Modeling syntax and semantics of πDemos in AToM^3

Presented by Eugene Syriani
August 28th, 2006

Overview

- πDemos, the article
  - Context
  - Structure
- Modeling the syntax: The Meta-Model
  - Modified structure
  - Time
- Modeling the semantics: Graph Grammars
  - Rules
- Words in action…

Introduction

- G. Birtwistle: Calgary, Canada
  C. Tofts: Swansea, Wales

- Operational semantics of process-oriented simulation languages – Part 1: πDemos, 1993

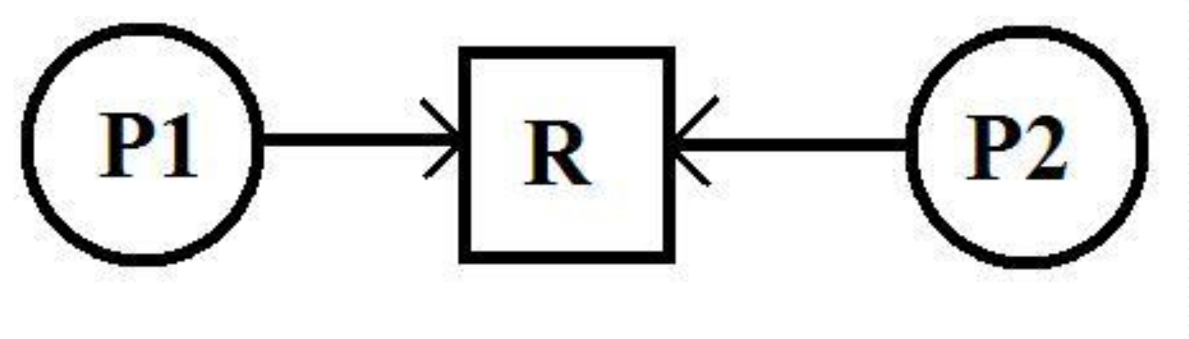
What is πDemos?

- πDemos is a small process-oriented discrete event simulation language. It is a TEXTUAL language

- πDemos operational semantics enables a complete control on
  - Synchronization
  - Event-list scheduling
  - Inter-process communication

πDemos’ structure

- Process vs Resource



πDemos' structure

- Process vs Resource

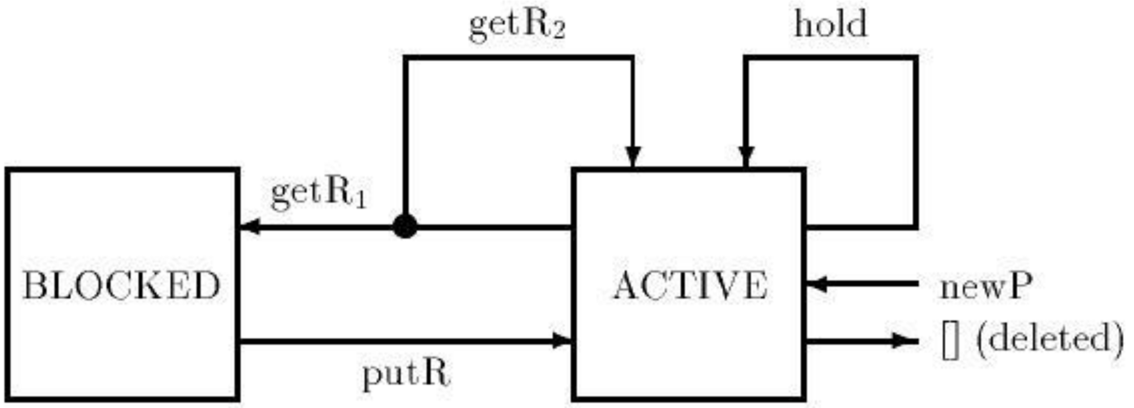
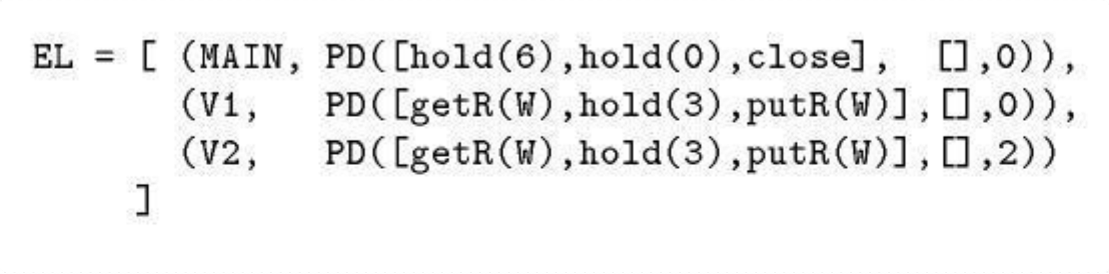


