Xholon

Ken Webb

Bar Camp - Ottawa
Carleton University
March 31, 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.

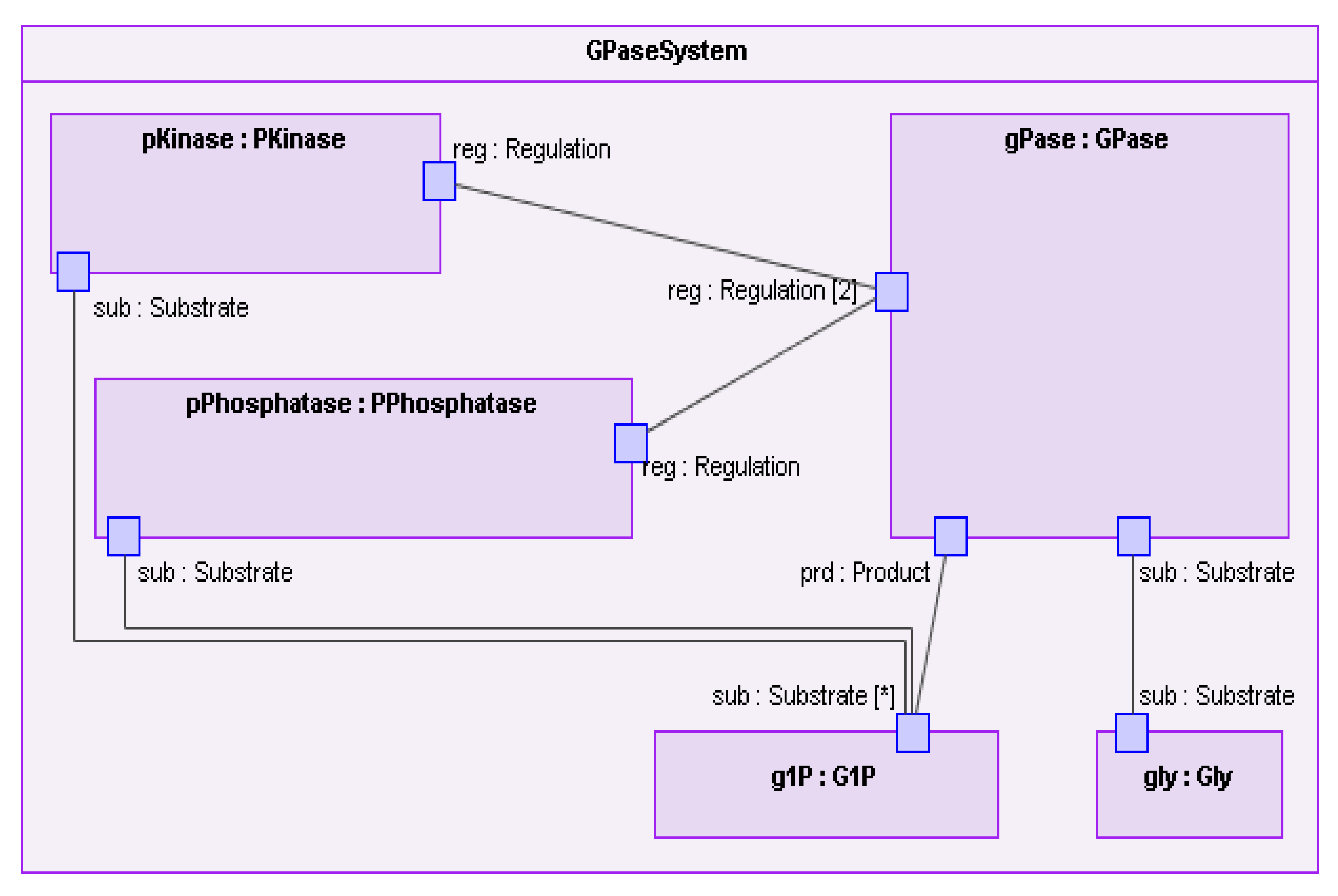
Examples – Why trees are important

- 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



Reformulation

- I'm not doing a good job of explaining exactly what Xholon is, how it differs from other similar concepts, and what its benefits are.
- I'm trying to come up with a better way of describing what Xholon is.
- The next few slides present an alternative explanation that I'm working on.

Summary of My Reformulation

- Spreadsheets exhibit a certain set of characteristics.
- Many other systems also have these.
- Xholon generalizes these and allows building blocks from multiple domains to work together.
- Biological systems exhibit the same characteristics, in a very robust way.

Spreadsheets

- A spreadsheet or worksheet is made up of cells.
- Cells are organized into a rectangular table or grid.
- A cell (a universal building block for this domain) can contain a value or a formula.
  - Many types of values and formulas are possible.
- A formula cell contains references to other cells.

NetLogo

- A model is made up of patches and turtles.
- These are organized into a rectangular grid.
- Patches are passive, and turtles are active.
  - Many types of patches and turtles are possible.
- An active turtle references other patches and turtles.

Xholon generalizes these ideas

- A model is made up of interacting building blocks.
- Building blocks are organized into a composite structure tree, overlaid by networks including tables, grids, and any type of graph.
- A building block can be active, passive, a container, or a combination.
  - Many domain-specific types are possible.
- An active building block contains port-based connections to other building blocks.

Biology

- Biology already generalizes these concepts.
    - A cell is made up of compartments and chemicals.
    - Organisms, cells, compartments and chemicals are organized into a composite structure that covers the entire planet, with numerous cross-connections creating many different networks.
    - Entities can be active, passive, containers are combinations of these.
        - Huge numbers of different types exist.
    - Entities have binding sites that let them connect to other entities.
    - And much more.

UML 2 Composite Structure, OO

- A system is made up of objects.
- Objects are organized into containment hierarchies.
- An object can be a container, and/or an active or passive object.
  - Many object subclasses are possible.
- Objects can point to or refer to each other, allowing interactions at runtime.

Universal Building Blocks

- Each domain has one or more universal building blocks, that can be specialized to create the objects of that domain.
- Xholon knows how to make universal building blocks from multiple domains work together.
- In biology, chemicals, cells and organisms are universal building blocks, each at a different level.

Xholon = core + plugins + GUIs

- PLUS, ability for building blocks in different plugins to interact at runtime.

Examples

- A swarm-based simulation in which agents contain state machines.
- Spreadsheets in a model, to record and act on values produced by agents in other parts of the model.
-

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

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