Modelling the Reactive Behaviour of SVG-based User Interfaces with Hierarchically-linked Statecharts

McGill
School of Computer Science

Universiteit Antwerpen
Department of Mathematics and Computer Science

MSDL
Modelling, Simulation and Design Lab

Jacob Beard, Hans Vangheluwe

October 2nd, 2009

SVG Open 2009

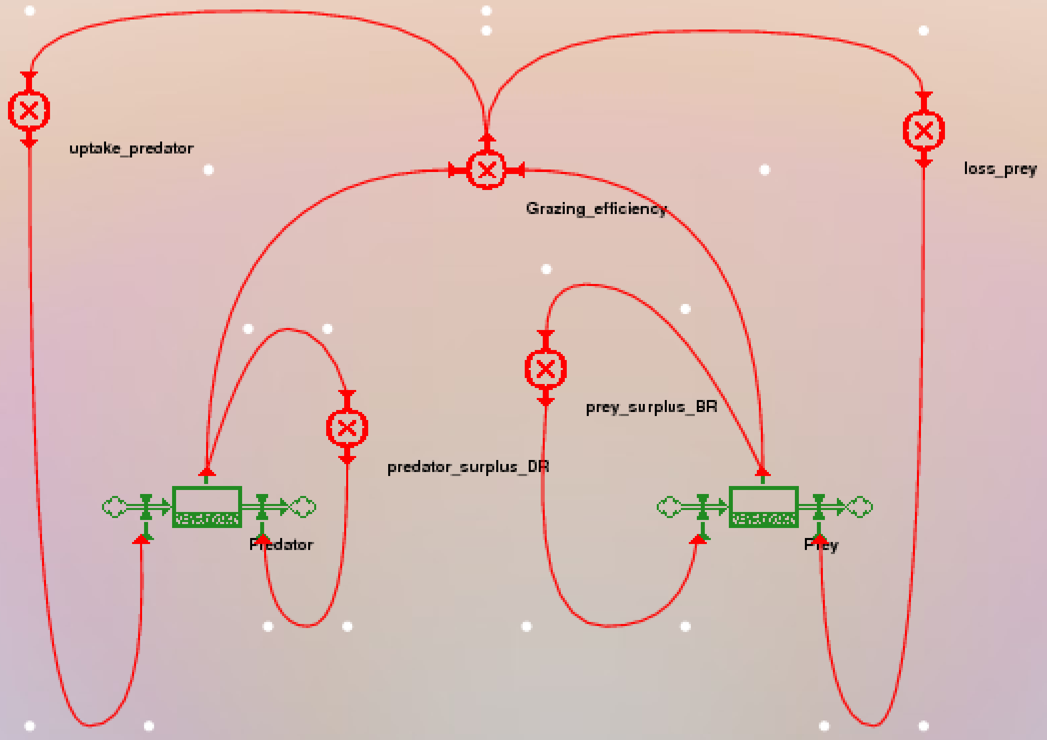
Presentation Structure

- Goals
- Previous Work
- Developing with Hierarchically-linked Statecharts
- Future work

Our Goals

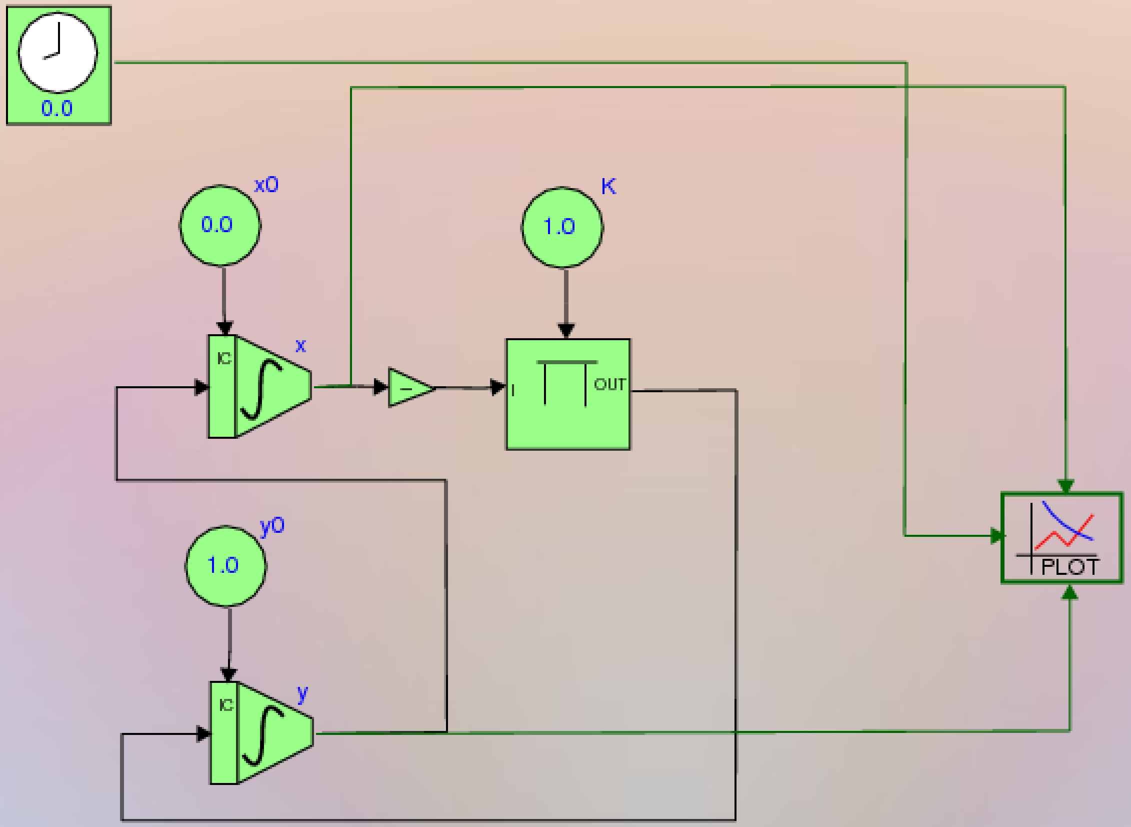
- Ease development of structurally and behaviourally complex UI's.
- UI development is hard.
- Minimize accidental complexity
- *Model everything*:
  - Faster to develop
  - More maintainable
  - Greater reliability

