Xholon – an ABM tool based on "Monkey Topology"

Ken Webb
Primordion

SwarmFest 2007
DePaul University, Computer Science, Chicago
July 12-14, 2007

ken@primordion.com

Abstract

Xholon is an open source multi-paradigm modeling, transformation and simulation tool, in which applications are constructed using XML and Java. It supports the Unified Modeling Language (UML 2.1), systems biology modeling, other types of modeling, and many of the features found in existing agent-based modeling tools. It focuses especially on integration of various approaches. It can, for example, simulate 2D-grid and agent-based models created using UML tools, and can combine multiple agent-based grids in the same model with agent migration between the grids. It combines an optional limited NetLogo-like syntax for navigating and exploring a flat grid space, with the standardized XML-based XPath expression language used for navigating and exploring hierarchical tree spaces. Xholon is based on Object-Oriented and UML concepts of class, composite structure, ports and bindings to connect objects, and active objects (agents) that contain state machines and other behaviors. The central structuring mechanism, what gives Xholon the flexibility it needs to support multiple paradigms, is the tree. This presentation will focus on this hierarchical tree-ness (what could be called "monkey topology"), and how this is used in Xholon to extend the basic concepts of agent based modeling. The presentation will be largely practical in nature, and will include a mini-demo.

Main Points

- Xholon is based on tree structures rather than grids.
  - Grids are superimposed over nodes in a tree.
  - Agents and agent relationships can be defined independently of the type of grid/network.
- Xholon combines an XPath syntax for navigating trees, and an optional XLogo syntax for navigating grids.
- The focus on trees provides flexibility, without having to give up ABM concepts of grids.
  - Navigation between multiple grids.
  - Complex internal structure for agents.

Background

- ObjecTime, IBM Rational Rose RealTime commercial products.
- Jointly published several academic papers.
  - These models are inaccessible to other researchers.
  - And, I started using these tools in new ways.
- I started to create my own modeling tool, based on Unified Modeling Language version 2.x (UML 2.x), and built using Java and XML.

Xholon

- A research project and software development tool that executes models of systems.
  - Including event-driven applications
  - Multi paradigm
  - Systems can be of arbitrary size
  - Embedded systems, controllers; Agent-based, swarms, etc.
- Goal of Xholon
  - To be able to model and execute a broad range of event-driven and complex systems, using same basic constructs in all of them.

Core Concept - Trees

- Everything in Xholon is a node in a tree.
- Tree nodes can cross connect with each other, using UML ports and connectors.
  - Networks, graphs, grids are overlaid on top of the primary tree structure.
- Any node in the tree can be an active object or agent, and can navigate the tree to interact with any other object.
  - Can move to a new part of the tree, can create new nodes and subtrees, can move/delete other nodes, can act on other passive nodes, etc.

Examples – Why trees are important

- UML state machines as tree structures.
- Genetic programming for tree manipulation.
- XML as a standard for presenting trees in text.
- Biological systems as one system inside another.
- Other data structures, such as grids, can be overlaid on top of trees.
- Agents can readily navigate to any other node.

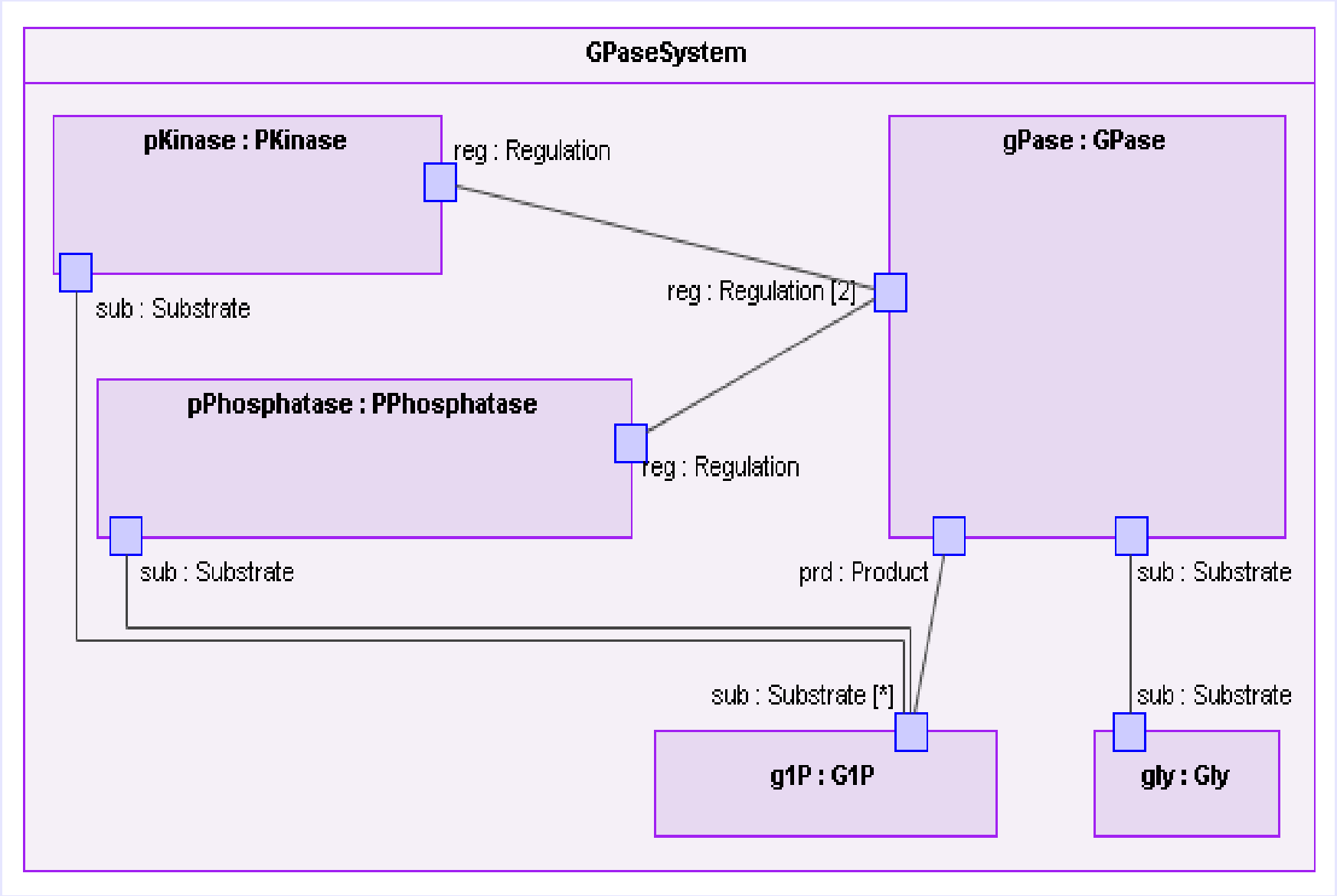
Xholon Modeling Constructs

- The basic Xholon modeling constructs are aligned with UML 2 constructs.
  - UML is a good starting point.
  - Topcased (open source), MagicDraw, Poseidon
- These constructs include - classes, composite structure, parts, ports, connectors, state machines.
- Active objects are agents, each with its own independent behavior.