Figure 1: The ways processes change state

That’s all nice, but…



```
EL = [ (MAIN, PD([hold(6),hold(0),close],  [],0)),
       (V1,   PD([getR(W),hold(3),putR(W)],[],0)),
       (V2,   PD([getR(W),hold(3),putR(W)],[],2))
     ]
```

Now let's use 2006 technology

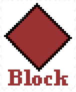
We want to model the syntax and semantics of πDemos

First, a Meta-Model

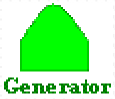
“Everything is a model”

Rules / Actions are *Blocks*

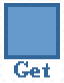
- Block



- Generator



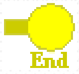
- Get



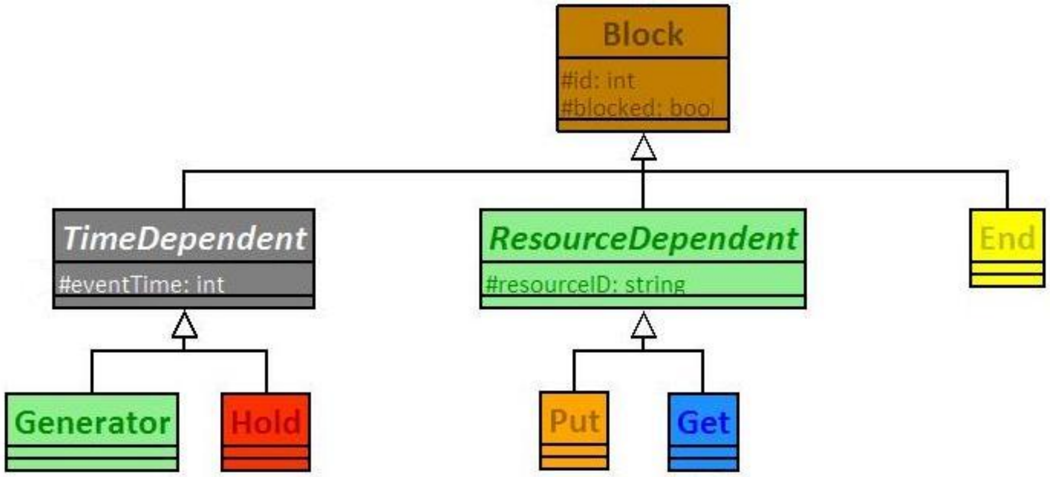
- Hold

- Put

- End



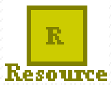
UML class diagram

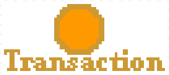


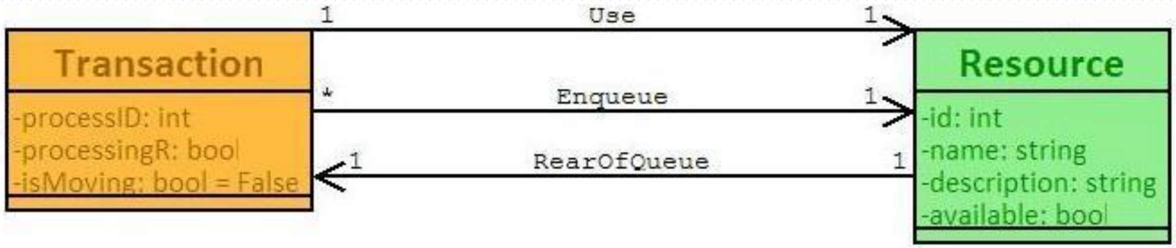
First, a Meta-Model

“Everything is a model”

Resource and transaction UML class diagram

- Resource 

- Transaction 



Transaction
-processID: int
-processingR: bool
-isMoving: bool = False

Resource
-id: int
-name: string
-description: string
-available: bool

First, a Meta-Model

"Everything is a model"

