

Open-Source from the Trenches

Ken Webb

Open Source Software Seminar
Carleton University, School of Computer Science
March 16, 2007

ken@primordion.com

Background

- ◆ ObjecTime, IBM Rational Rose RealTime commercial products.
- ◆ Published several papers with Tony White.
- ◆ 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 UML 2.0, 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.

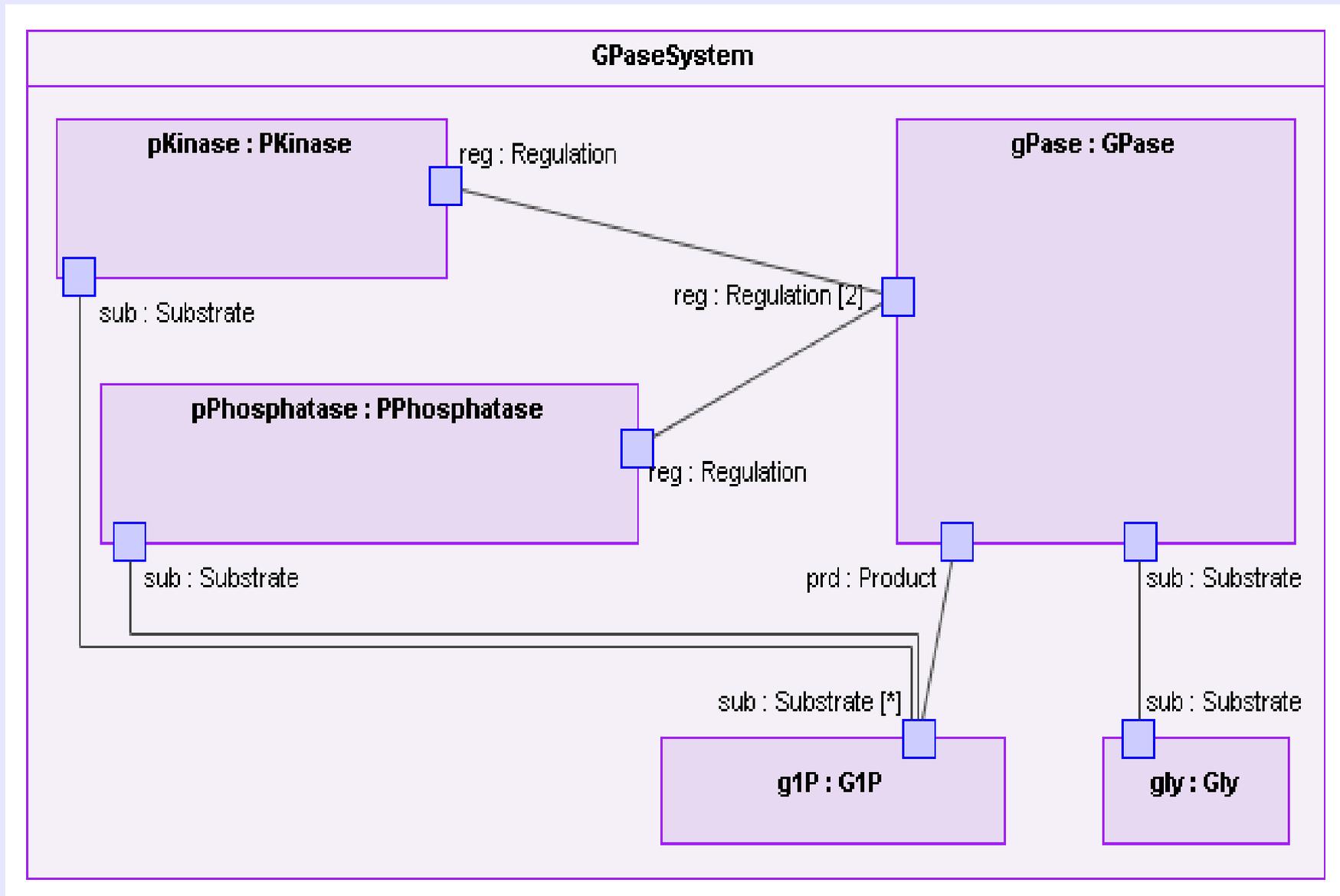
Examples – Why trees are important

- 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.

Xholon Modeling Constructs

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

Composite Structure, Ports



Use of Open Source

- ◆ Xholon depends on lots of other products, mostly open-source.
 - ◆ XPP3 for reading in XML configuration files.
 - ◆ JFreeChart and gnuplot for line charts.
 - ◆ JUNG for tree and network graphs.
 - ◆ ECJ for genetic programming.
 - ◆ UMLGraph, plotutils for runtime sequence diagrams.
 - ◆ libsbml to import biology simulations.
 - ◆ xalan-j for transformation from UML to Xholon
 - ◆ MagicDraw (commercial) for UML 2.0 modeling.
 - ◆ Most of these include other open-source packages.

Use of Open Standards

- ★ As important as open source
 - ★ Sun's Java is now open source
 - ★ XML is a family of useful standards
 - ★ UML is a great starting point for modeling
 - ★ XPath-based tree navigation is central to Xholon
 - ★ Systems Biology Markup Language (SBML)
 - ★ XSLT
 - ★ PNG

Demo of Xholon

- Cell, Life models (composite structure)
- Elevator controller (state machines, UML)
- Brusselator (SBML, Math Integration)
- Two very recent enhancements:
 - Turtle geometry, NetLogo-like
 - Observer Pattern

Some practical issues

- Bloat, from the large number of excellent open-source packages available that can/could be used in conjunction with Xholon.
 - One modeling package I recently downloaded needed 500MB to run, because of all the open-source packages it uses.
- Time consuming to keep on top of possibly useful open-source packages; evaluation
- Packaging, dependencies
 - How to provide my users with all the other packages.

Some Practical Benefits of Open Source

- ★ Provides a rich assortment of capabilities that can be brought together in new ways.
- ★ Legitimizes non-commercial software development.
- ★ Provides a legal framework; GPL, other licenses.

Conclusions

- I'm happy to help anyone get started using Xholon in a research project.
- There are lots of aspects of the Xholon project that need help from contributors.

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

Contact Ken: ken@primordion.com