcess /CELL/CYTOPLASM
R_toy4 835100 ExpressionFluxProcess /CELL/CYTOPLASM
R_toy5 281544 ExpressionFluxProcess /CELL/CYTOPLASM
R_toy6 890963 ExpressionFluxProcess /CELL/CYTOPLASM
R_toy7 364999 ExpressionFluxProcess /CELL/CYTOPLASM
R_toy8 732347 ExpressionFluxProcess /CELL/CYTOPLASM
R_toy9 678779 ExpressionFluxProcess /CELL/CYTOPLASM
BargraphWindow View Selected
Process Property
FullID: Process:/CELL/CYTOPLASM:R_toy7
Summary Properties Variable References
Name:
ClassName: ExpressionFluxProcess
Activity: 364998.935696
Connected Stepper: DE
Is Continuous?: True
Priority: 0
Variable References
Total Negative Zero Positive
2 1 0 1
Message
Loading Model file /home/kai/ecell/doc/samples/Drosophila/Drosophila.eml
0.0:Start
4199189.62523:Stop

E-cell运行所需要的文件

- EML格式的模型文件
- 共享库
- 自动运行模拟过程的ESS脚本

E-cell命令

- ecell3-em2eml
- ecell3-dmc
- ecell3-session-monitor
- ecell3-session
- $ECELL3_DM_PATH=.:/home/example/mydms
- $ export ECELL3_DM_PATH

E-cell建模

实体对象：

- 变量
- 过程
- 系统

Stepper对象

E-cell建模

对象ID

字母、数字、下划线的组合

路径

“/”根系统，“.”当前系统，“..”super-system

FullID

EntityType:SystemPath:ID

FullPN

FullID :property_name

E-cell建模

对象属性的类型

实数：1.03，3.33e+10

整数： 2，100

字符串： 单引号，双引号，三引号，无引号。

_C10_A, Process:/A/B:P1, "It can include spaces if double-quoted.",’single-quote is available too, if you want to use "double-quotes" inside.’

列表：[A 10 [1.0 "a string" 1e+10]]

类型间的自动转换

最少惊讶原则

```
Stepper ODEStepper( ODE_1 )
{# no property}
System System( / )
{StepperID ODE_1;
Variable Variable( SIZE )
{
Value 1e-18;
}
Variable Variable( S )
{
Value 10000;}
Variable Variable( P )
```

TYPE CLASSNAME(ID)

"""INFO (optional)"""

{

PROPERTY_NAME_1 PROPERTY_VALUE_1;

PROPERTY_NAME_2 PROPERTY_VALUE_2;

...

PROPERTY_NAME_n PROPERTY_VALUE_n;

}

TYPE ：变量，过程，系统，stepper

classname：MassActionFluxProcess

宏和预处理

嵌入python代码

@(python expression)

如：

```
@{  def   f( str ):
                return str + ’ is true.’}
@f( ’Video Games Boost Visual Skills’ )

@include( ’foo.em’ )
```

模型的结构

Stepper0

Stepper1

.....

System0

System1

.....

System System(/)

{# ... properties of this System itself comes here..

Variable Variable(V0) {}

Variable Variable(V1) {}

连接实体对象和Stepper

```
Stepper SomeClassOfStepper( STEPPER0 ){}
Stepper AnotherClassOfStepper( STEPPER1 ) {}

System System( / ){# connected to STEPPER0
StepperID STEPPER0;

Process AProcess( P0 )# connected to
   STEPPER0
{# No StepperID specified.}
```

连接过程和变量

[reference_name FullID coefficient accessor_flag]

[reference_name FullID coefficient]

[reference_name FullID]