Forrester System Dynamics model of Predator-Prey

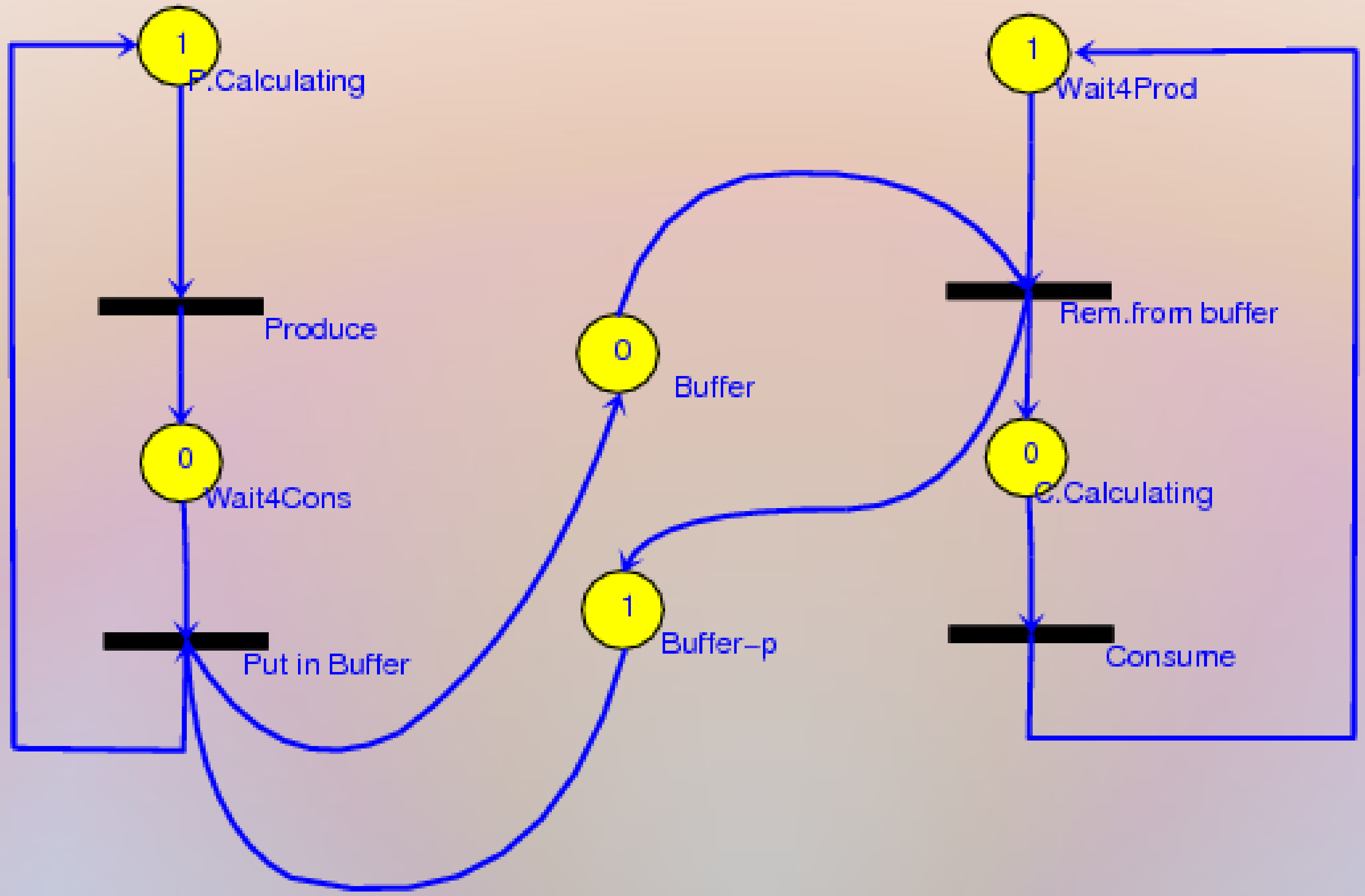


2-species predator-prey system

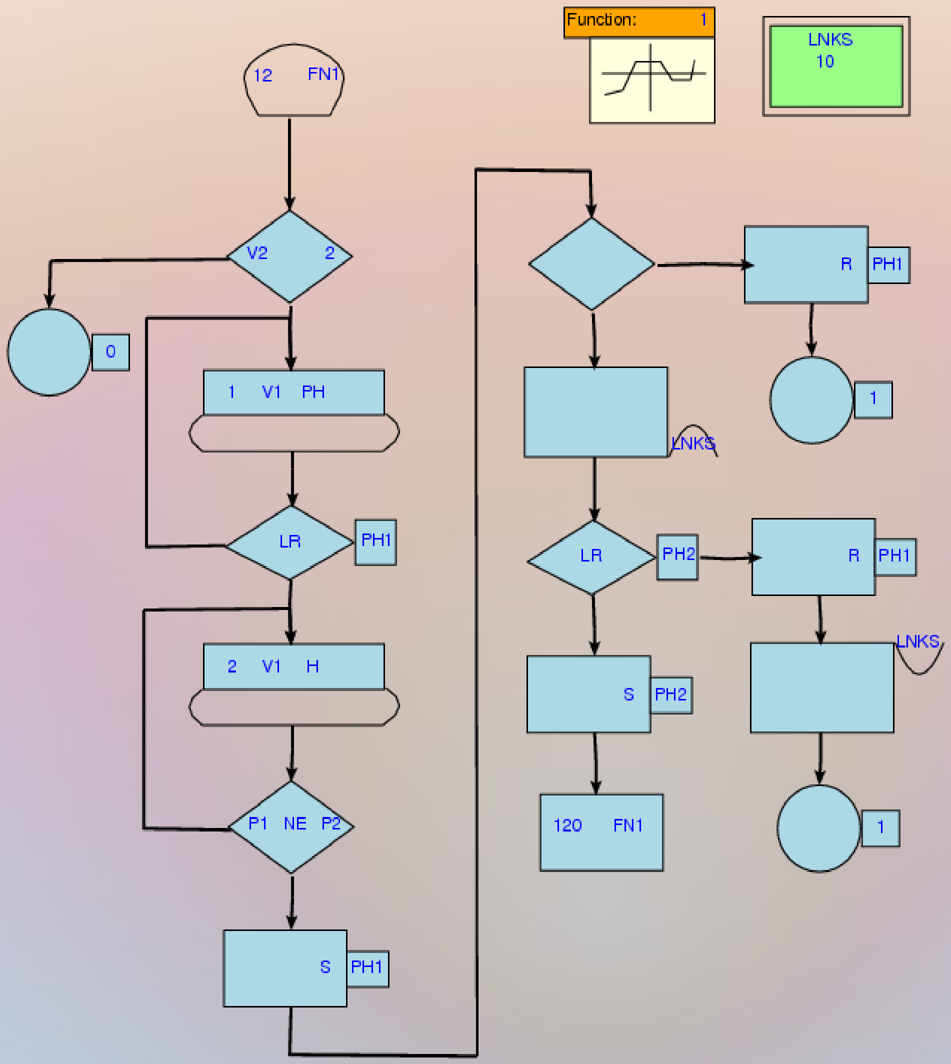
Causal Block Diagram model of Harmonic Oscillator