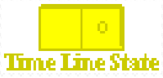
Time

- Time

- Head

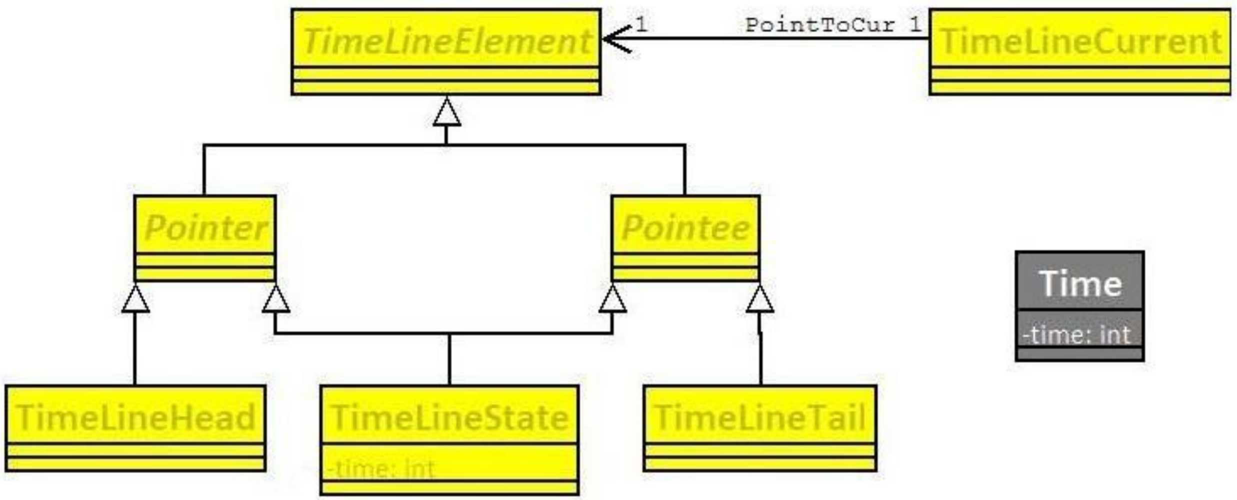
- Tail

- State

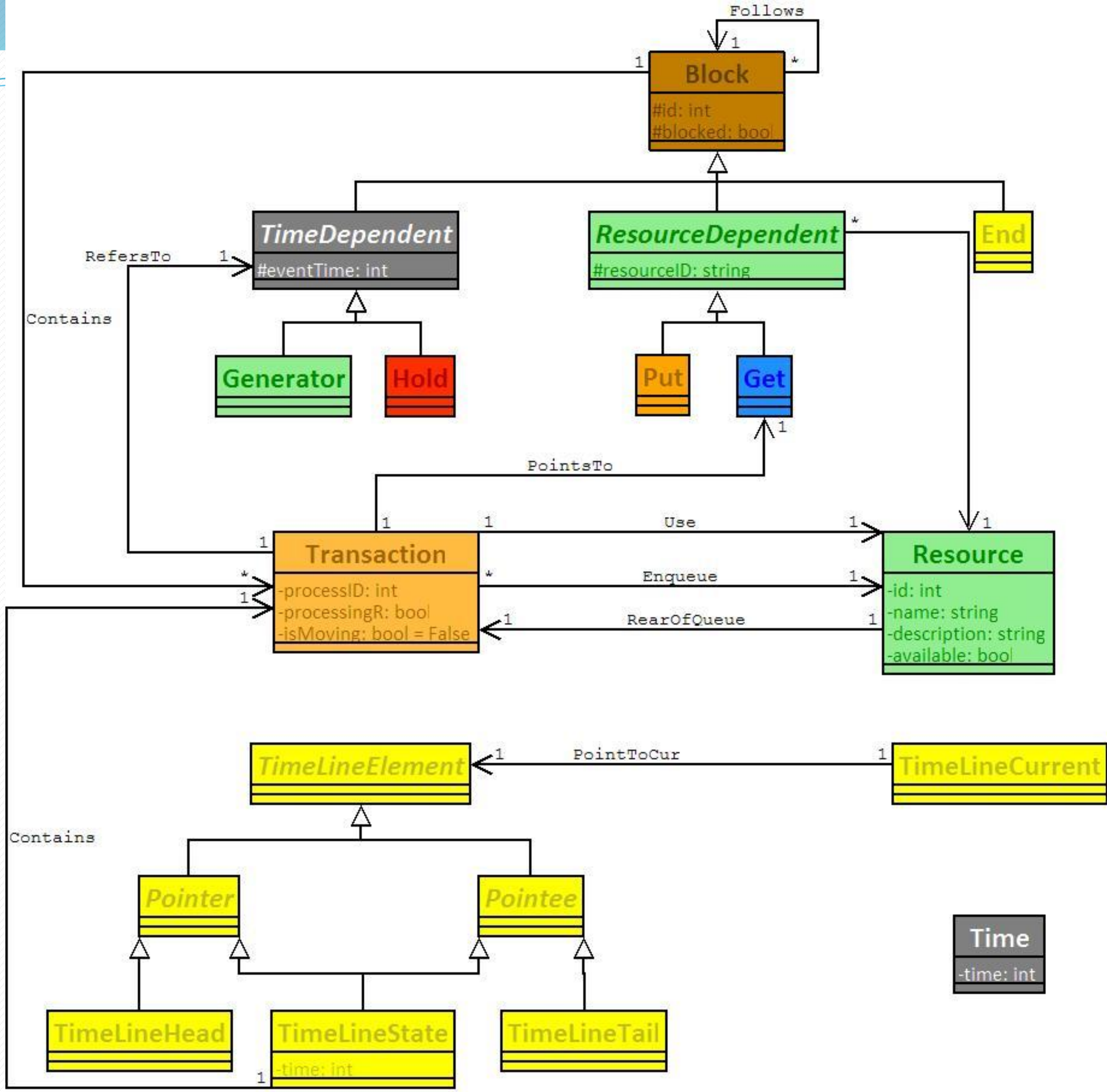


- Current

UML class diagram

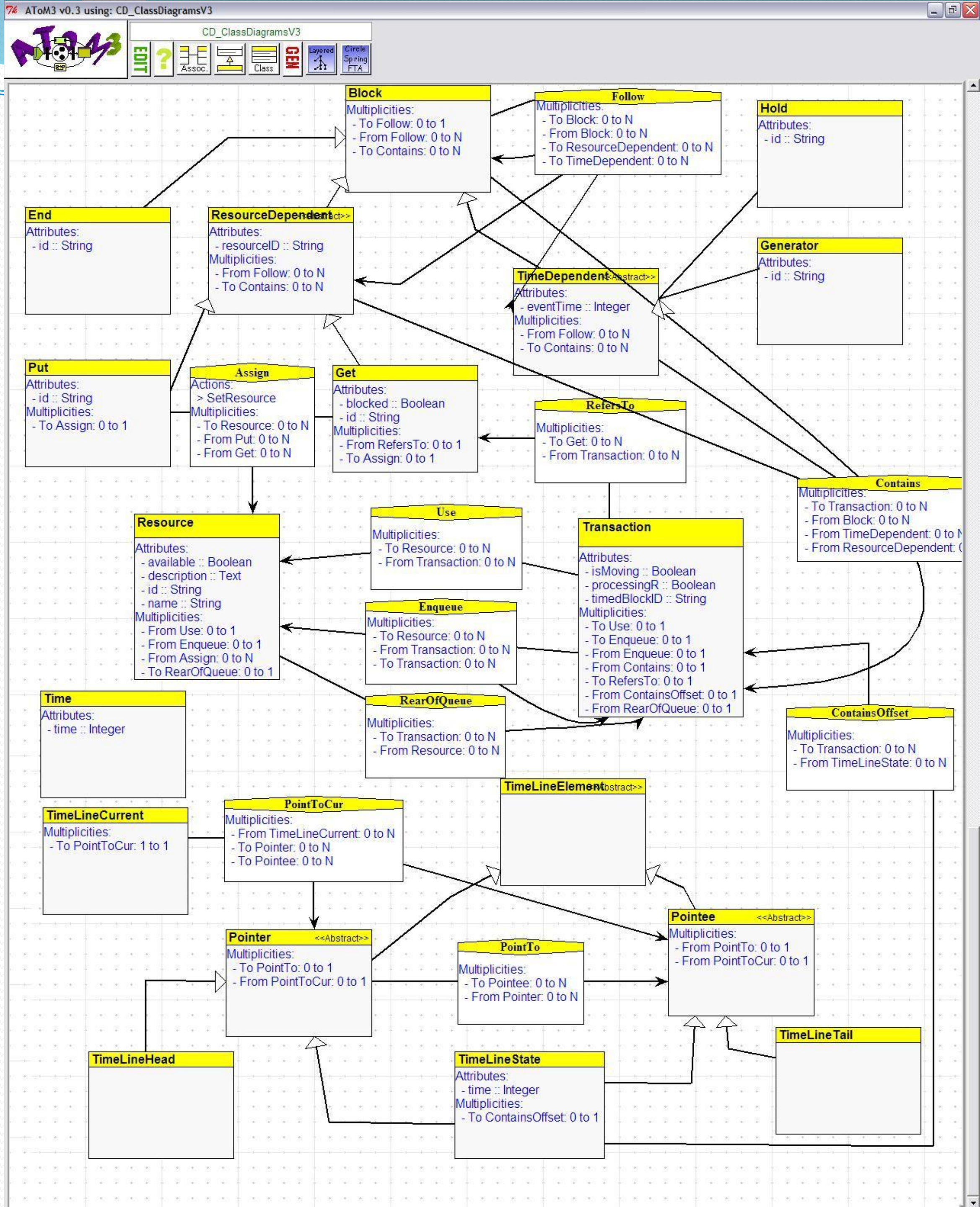


The Meta-Model



Using AToM^3

The big picture



Block
Multiplicities:
- To Follow: 0 to 1
- From Follow: 0 to N
- To Contains: 0 to N

Follow
Multiplicities:
- To Block: 0 to N
- From Block: 0 to N
- To ResourceDependent: 0 to N
- To TimeDependent: 0 to N

Hold
Attributes:
- id :: String

End
Attributes:
- id :: String

ResourceDependent <<Abstract>>
Attributes:
- resourceID :: String
Multiplicities:
- From Follow: 0 to N
- To Contains: 0 to N

Generator
Attributes:
- id :: String

TimeDependent <<Abstract>>
Attributes:
- eventTime :: Integer
Multiplicities:
- From Follow: 0 to N
- To Contains: 0 to N

Put
Attributes:
- id :: String
Multiplicities:
- To Assign: 0 to 1

Assign
Actions:
> SetResource
Multiplicities:
- To Resource: 0 to N
- From Put: 0 to N
- From Get: 0 to N