Some Xholon ABM Features

- Moore, Von Neumann, Hex 2D neighborhoods.
  - Toroidal or non-wrapping grid.
- Line charts, histograms, view individual agents.
- Start, stop, pause/unpause, step, refresh.
- Colt probability distributions.
- Multiple grids in same model.
- Agents can be arbitrarily complex.
  - Can include state machines, neural networks, etc.
- Incorporates ECJ to evolve agent behaviors.
- Implements all 16 “stupid models”.

Composite Structure, Ports



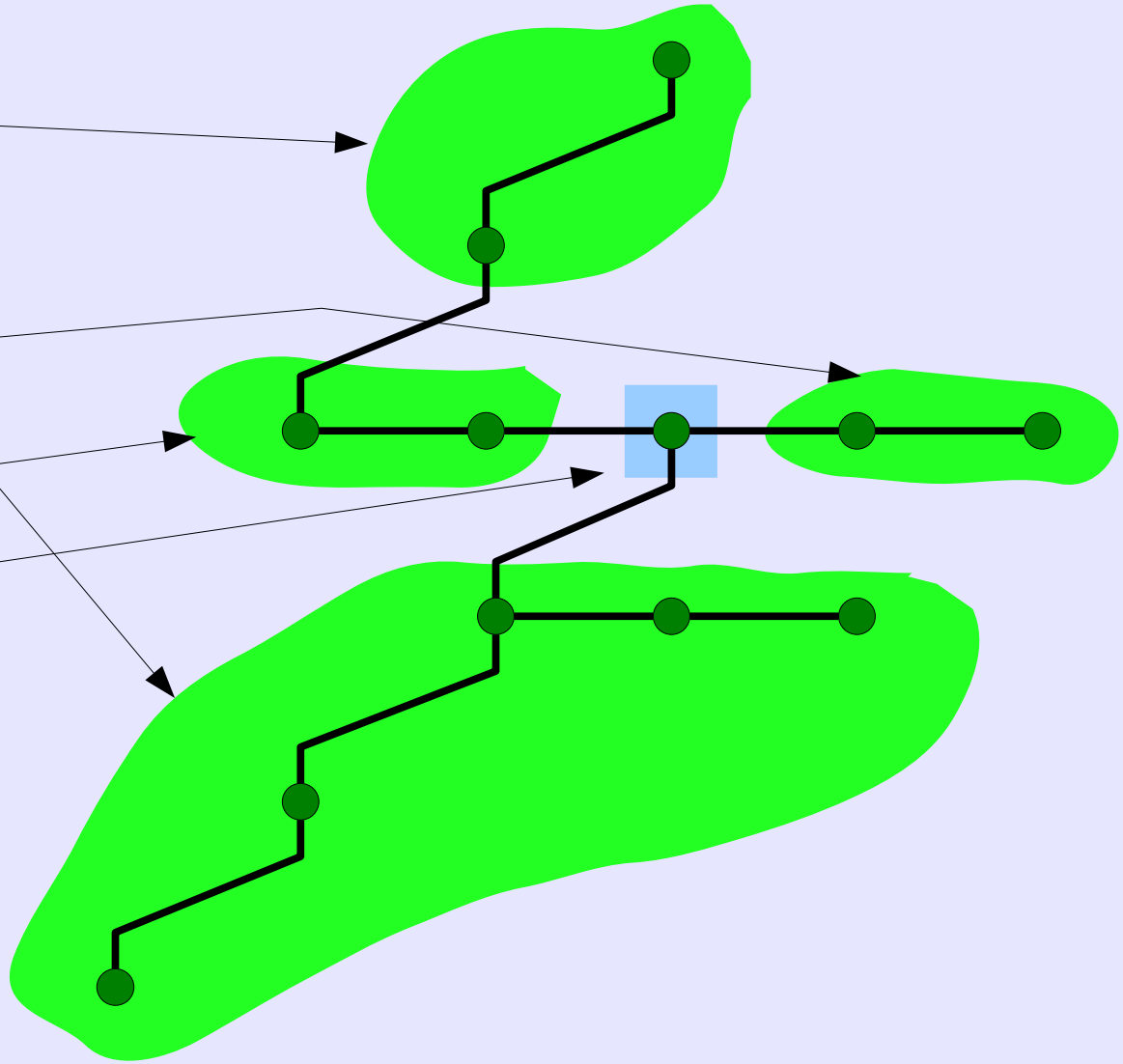
This defines a tree structure, with lateral connections between nodes in the tree.

XPath

- An expression language for navigating XML documents.
- A W3C standard (http://www.w3.org/TR/xpath).
- Models an XML document as a tree of nodes.
- Xholon extensively uses a subset of XPath 1.0
  - To allow nodes to locate other nodes that match certain conditions.
- An expression can return a single node, or a node set.
- An XPath expression is an ordered set of instructions that a monkey could follow to navigate from its current location (the context node) to any other part of the tree.

Common XPath Axes

- ancestor
- descendant
- following
- preceding
- self
- child
- parent
- attribute



XPath expressions in Xholon

ancestor::TheSystem/PatchOwner

ancestor::GPaseSystem/Atp[@roleName='atp1']

ancestor::Grid/../Statistics

ancestor::Elevator/ElevatorPanel/DoorControlButton[@roleName='bOpen']/attribute::port[2]/attribute::replication[1..*]

.[@uid='121219297']/ancestor::StateMachine/descendant::Transition[@uid='968519337']

NetLogo or StarLogo Syntax

- A sequence of logo commands is an ordered set of instructions that a turtle could use to navigate from its current location to any other part of the grid.
- One simple example:

```
forward 10
right 45
backward 5
left 90
```