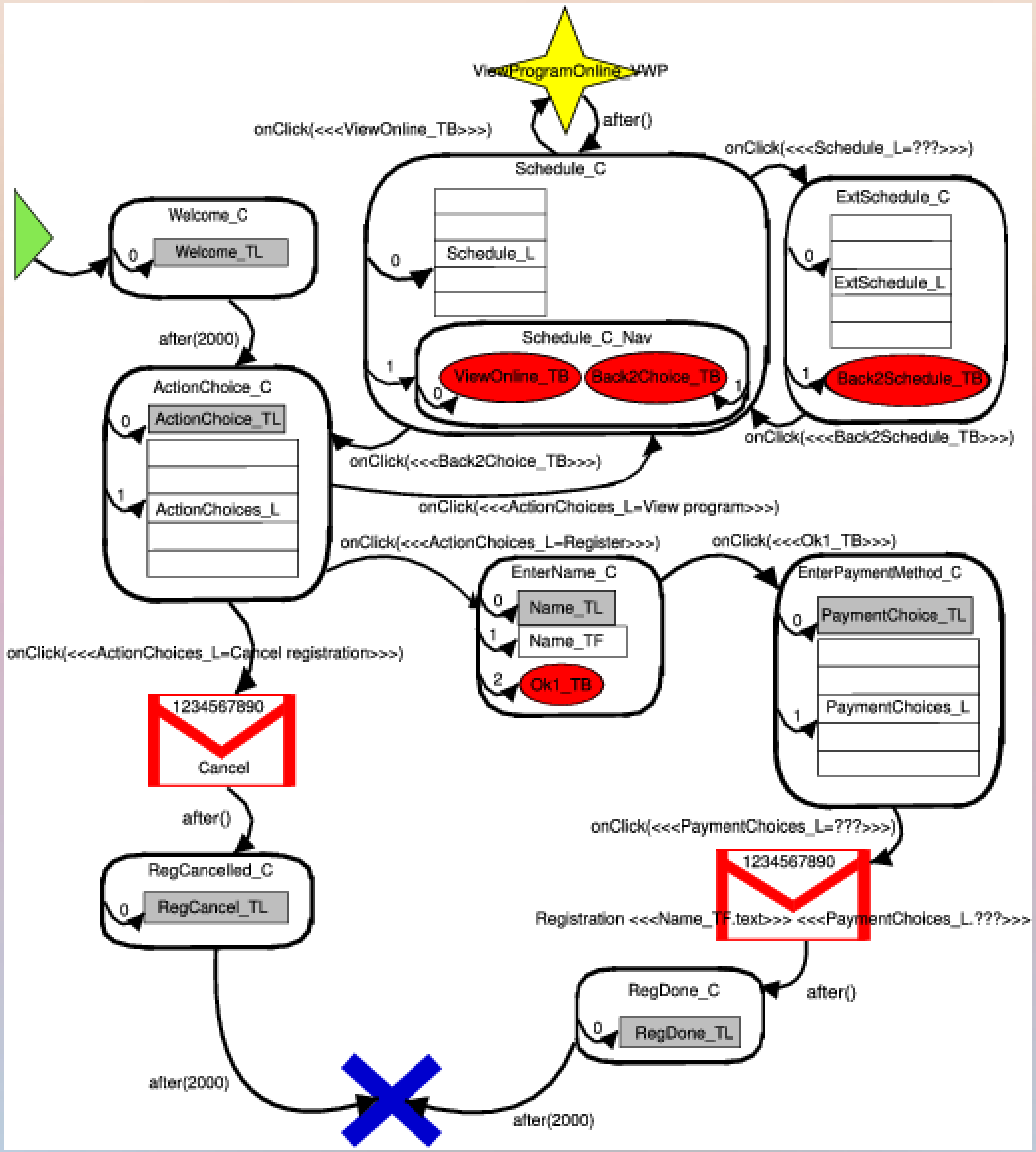
Petri Net model of Producer Consumer



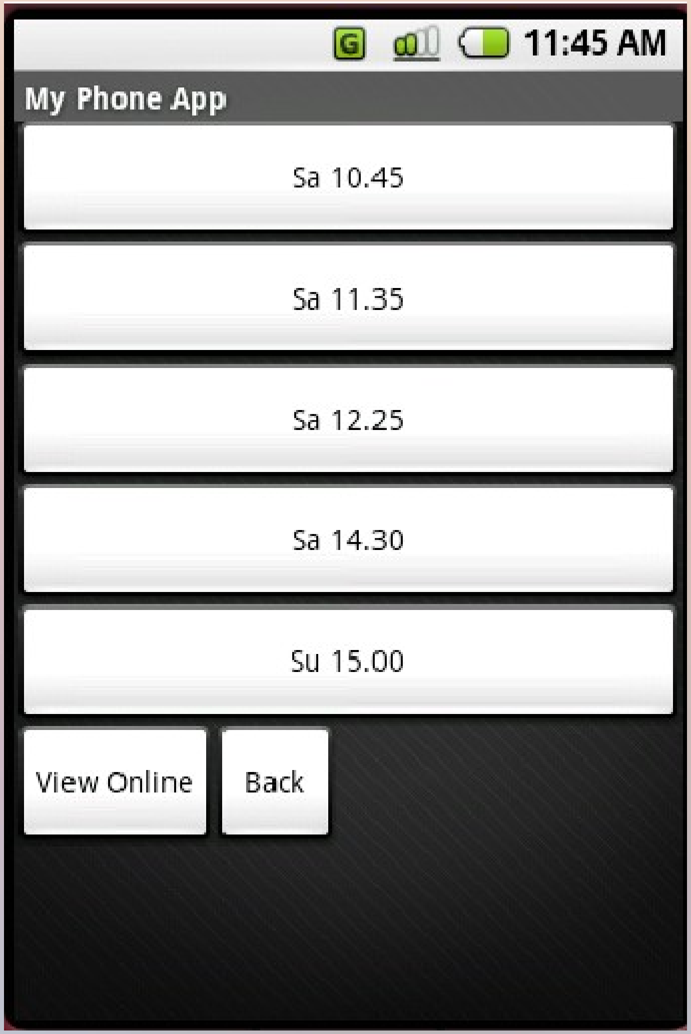