Get
Attributes:
- blocked :: Boolean
- id :: String
Multiplicities:
- From RefersTo: 0 to 1
- To Assign: 0 to 1

RefersTo
Multiplicities:
- To Get: 0 to N
- From Transaction: 0 to N

Contains
Multiplicities:
- To Transaction: 0 to N
- From Block: 0 to N
- From TimeDependent: 0 to N
- From ResourceDependent: 0

Resource
Attributes:
- available :: Boolean
- description :: Text
- id :: String
- name :: String
Multiplicities:
- From Use: 0 to 1
- From Enqueue: 0 to 1
- From Assign: 0 to N
- To RearOfQueue: 0 to 1

Use
Multiplicities:
- To Resource: 0 to N
- From Transaction: 0 to N

Transaction
Attributes:
- isMoving :: Boolean
- processingR :: Boolean
- timedBlockID :: String
Multiplicities:
- To Use: 0 to 1
- To Enqueue: 0 to 1
- From Enqueue: 0 to 1
- From Contains: 0 to 1
- To RefersTo: 0 to 1
- From ContainsOffset: 0 to 1
- From RearOfQueue: 0 to 1

Enqueue
Multiplicities:
- To Resource: 0 to N
- From Transaction: 0 to N
- To Transaction: 0 to N

Time
Attributes:
- time :: Integer

RearOfQueue
Multiplicities:
- To Transaction: 0 to N
- From Resource: 0 to N

ContainsOffset
Multiplicities:
- To Transaction: 0 to N
- From TimeLineState: 0 to N

TimeLineElement <<Abstract>>

TimeLineCurrent
Multiplicities:
- To PointToCur: 1 to 1

PointToCur
Multiplicities:
- From TimeLineCurrent: 0 to N
- To Pointer: 0 to N
- To Pointee: 0 to N

Pointee <<Abstract>>
Multiplicities:
- From PointTo: 0 to 1
- From PointToCur: 0 to 1

Pointer <<Abstract>>
Multiplicities:
- To PointTo: 0 to 1
- From PointToCur: 0 to 1

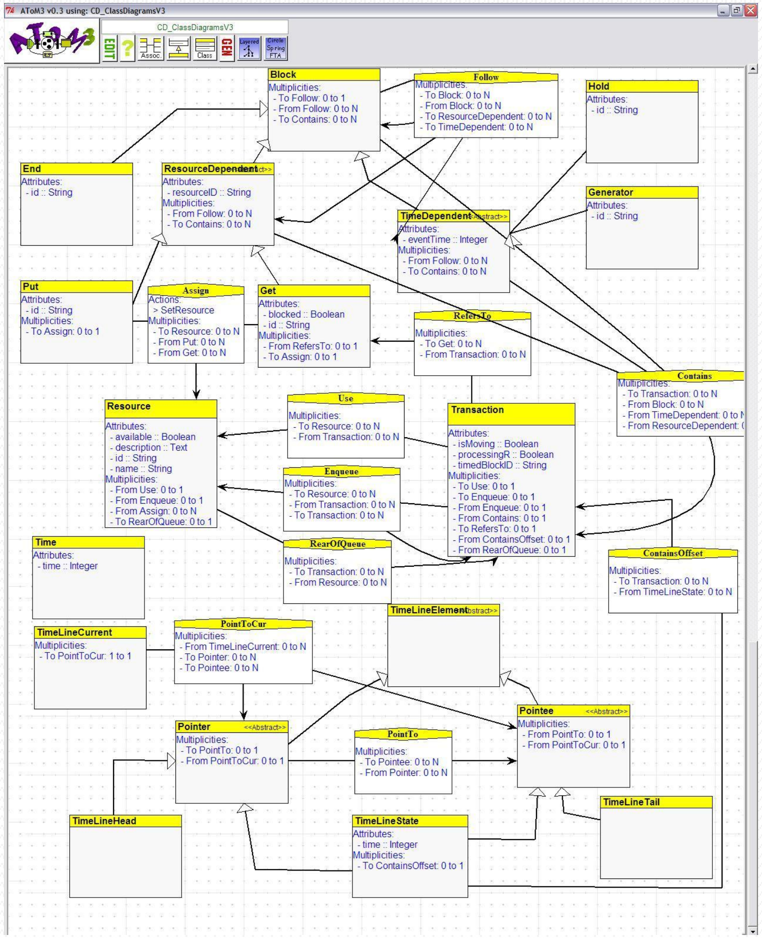
PointTo
Multiplicities:
- To Pointee: 0 to N
- From Pointer: 0 to N

