Simulation Engine

Visitors

Concepts

Scheduler

curTime : Time = 0

Instance() : Scheduler*
Scheduler()
put(ev : Event*) : void
getCurTime() : Time
processOneEvent() : bool
processNow() : bool

<<Interface>>
ITimedElement
(from Abstract Concepts)

-target   1

0..1 | theScheduler

0..1 | -first

Event

id : ID = i
time : Time = tm

Event()
create()
getID()
getTarget()
getTime()
setTime()
getNext()
setNext()

-next

0..1

previous | 0..1

Visitor

- visitModel()
- visitUniformGenerator()
- visitExpGenerator()
- visitFlowSource()
- visitFlowSink()
- visitPipe()
- visitQueuedServer()
- visitSwitch1to3()
- visitCrossSemaphore()

VisTxtOutputGenerator

- out : ostream* = outputStream

- VisTxtOutputGenerator()
- output()
- writeNameType()
- writeDescrLn()

Abstract Concepts

Random Generators

Modeling Concepts

ModelElement

- idCounter : MEID = 0
- id : MEID = ++idCounter
- name : String
- descr : String

- ModelElement()
- ~ModelElement()
- accept()
- getID()
- getName()
- setName()
- getDescr()
- setDescr()

FlowElement

- counter : unsigned long = 0

- FlowElement()
- acceptFlow()
- process()
- send()
- getNumOfOccurrences()
- resetCounter()

source

*

0..1 +target

f...

ControlElement

IFlowSource

RandomGenerator

🔒 *urndNum : unsigned long = urndSeed*

🔹 *accept(vis : Visitor*) : void*
🔹 *getDistributionName() : String*
🔹 *getMeanValue() : double*
🔹 getRandom() : double
🔹 urnd() : double
🔹 *convert(x : double) : double*

UniformGenerator

🔒 lower : double = lowerBound
🔒 upper : double = upperBound

🔹 UniformGenerator()
🔹 accept()
🔹 getDistributionName()
🔹 getMeanValue()
🔹 convert()

ExpGenerator

🔒 mean : double = meanValue

🔹 ExpGenerator()
🔹 accept()
🔹 getDistributionName()
🔹 getMeanValue()
🔹 convert()

RandomTimedFlowElement
(from Abstract Concepts)

FlowElement
(from Abstract Concepts)

FlowSource

🔒 onOff : bool = true

🔹 FlowSource()
🔹 accept()
🔹 acceptFlow()
🔹 notify()
🔹 switchOn()
🔹 switchOff()
🔹 isOn()

FlowSink

🔹 FlowSink()
🔹 accept()
🔹 process()

IFlowSource

(from Abstract Concepts)

Abstract Concepts

Random Generators

Interfaces

Modeling Concepts

Visitors

Event

Sched

Simulation Engine

SimEng

Visitor

VTxtOutputGen

Visitor

VTxtOutputGen

VModelChecker

VModelDocumenter

VSimReporter

VModelChecker

VModelDocumenter

VSimReporter

RndGen

UniRndGen

ExpRndGen

RndGen

FlowSource

FlowSink

Pipe

Modeling

QueServer

Switch1to3

CrossSem

FlowSource   FlowSink   Pipe   QueServer

FlowSource   FlowSink   Pipe   QueServer

Switch1to3   CrossSem

Switch1to3   CrossSem

Model   AbstractConcepts   Interfaces

AbstractConcepts   Visitor

RndGen   Event

Visitor
- visitModel()
- visitUniformGenerator()
- visitExpGenerator()
- visitFlowSource()
- visitFlowSink()
- visitPipe()
- visitQueuedServer()
- visitSwitch1to3()
- visitCrossSemaphore()

VisTxtOutputGenerator
- out : ostream* = outputStream
- VisTxtOutputGenerator()
- output()
- writeNameType()
- writeDescrLn()

VisTxtOutputGenerator
- VisTxtOutputGenerator()
- output()
- writeNameType()
- writeDescrLn()

VisModelChecker
- VisModelChecker()
- outputViolation()
- visitUniformGenerator()
- visitExpGenerator()
- visitFlowSource()
- visitFlowSink()
- visitPipe()
- visitQueuedServer()
- visitSwitch1to3()
- visitCrossSemaphore()

VisModelDocumenter
- VisModelDocumenter()
- visitModelElement()
- visitFlowElement()
- visitUniformGenerator()
- visitExpGenerator()
- visitFlowSource()
- visitFlowSink()
- visitPipe()
- visitQueuedServer()
- visitSwitch1to3()
- visitCrossSemaphore()

VisTxtOutputGenerator
- VisTxtOutputGenerator()
- output()
- writeNameType()
- writeDescrLn()

VisSimulationReporter
- VisSimulationReporter()
- visitFlowElement()
- visitUniformGenerator()
- visitExpGenerator()
- visitFlowSource()
- visitFlowSink()
- visitPipe()
- visitQueuedServer()
- visitSwitch1to3()
- visitCrossSemaphore()

Sequence diagram participants:
- aClient
- theModel : Model
- aConcreteElementA : QueuedServer
- aConcreteElementB : ExpGenerator
- aConcreteVisitor : VisModelDocumenter

Messages:
- accept(aConcreteVisitor)
- visitModel(theModel)
- getElements( )
- accept(aConcreteVisitor)
- visitQueuedServer(aConcreteElementA)
- getID( )
- read name
- ....
- accept(aConcreteVisitor)
- visitExpGenerator(aConcreteElementB)
- getDistributionName( )
- getMeanValue( )
- ...