GPSS model of Telephone Exchange



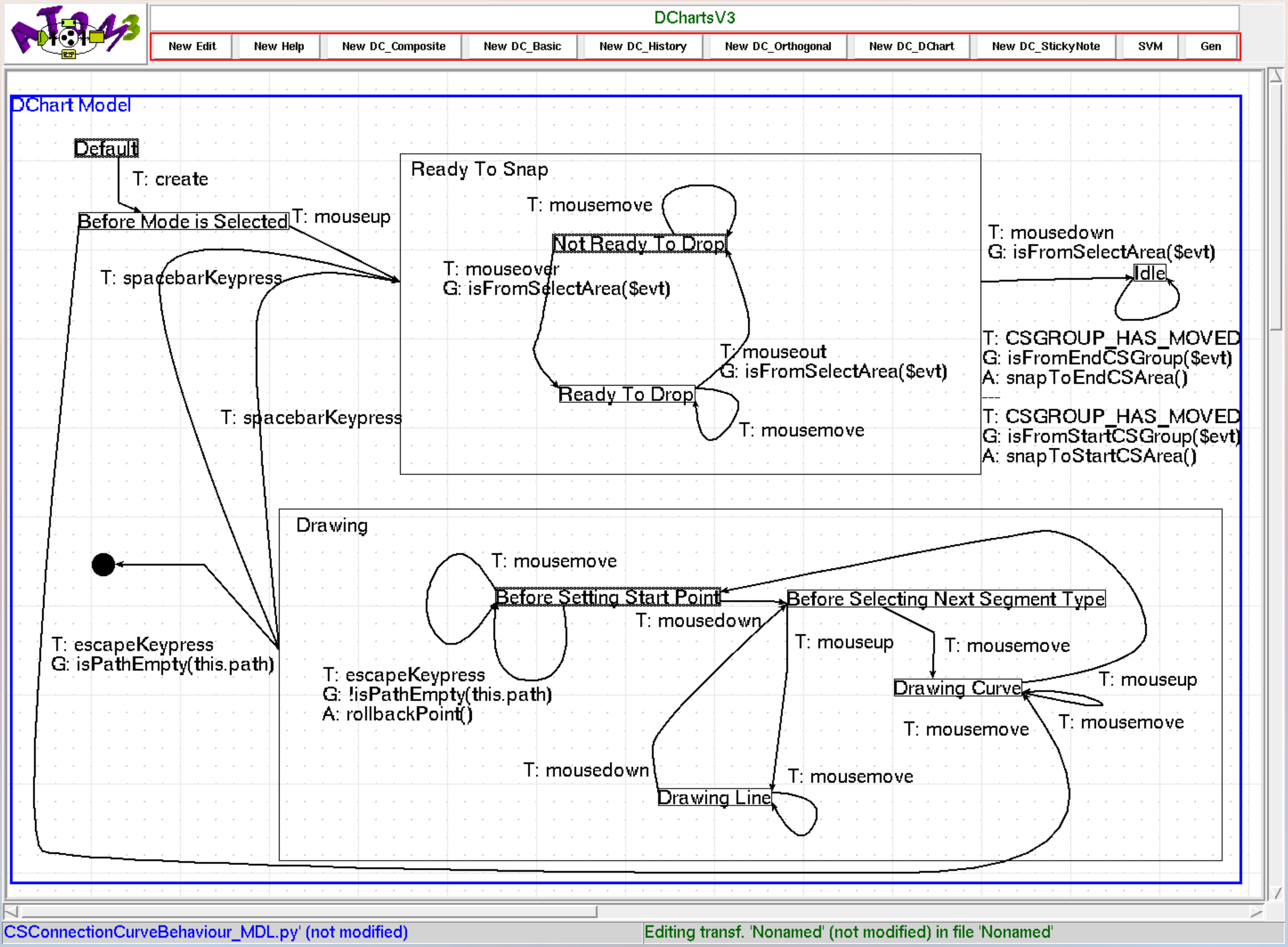
DS(V)M example application, the PhoneApps Domain-Specific model