TimeLineTail

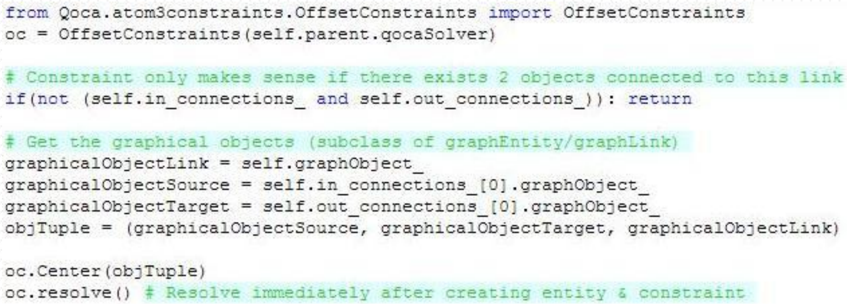
TimeLineHead

TimeLineState
Attributes:
- time :: Integer
Multiplicities:
- To ContainsOffset: 0 to 1

Using AToM^3

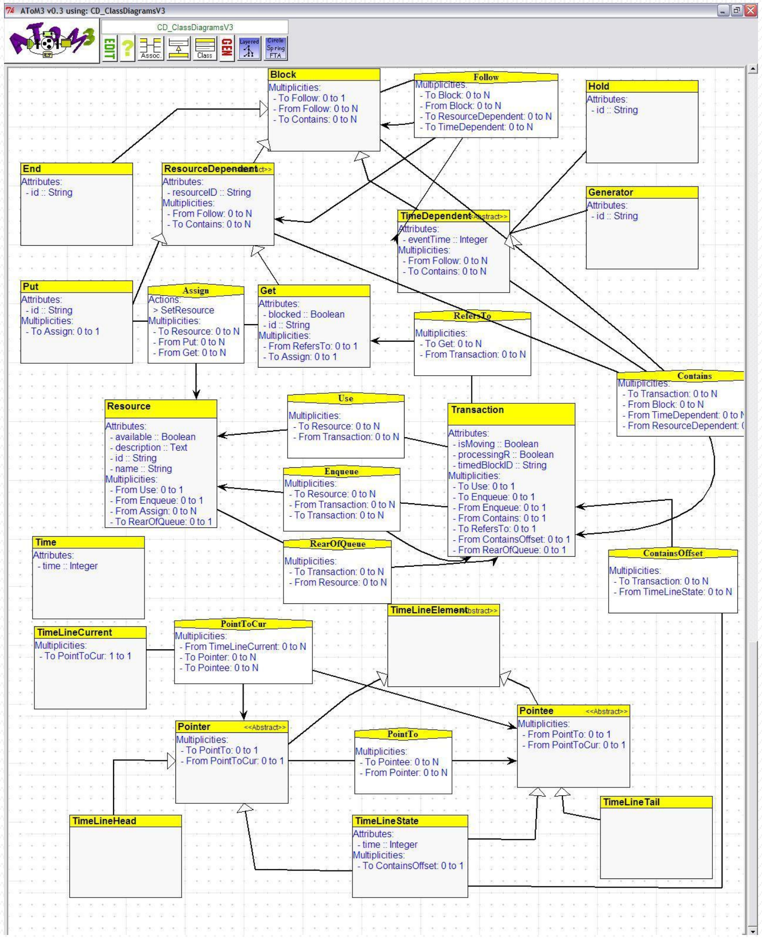


When *QOCA* is involved

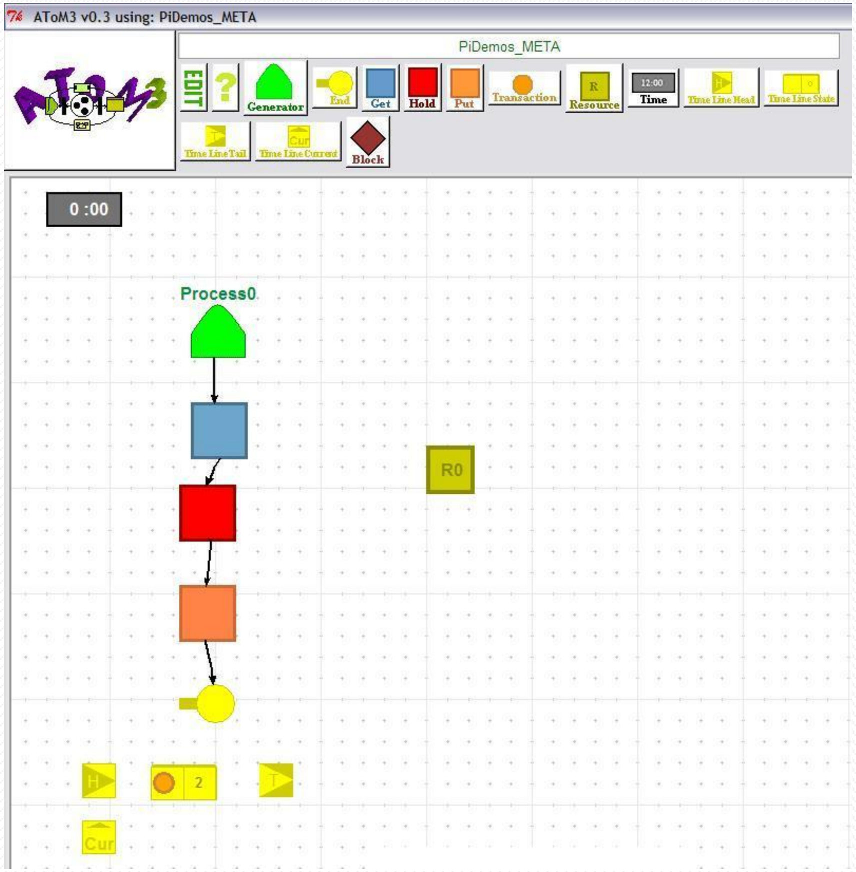


```
from Qoca.atom3constraints.OffsetConstraints import OffsetConstraints
oc = OffsetConstraints(self.parent.qocaSolver)

# Constraint only makes sense if there exists 2 objects connected to this link
if(not (self.in_connections_ and self.out_connections_)): return

# Get the graphical objects (subclass of graphEntity/graphLink)
graphicalObjectLink = self.graphObject_
graphicalObjectSource = self.in_connections_[0].graphObject_
graphicalObjectTarget = self.out_connections_[0].graphObject_
objTuple = (graphicalObjectSource, graphicalObjectTarget, graphicalObjectLink)

oc.Center(objTuple)
oc.resolve() # Resolve immediately after creating entity & constraint
```