Scheduler

curTime : Time = 0

Instance() : Scheduler*
Scheduler()
put(ev : Event*) : void
getCurTime() : Time
processOneEvent() : bool
processNow() : bool

<<Interface>>
ITimedElement
(from Abstract Concepts)

-target   1

0..1 | theScheduler

0..1 | -first

Event

id : ID = i
time : Time = tm

Event()
create()
getID()
getTarget()
getTime()
setTime()
getNext()
setNext()

-next

0..1

previous | 0..1

6: getTime( )   10: setTime()

4: getTime( )

5: getNext( )

theScheduler   first

Instance :   e2   e1 :   <<destroyed>>   e2 :   next previous   e3 :
Scheduler   Event   next previous   Event   Event

7: setNext(ev)

3: getTime( )   next   previous

8: setNext(e2)   9: setTime()

                              <<new>>   <<new>>

2: put(this)   previous   next

: RandomTimedFlowElement   ev :
                                        Event

1: <<create>>

Sequence Diagram

: RandomTimedFlowElement	ev : Event	Instance : Scheduler	e1 : Event	e2 : Event	e3 : Event

<<create>>

put(this)

getTime( )

getTime( )

getNext( )

getTime( )

setNext(ev)

setNext(e2)

setTime()

setTime()

Collaboration Diagram

2: processOneEvent( )

7: first = first->getNext()

4: getTarget( )

1: processNow( )

3: setTime(0)

el

aClient

G Instance : Scheduler

first

8: <<destroy>>
theScheduler

e : Event

5: getID( )

id

6: notify(id)

L F

el :
ITimedEl...

<<Interface>>
IFlowSource

- switchOn()
- switchOff()
- isOn()

<<Interface>>
ITimedElement

- notify(id : ID) : v...

Sequence Diagram

aClient | Instance : Scheduler | e : Event | el : ITimedElement

processNow( )

processOneEvent...

setTime(0)

while (first!=0 ...

getTarget( )

getID( )

notify(id)

first = first->getNext()

<<destroy>>

Class Diagram

ModelElement

- idCounter : MEID = 0
- id : MEID = ++idCounter
- name : String
- descr : String

- ModelElement()
- ~ModelElement()
- *accept()*
- getID()
- getName()
- setName()
- getDescr()
- setDescr()

FlowElement

- counter : unsigned long = 0

- FlowElement()
- acceptFlow()
- process()
- send()
- getNumOfOccurrences()
- resetCounter()

source
*

0..1 +target

f...

ControlElement

IFlowSource

Model

- Model()
- ~Model()
- Instance() : Model*
- accept(vis : Visitor*) : void
- add(me : ModelElement*) : void
- remove(me : ModelElement*) : void
- getElements() : CollectionME*
- clear() : void

model ◆ 1

-elements *

ModelElement

FlowElement

RandomTimedFlowElement

- RandomTimedFlowElement()
- ~RandomTimedFlowElement()
- getRndGen()
- raiseEvent()
- *notify()*

ITimedElement

-rndGen

RandomGenerator
(from Random Generators)

0..1   1

---

source :
FlowElement

source

1: acceptFlow( )

2: counter++
3: process( )
4: send( )

target

this :
FlowElement

source

5: acceptFlow( )

target

target :
FlowElement

---

source :
FlowElement

this :
FlowElement

target :
FlowElement

acceptFlow( )

counter++

process...

send( )

acceptFlow( )

+leftTarget

FlowElement
(from Abstract Concepts)

0..1   0..1

+rightTarget

leftTurn

Switch1to3

- ltProb : double = 0.0
- rtProb : double = 0.0

- Switch1to3()
- ~Switch1to3()
- accept()
- process()
- getLTProb()
- getRTProb()
- setProbabilities()

rightT...

n   *

0..1

1   -rndGen

UniformGenerator
(from Random Generators)

RandomTimedFlowElement
(from Abstract Concepts)

FlowElement
(from Abstract Concepts)

FlowSource

onOff : bool = true

FlowSource()
accept()
acceptFlow()
notify()
switchOn()
switchOff()
isOn()

IFlowSource
(from Abstract Concepts)

FlowSink

FlowSink()
accept()
process()

RandomTimedFlowElement
(from Abstract Concepts)

Pipe

Pipe()
accept()
process()
notify()

QueuedServer

onOff : bool = true
queSize : unsigned long = 0
isServing : bool = false
maxQueSize : unsigned long = 0

QueuedServer()
accept()
process()
notify()
switchOn()
switchOff()
isOn()
isCurrentlyServing()
getQueueSize()
getMaxQueueSize()

IFlowSource
(from Abstract Concepts)

ControlElement
(from Abstract Concepts)

horizontal

CrossSemaphore

curDir : CrossDir = horizontal
tVer : Time = 1
tHor : Time = 1

CrossSemaphore()
accept()
setTimes()
getTimeVertical()
getTimeHorizontal()
getCurDir()
notify()

ITimedElement

+horSource   0..2

IFlowSource

+verSource   0..2

vertical