DS(V)M example application: conference registration (Google Android)



AToM3



CSConnectionCurveBehaviour_MDL.py' (not modified)

Editing transf. 'Nonamed' (not modified) in file 'Nonamed'

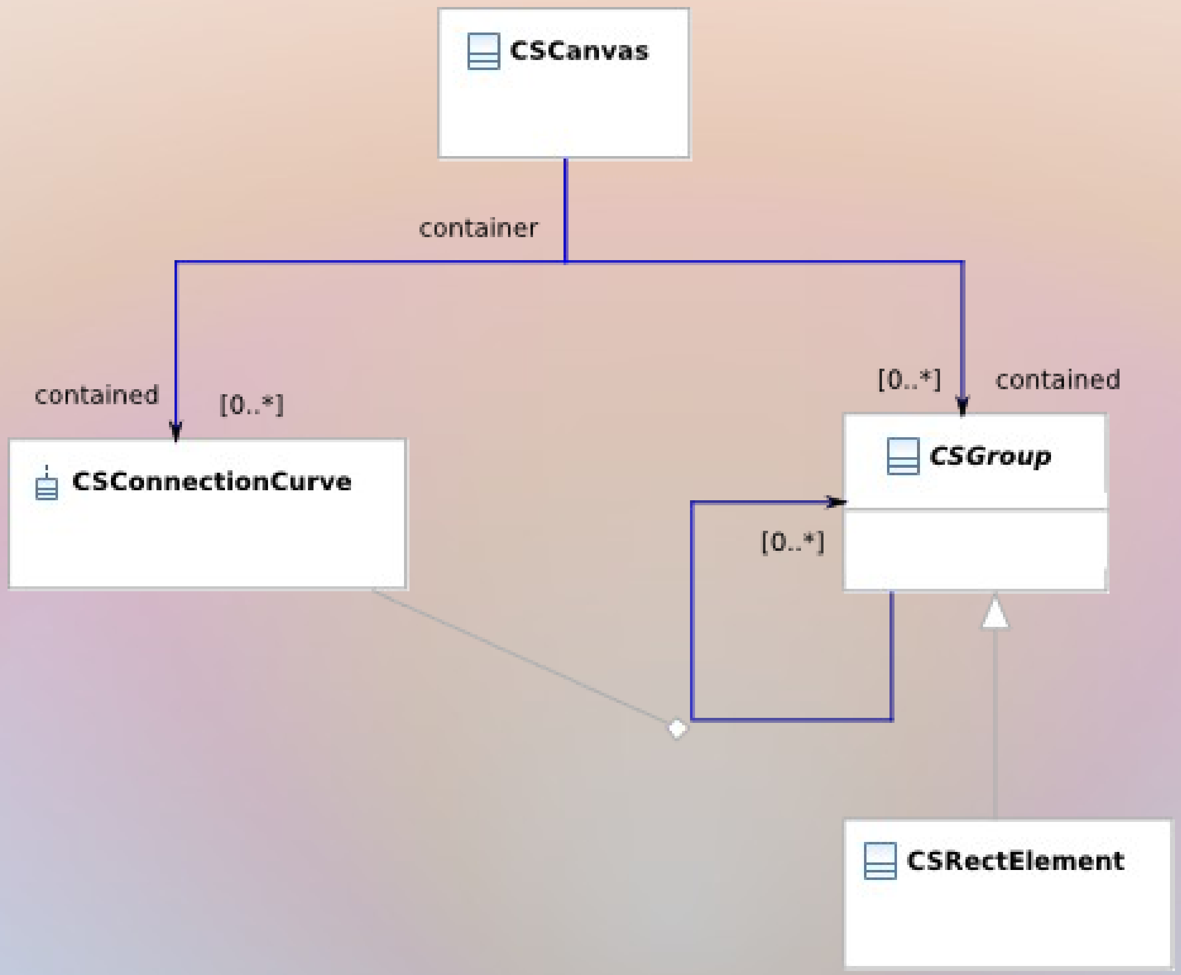
Developing with HlS

- *Model every aspect of the system-to-be-built, at the most appropriate level of abstraction, using the most appropriate formalism(s).*
- Class Diagrams and Statecharts to model UI