Using AToM^3



The Meta Model



A model

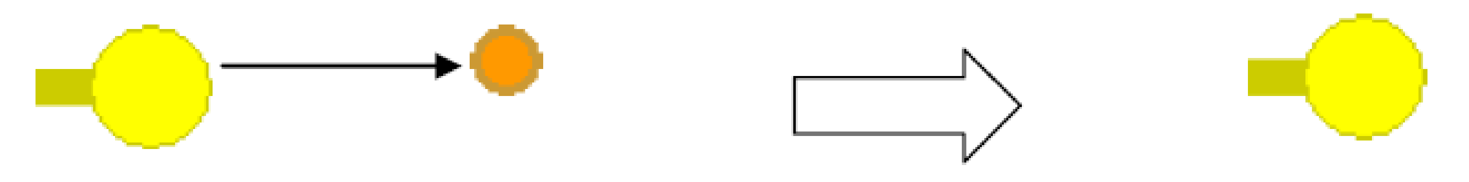
Now, let's give a meaning to the meta-model

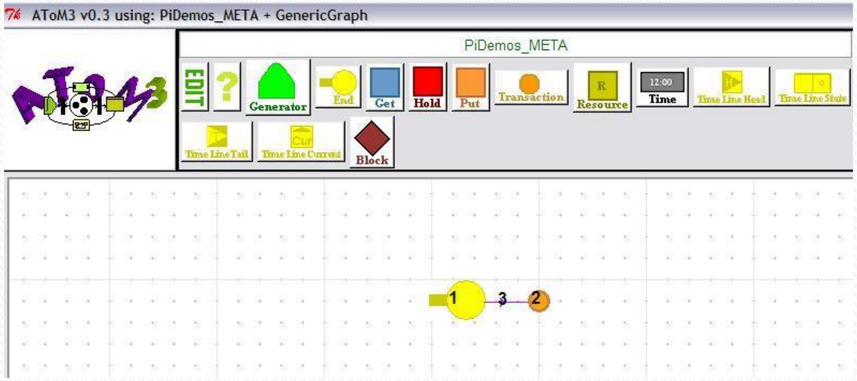
- Define a Graph Grammar
- 15 graph transformations are sufficient
- AToM^3 is a very nice and easy tool to use for graph transformations

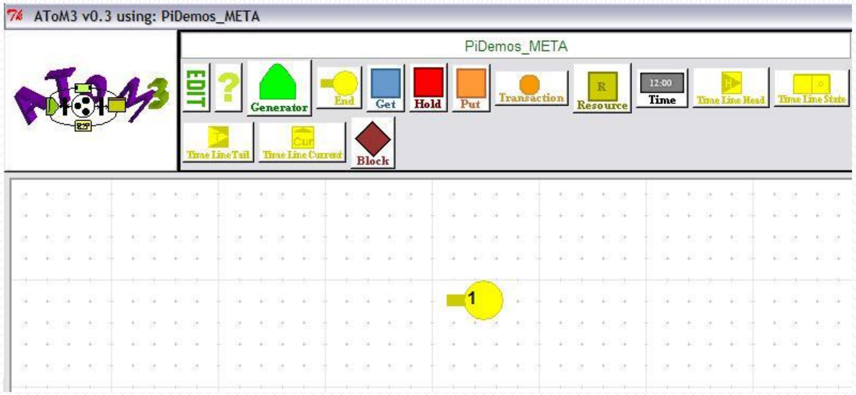
Example: *EXIT*

Define the LHS by means of labels on each item of a subgraph of a model instance.