XPath + XLogo

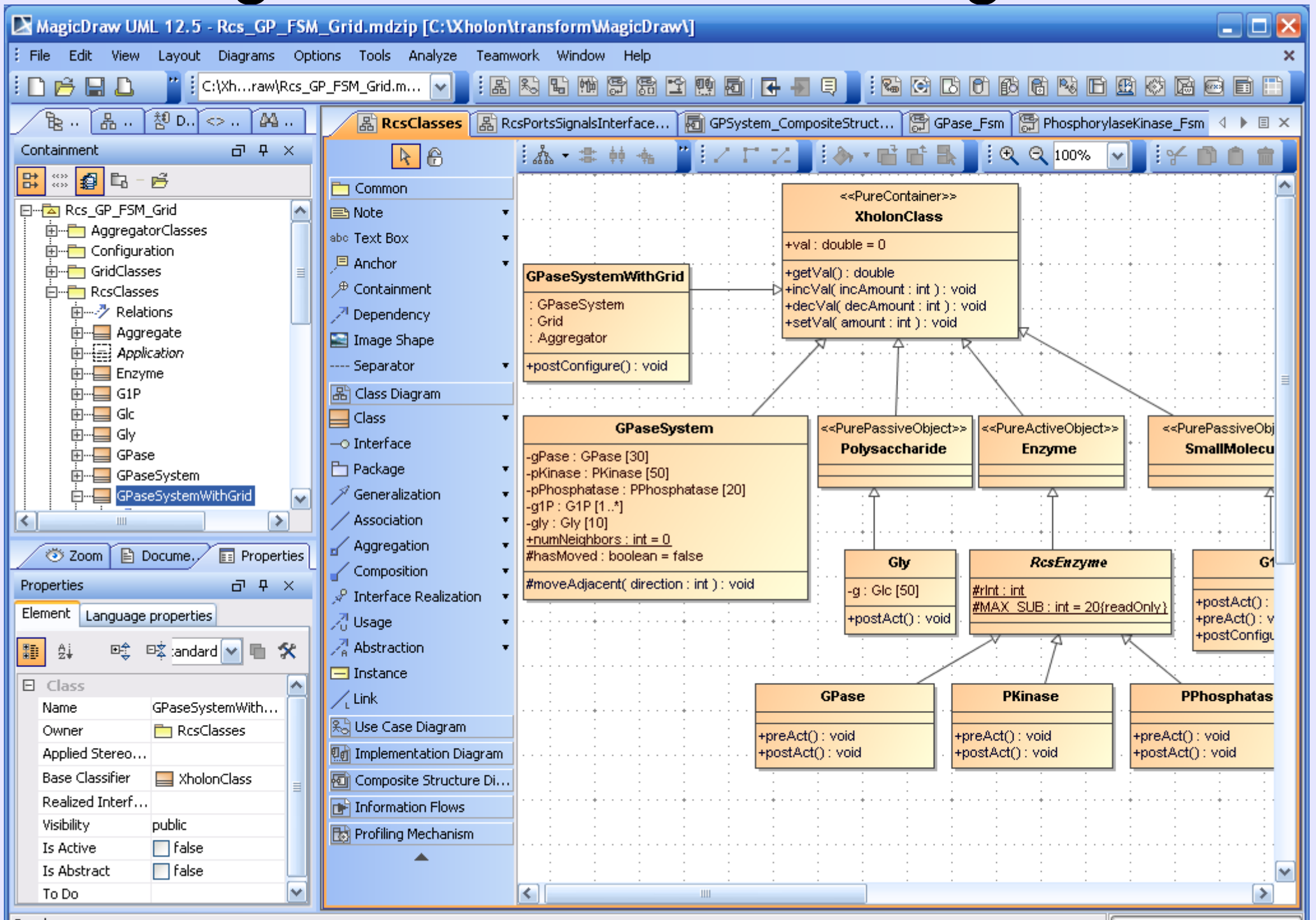
- Xholon combines the XPath and XLogo (NetLogo or StarLogo) syntaxes.
- Turtle Geometry
  - When an agent is moving within a grid, it can use the XLogo-like syntax.
- Monkey Topology
  - When an agent is moving through a tree, it can use the XPath-like syntax.

Standard combined syntax ?

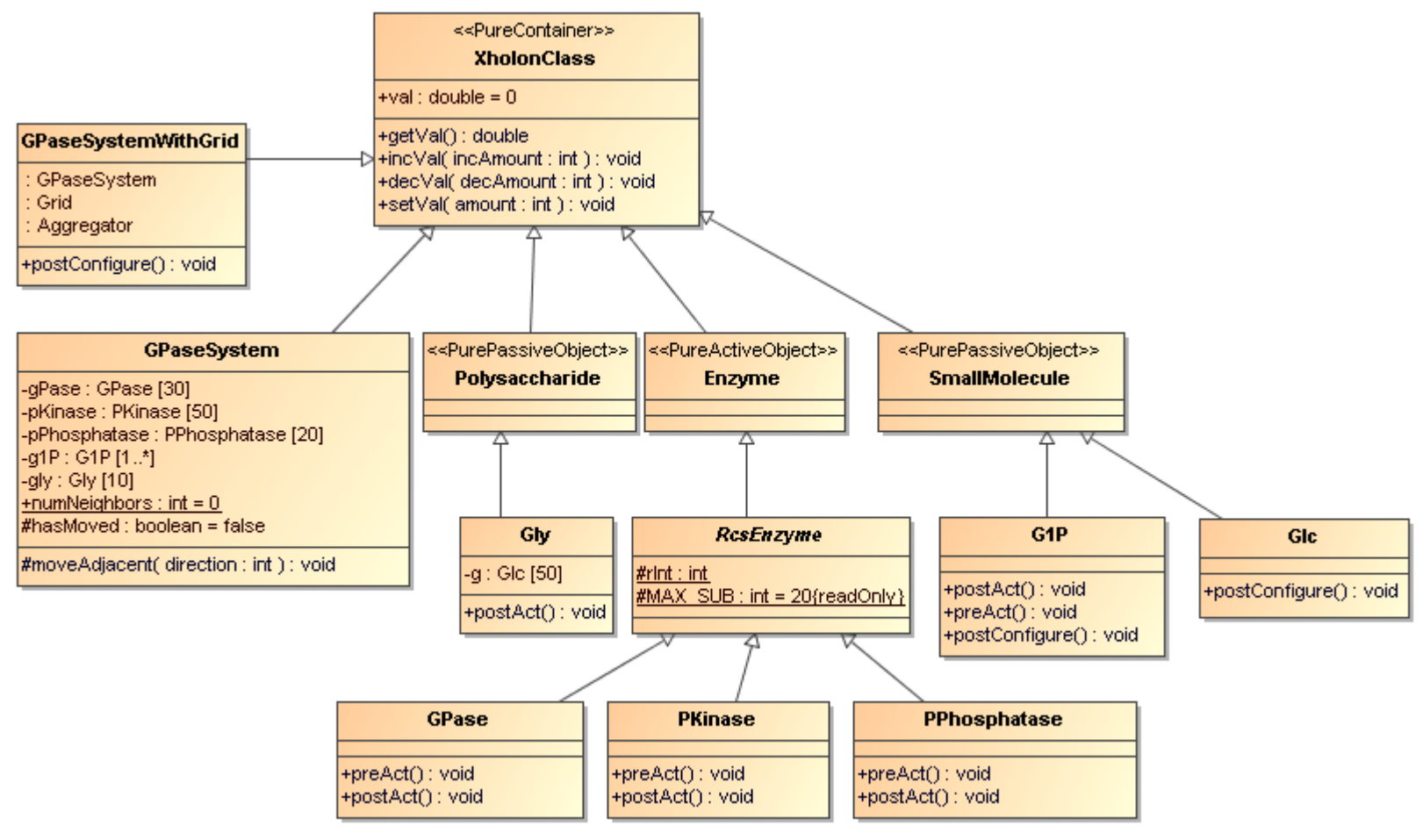
- Could there be a combined standardized syntax for agent navigation through any type of space?
- XPath is already a standardized syntax for navigating tree structures.
- How exactly could the XLogo syntaxes fit into the XPath approach?
- Are there any efforts underway to harmonize the NetLogo and StarLogo syntaxes, and what similar syntaxes are other ABM tools using?
- Could there be a formal ABM syntax?