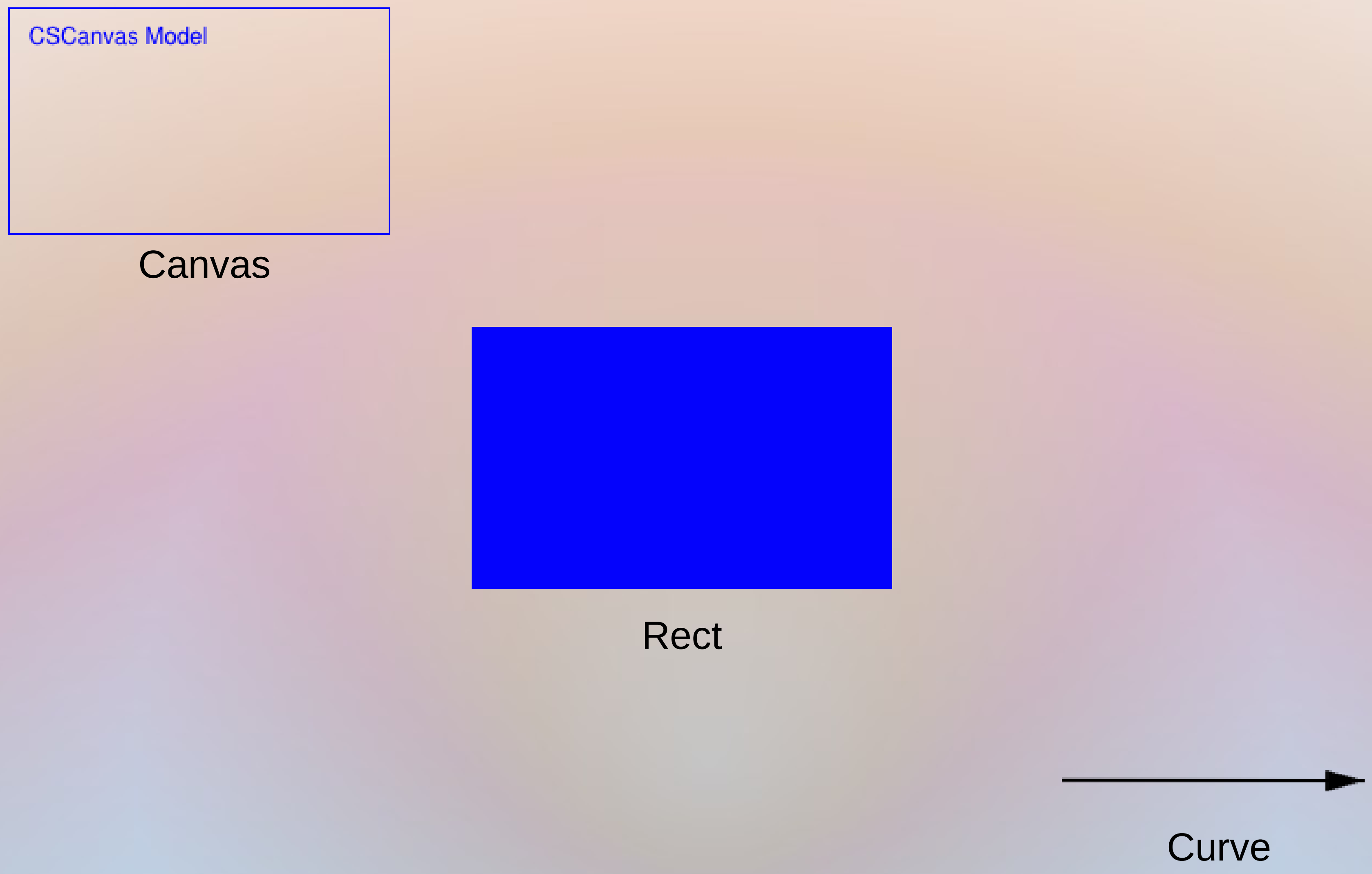
HlS Workflow

- Based on language engineering.
- Model
  - Abstract Syntax
  - Concrete Syntax
  - Behaviour
- Compile