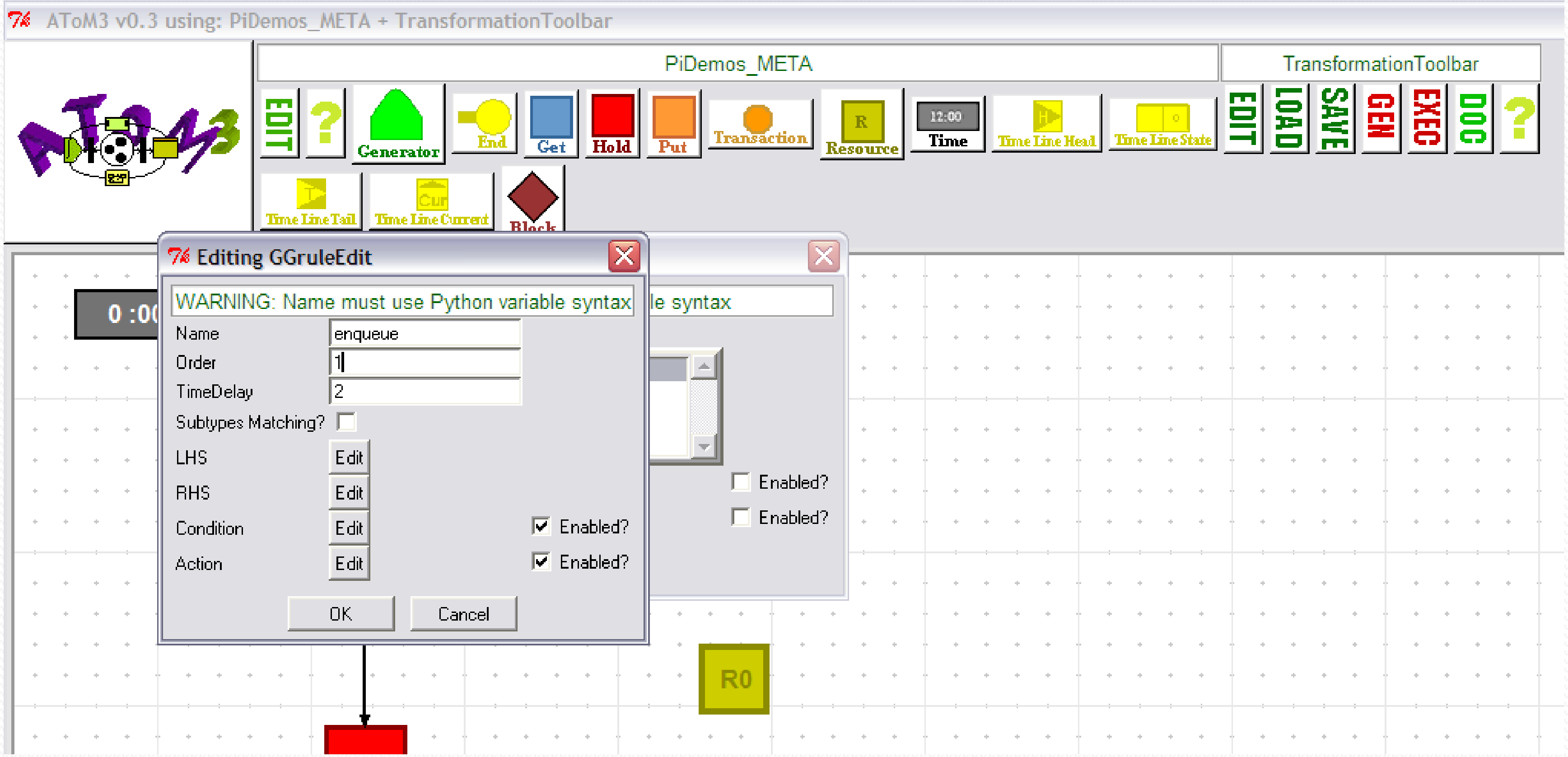
On the RHS, specify what it should be replaced by







Using AToM^3



AToM3 v0.3 using: PiDemos_META + TransformationToolbar

Editing GGruleEdit

WARNING: Name must use Python variable syntax

Name: enqueue

Order: 1

TimeDelay: 2

Subtypes Matching? □

LHS Edit

RHS Edit

Condition Edit ☑ Enabled?

Action Edit ☑ Enabled?

OK Cancel

The Graph Grammar

Words in action!

Further work

- Enable loops in processes, with conditions

- Non-determinism is possible
  - Closer to reality
  - Proof of termination is NP-Complete

- Let the process really do something, not just halt
  - Problem: time is not known in advance