coefficient：整数值，默认0

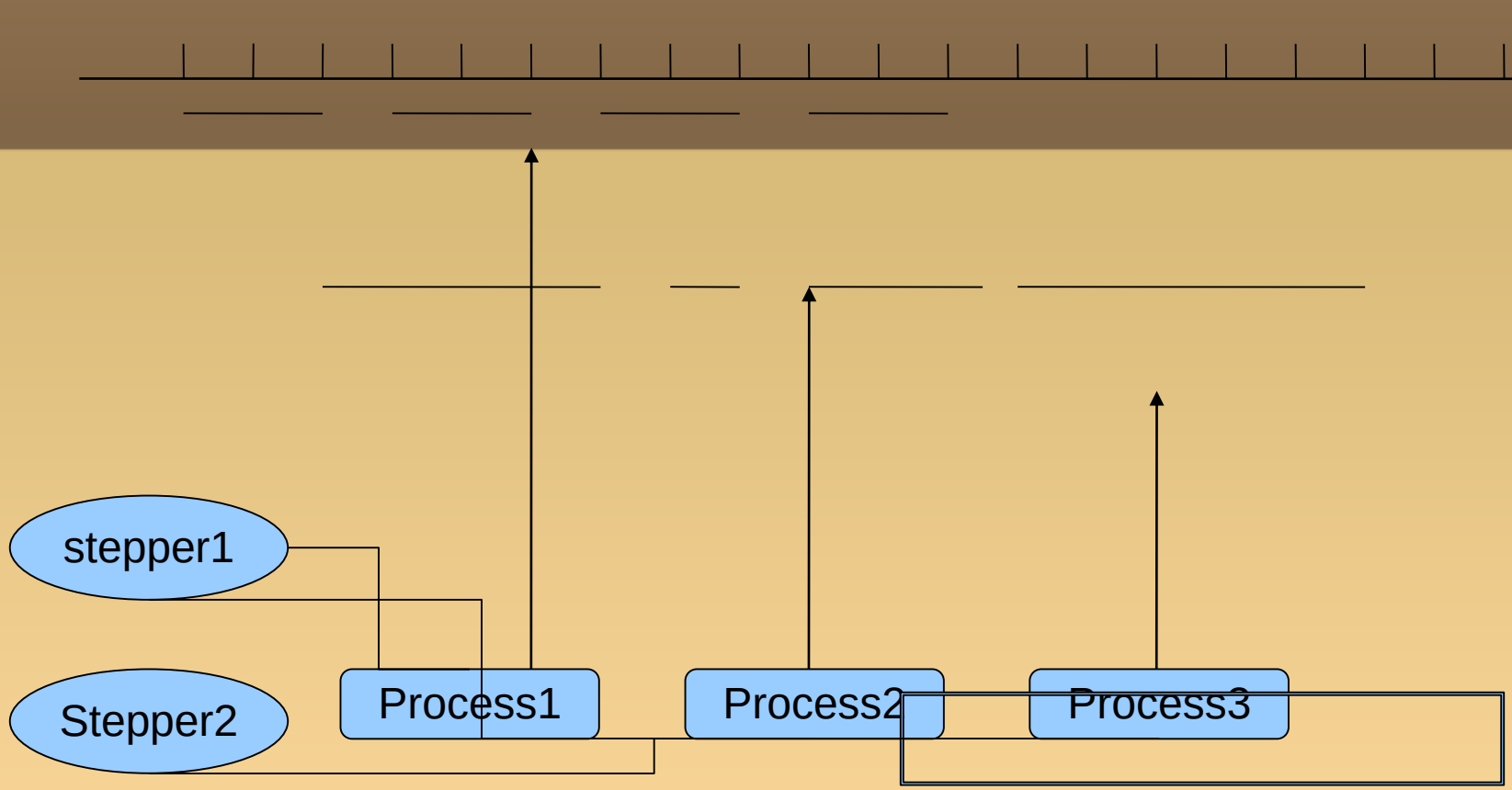
accessor_flag：1或0，默认1

System System(/){

Variable Variable(S) {}

Variable Variable(P) {}

Stepper



离散型classes

DiscreteEventProcess

NRStepper：Gillespie’s stochastic algorithm

GillespieProcess

DiscreteTimeStepper：自定义timer

PassiveStepper：监听特定事件

PythonProcess：自定义initialize()，fire()

PythonEventProcess：updateStepInterval()