Demo of Xholon

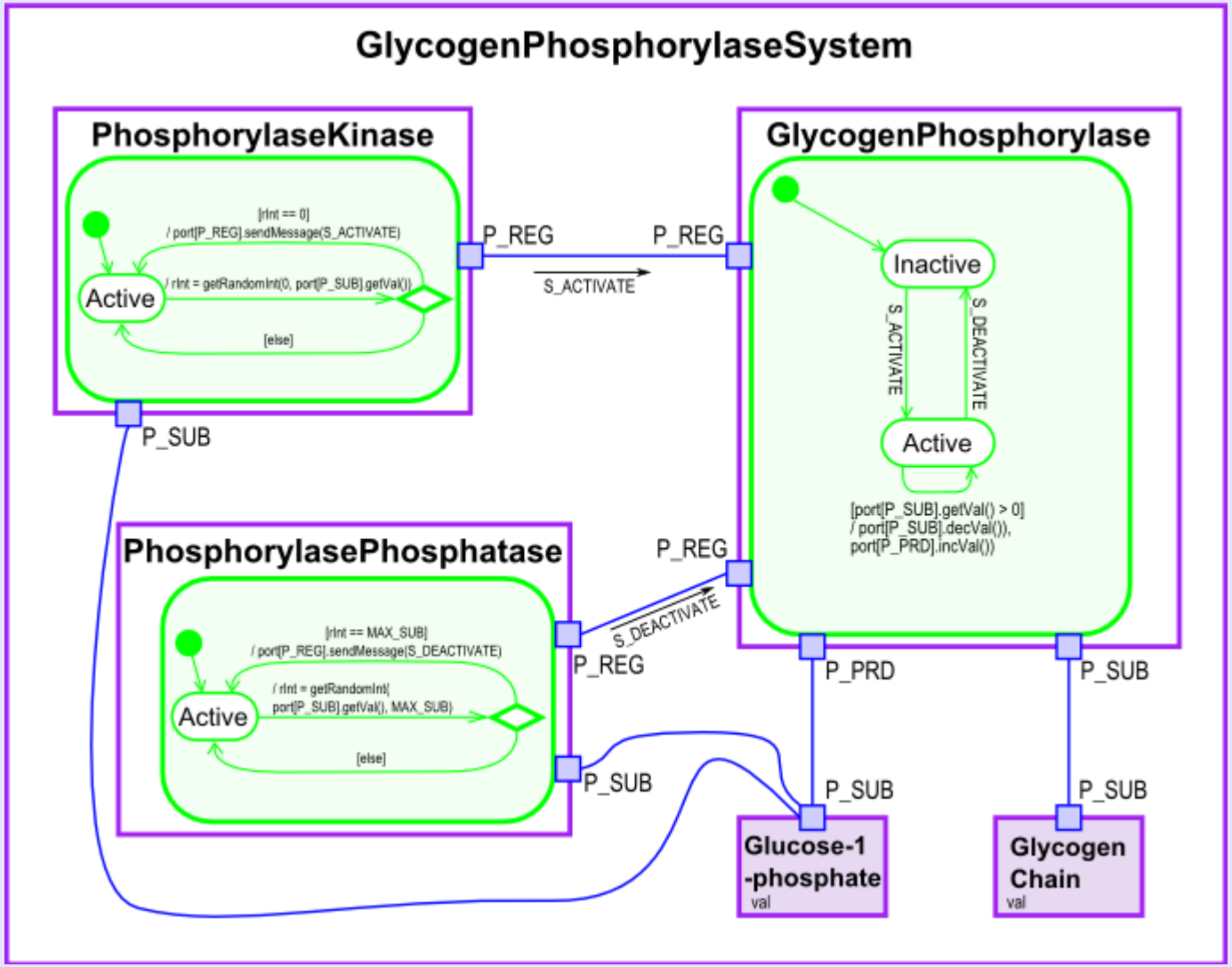
MagicDraw UML Modeling Tool



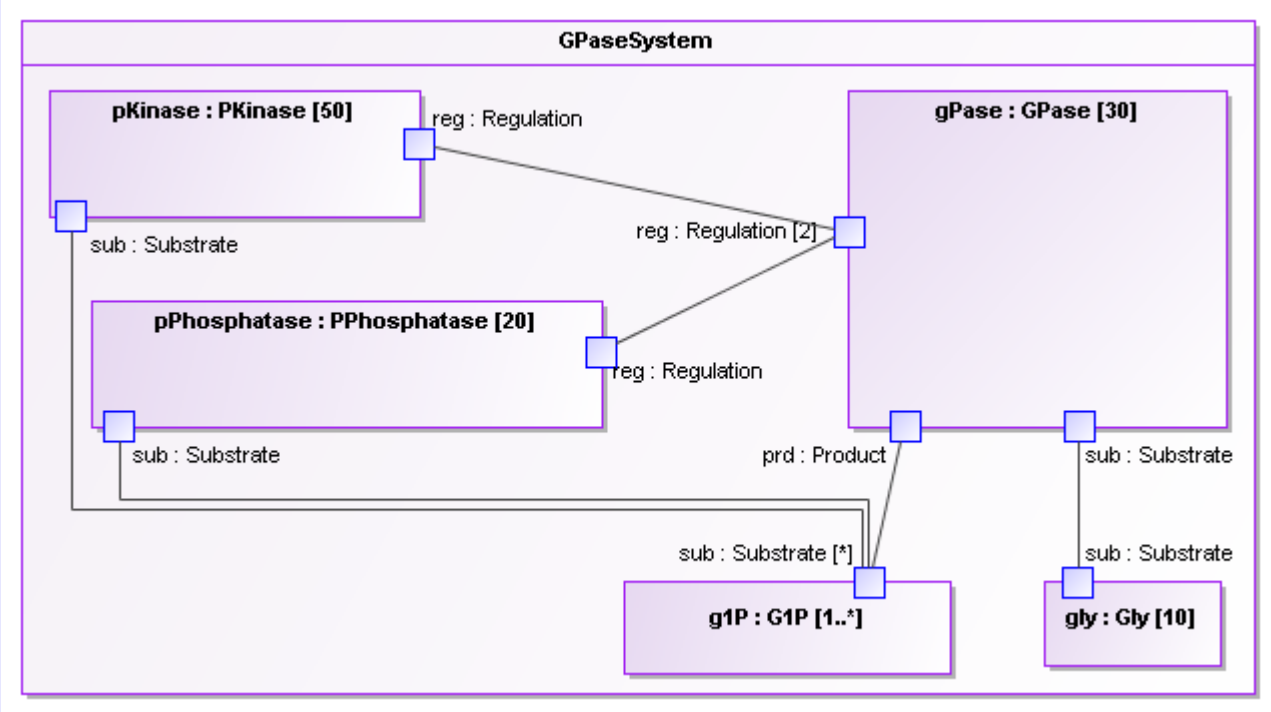
RCS – UML - Agent Classes



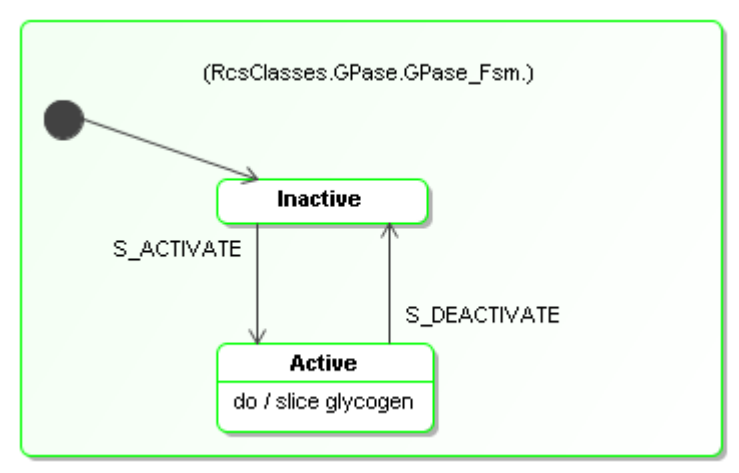
RCS – Agent Architecture



RCS – UML - Composite Structure

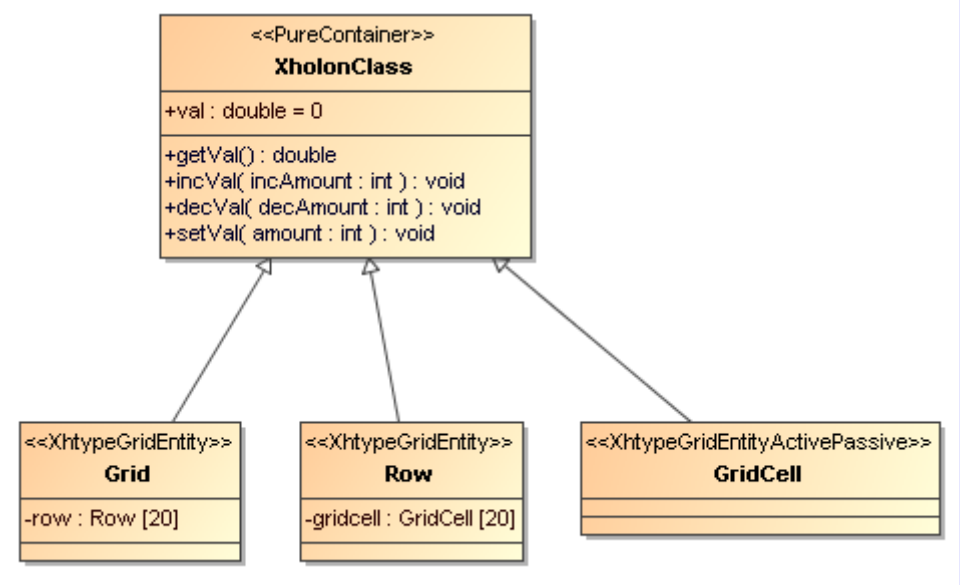


RCS – UML - State Machine