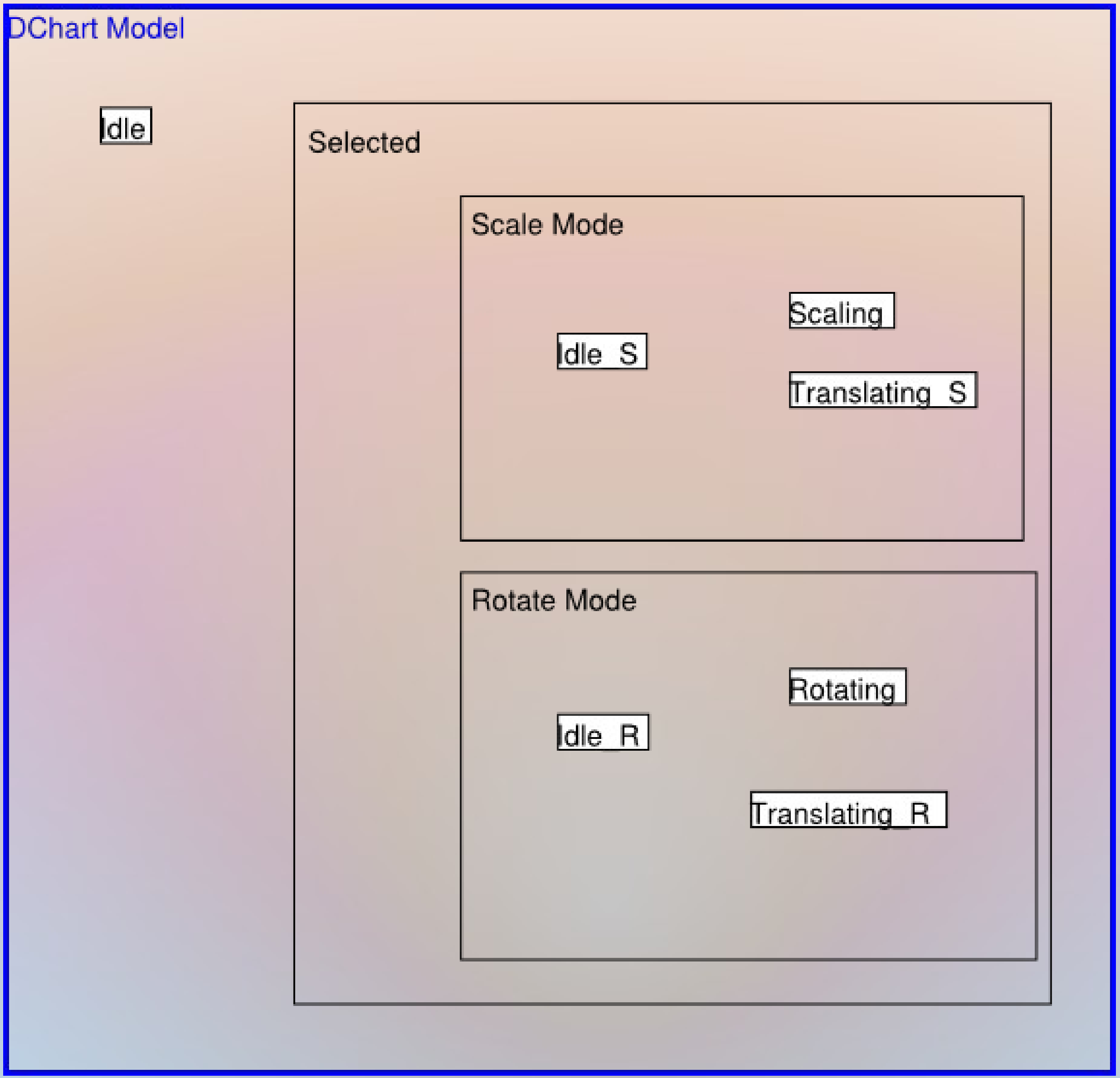
Inkscape Example (Abstract Syntax)



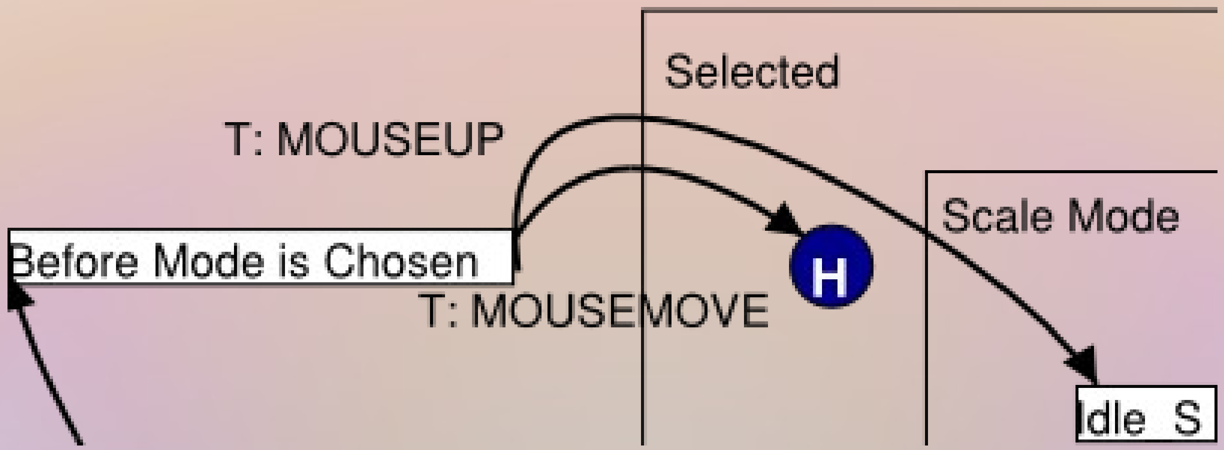
Inkscape Example (Concrete Syntax)