连续型classes

Ordinary Differential Equation

Differential-Algebraic Equation

ODEStepper：Tolerance，MinStepInterval

DAEStepper

MaxStepInterval

FixedODE1Stepper

MassActionFluxProcess：k

连续型classes

pre-defined reaction rate classes

PythonFluxProcess

Generic differential-algebraic Steppers

Algebraic Processes

ExpressionAlgebraicProcess

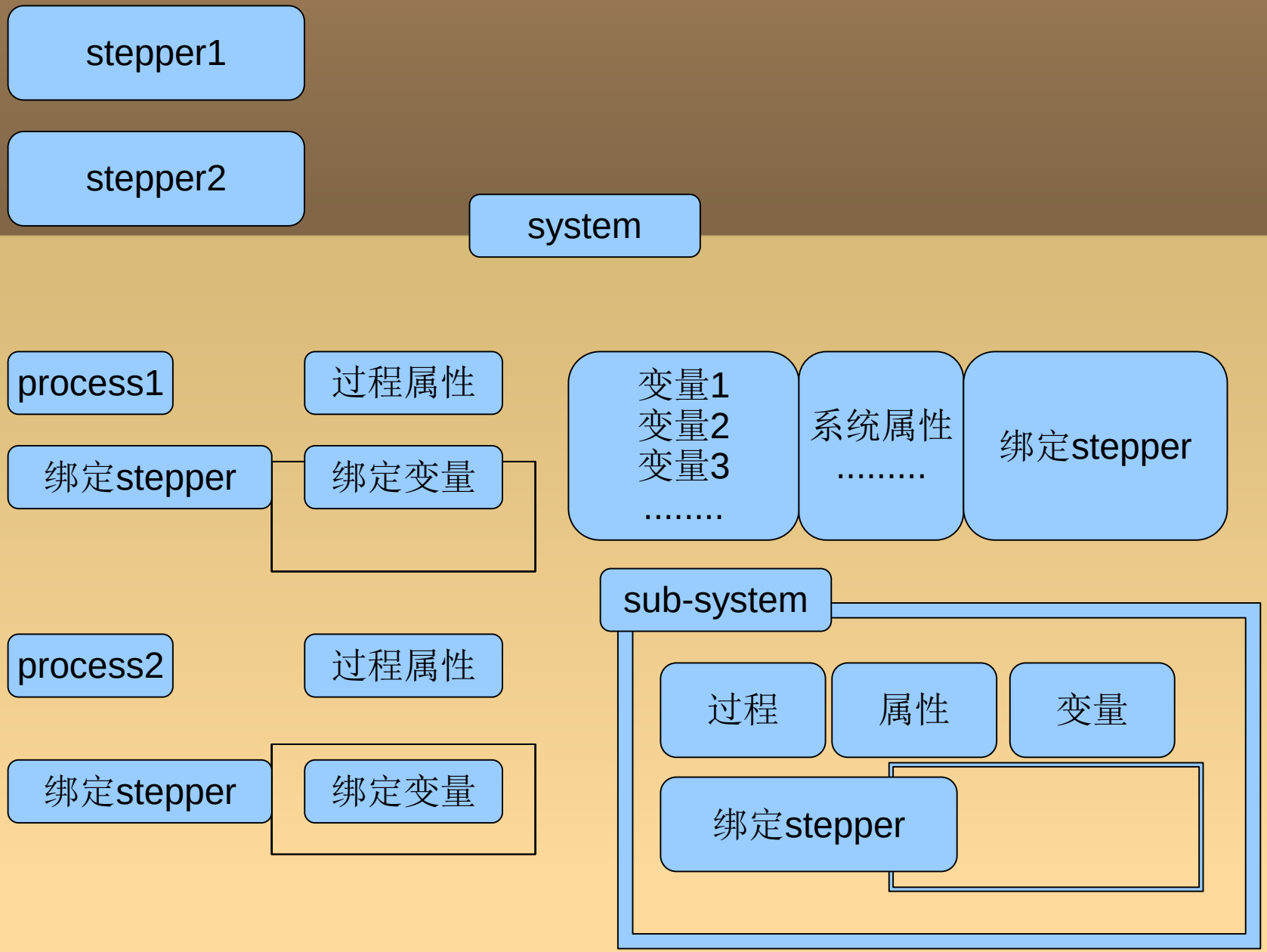
单位

时间：秒

体积：升

浓度：mol/L

Num/L


stepper1
stepper2
system
process1
过程属性
绑定stepper
绑定变量
变量1
变量2
变量3
.....
系统属性
.....
绑定stepper
sub-system
process2
过程属性
过程
属性
变量
绑定stepper
绑定变量
绑定stepper

ESS脚本

E-Cell Session Script

- 装载EML文件
- 设置程序参数
- 运行程序
- 数据处理及保存

ecell3-session [-f model.eml] [-e] ess.py

ecell3-session [-f model.eml]

ESS

运行：

ecell3-session>>> loadModel(’simple.eml’)

simple.eml, t=0>>>

self.run(10) 或者 theSession.run(10)

Step()

getCurrentTime()

message(message)

模型数据操作

创建stub：

createEntityStub()

createStepperStub()

createLoggerStub()

例： s1=createEntityStub(’Variable:/CELL/MT1:ADP’)

aStepperStub=createStepperStub(’STEPPER_01’)

aLoggerStub=createLoggerStub(
’Variable:/CELL/MT1:GLUCOSE:Concentration’）

模型数据操作

```
aStepperStub=createStepperStub('STEPPER_01'
    )
if aStepperStub.exists():
# it already exists
else:
aStepperStub.create()
```

模型数据操作

aValue = aStub.getProperty('Activity')

aValue = aStub['Activity']

aStub.setProperty('Activity', aNewValue)

aStub['Activity'] = aNewValue

只对EntityStub和StepperStub有效

Loggerstub：

getData()

getData(starttime [, endtime])

getData(starttime, endtime, interval)

```
loadModel( 'simple.eml' )
ATP = createEntityStub( 'Variable:/CELL:ATP' )
while 1:
        ATPValue = ATP[ 'Value' ]
        message( 'ATP value = %s' % ATPValue )
        if ATPValue <= 1000:
                break
        run( 10 )
message( 'Stopped at %s.' % getCurrentTime() )
```

操作ECD数据文件

import ecell 或者 import ecell.ECDDataFile

aDataFile = ecell.ECDDataFile(DATA)
aDataFile.save(’datafile.ecd’)

aDataFile = ecell.ECDDataFile()
aDataFile.load(’datafile.ecd’)
DATA = aDataFile.getData()

ECD文件

数据名称

setDataName(name)， getDataName()

标签

setLabel(labels)， getLabel()

注释

setNote(note)，getNote()

ECD文件

#DATA:

#SIZE: 5 1010

#LABEL: t value avg min max

#NOTE:

#

#----------

0 0.1 0.1 0.1 0.1

...