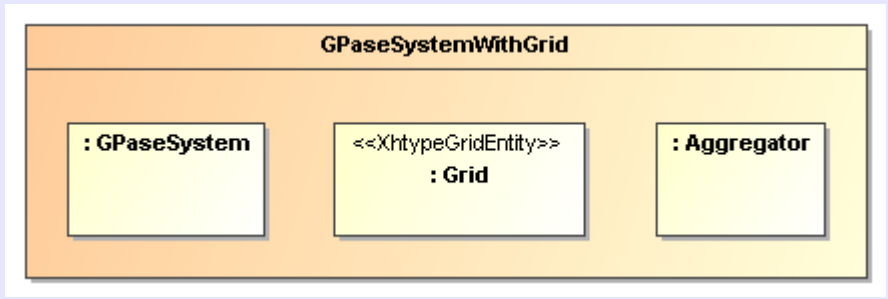


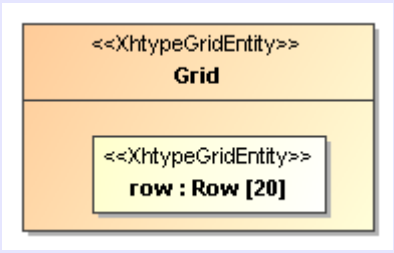
RCS – UML - Grid Classes

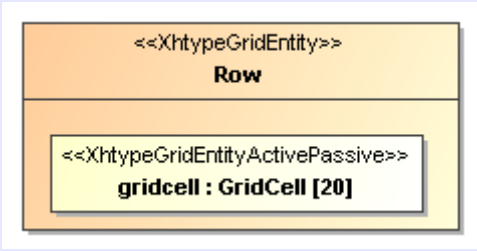
Class inheritance hierarchy



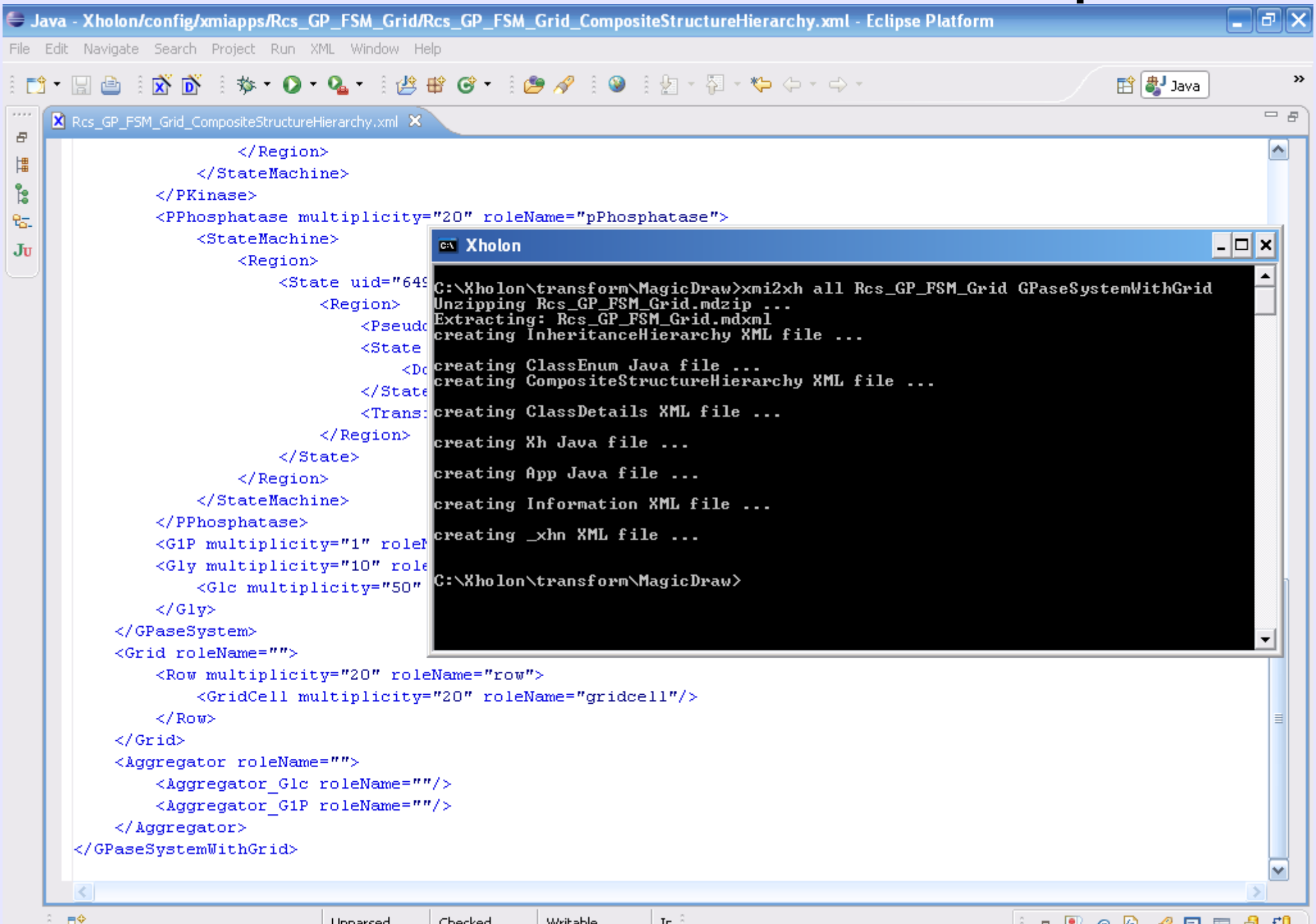
Composite Structure



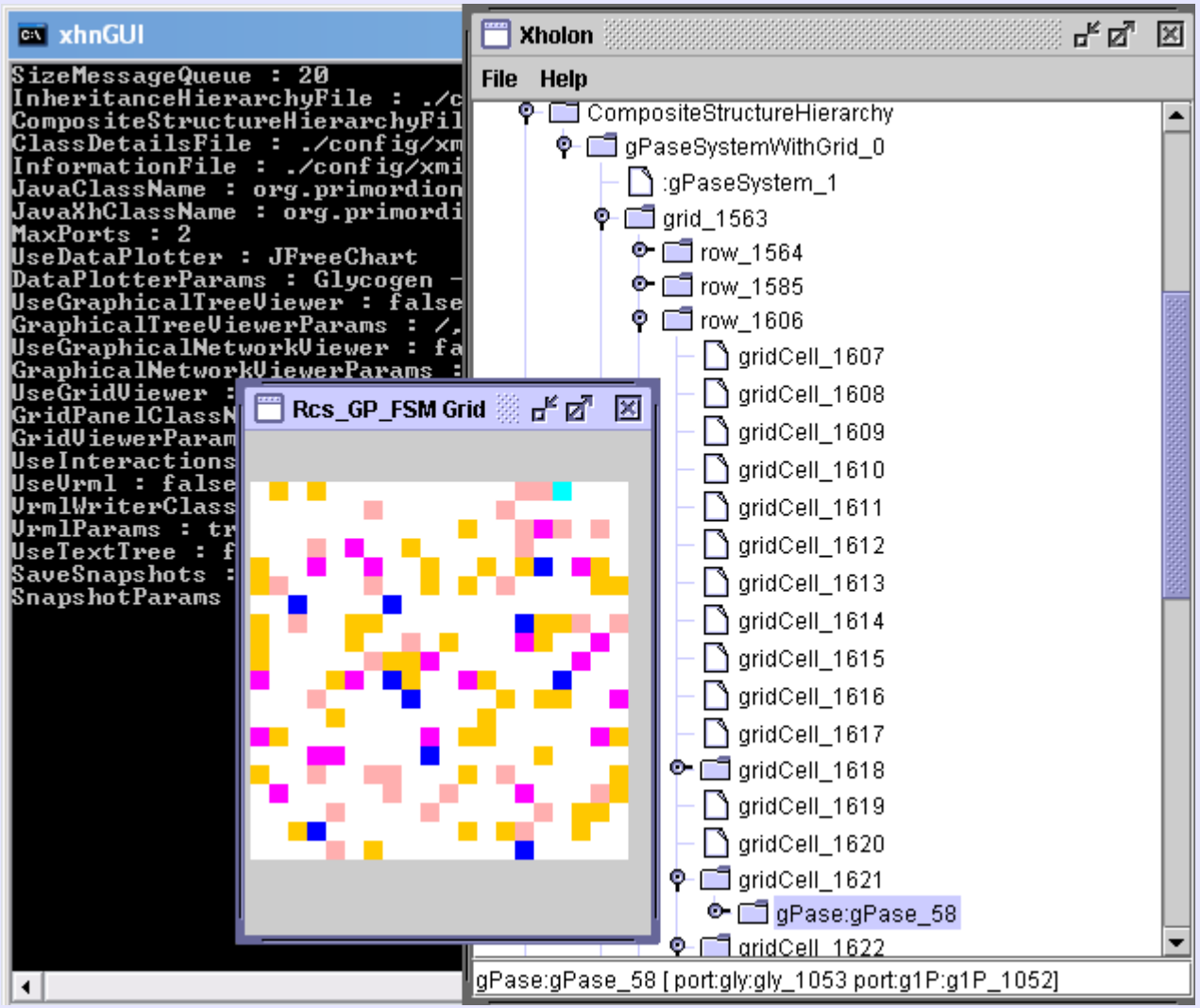




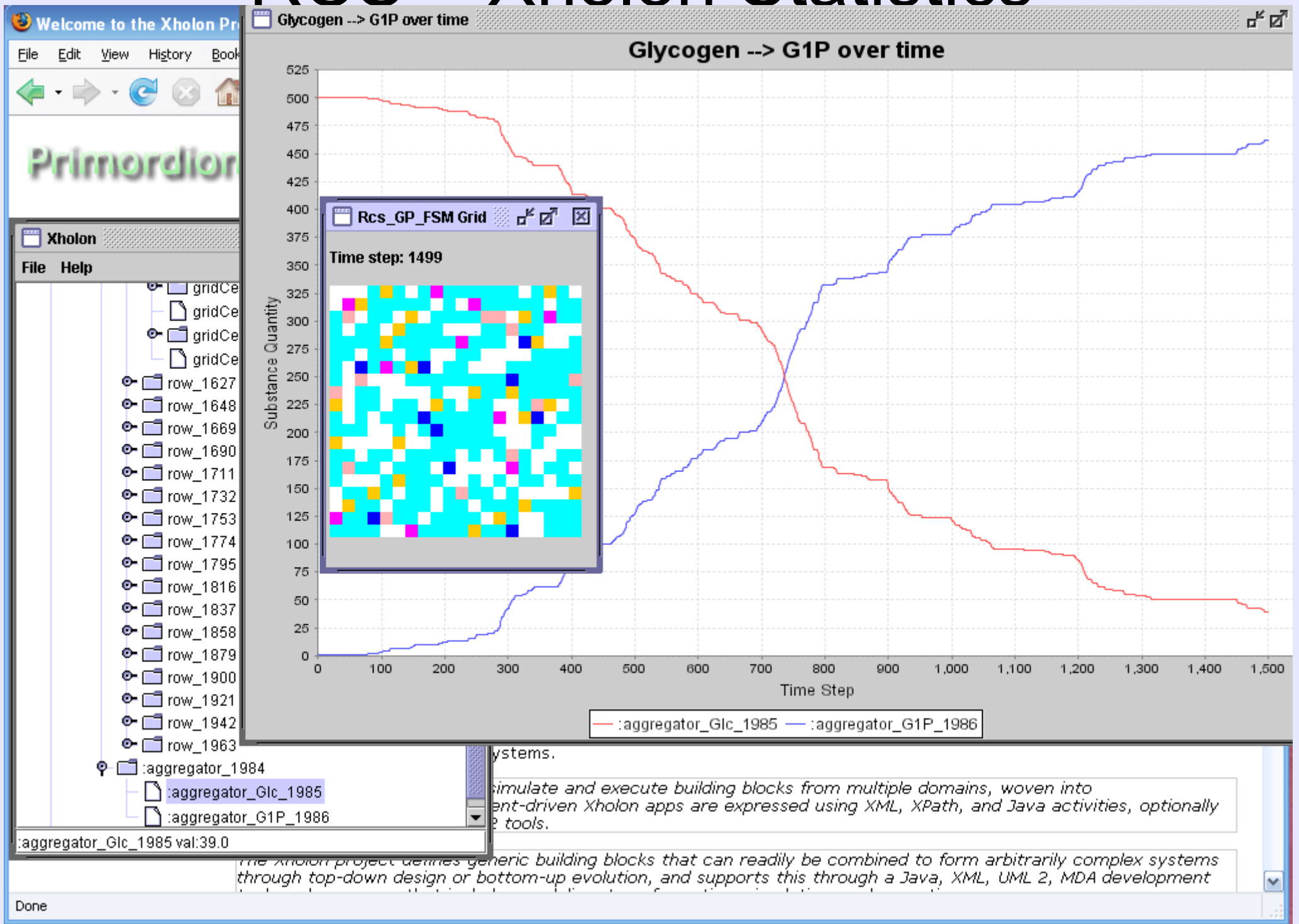