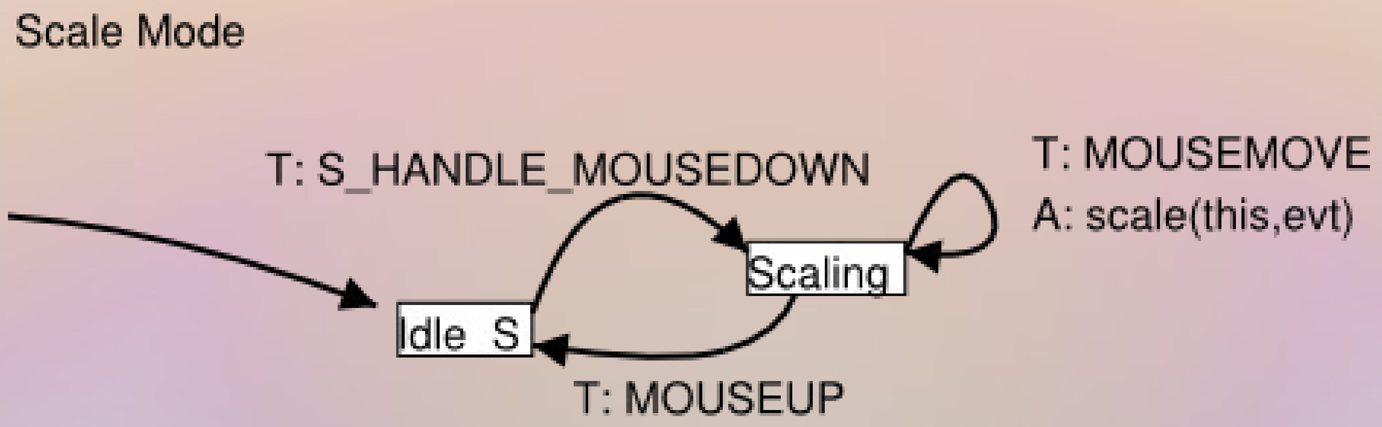
Inkscape Example (Behaviour)



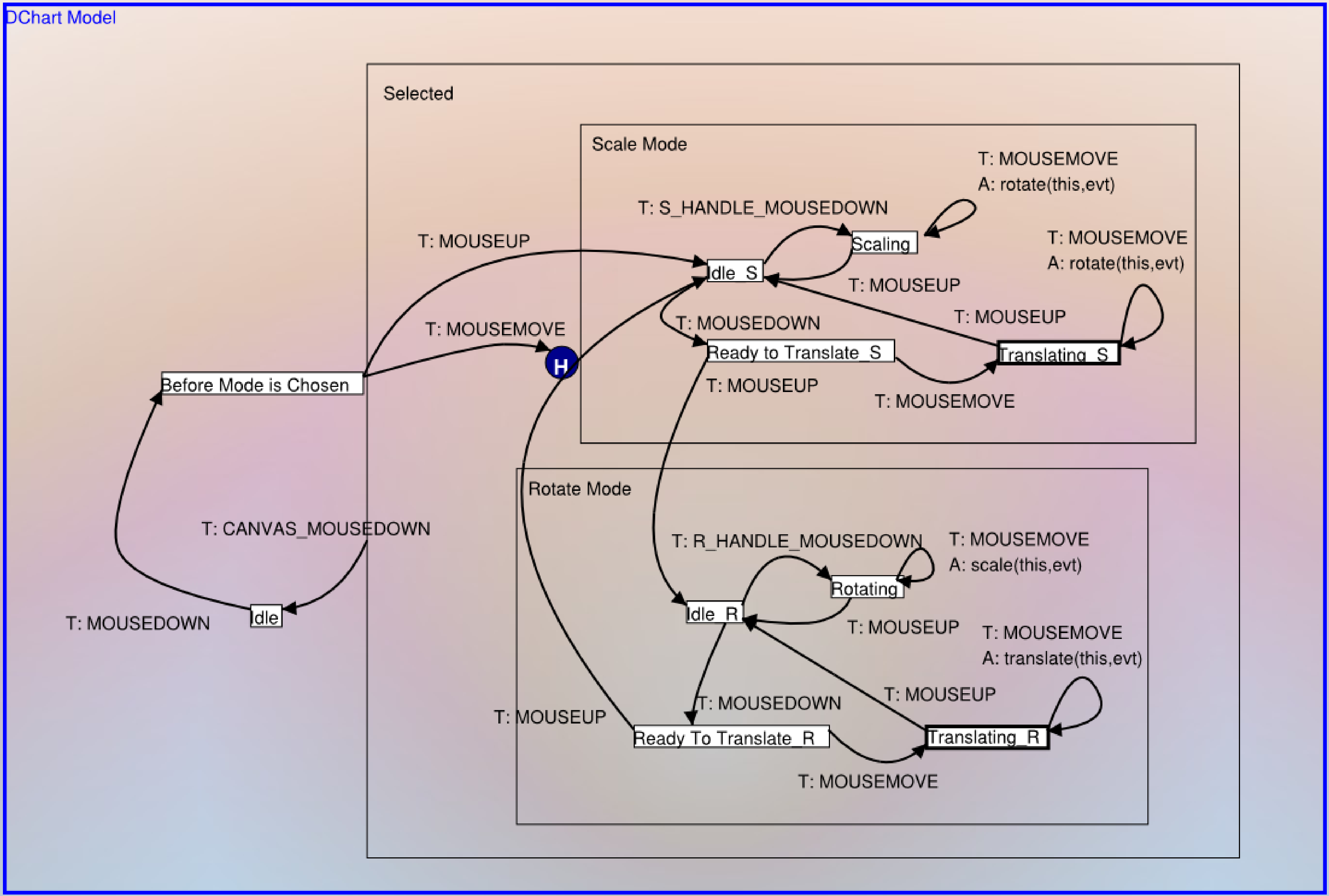
Inkscape Example (Behaviour)



Inkscape Example (Behaviour)



Inkscape Example (Behaviour)



Inkscape Example

- Compile with SCCJS
  - Produces .js file
  - Defines statechart constructor function
- Write “glue” code

Future Work

- Develop AToMPM
- Explore different “optimal” formalisms for UI specification and synthesis
- Explore use of HlS as “assembly language” for higher-level specification languages (such as Task Models)
- Analysis of models (possibly after transformation to appropriate formalisms such as Petri Nets)
- Target: handheld devices, from DSVLs