Model Driven Software Development



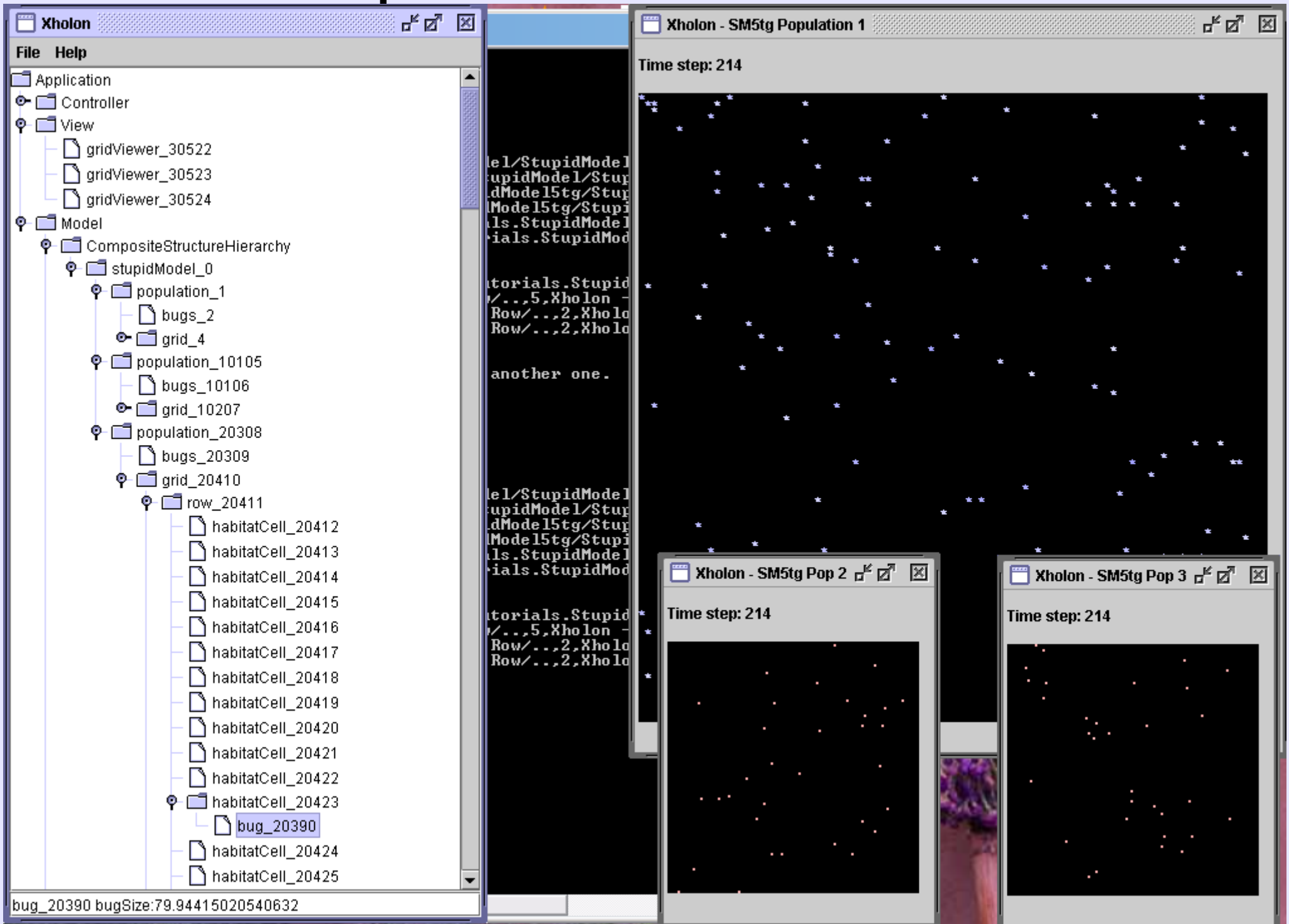
RCS – Execution in Xholon



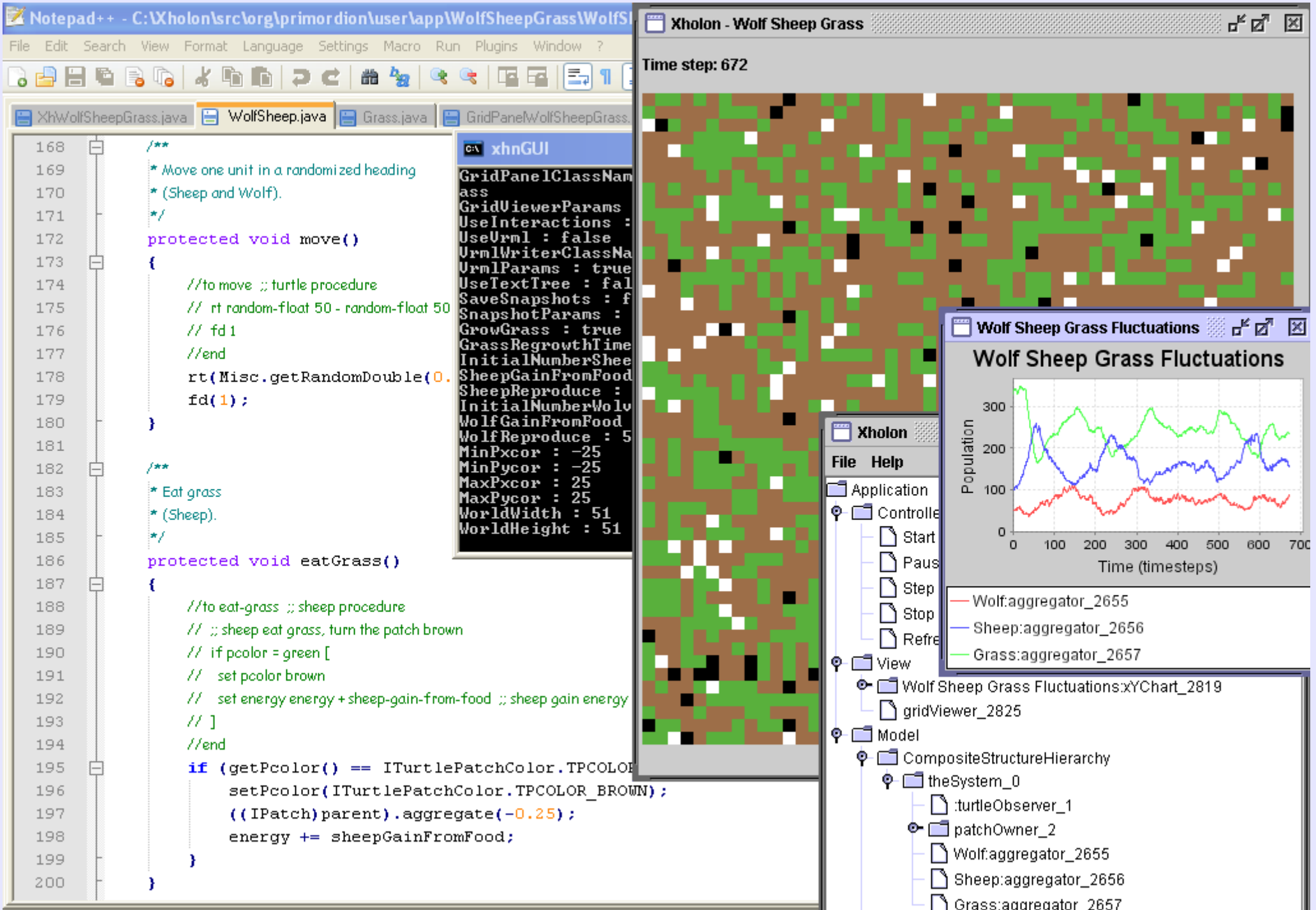
RCS – Xholon Statistics



Stupid Model – 3 Grids



Wolf, Sheep, Grass originally in NetLogo, also in Repast



NetLogo-like Syntax in Xholon

```
/**
     * Eat grass
     * (Sheep).
     */
    protected void eatGrass()
    {
        //to eat-grass  ;; sheep procedure                                    NetLogo
        //  ;; sheep eat grass, turn the patch brown
        //  if pcolor = green [
        //    set pcolor brown
        //    set energy energy + sheep-gain-from-food  ;; sheep gain energy by eating
        //  ]
        //end
        if (getPcolor() == ITurtlePatchColor.TPCOLOR_GREEN) {  // Xholon Java
            setPcolor(ITurtlePatchColor.TPCOLOR_BROWN);
            ((IPatch)parent).aggregate(-0.25);
            energy += sheepGainFromFood;
        }
    }
```

Questions ?

http://www.primordion.com/Xholon/

Contact Ken: ken@primordion.com