Lab 2: Modeling in Ptolemy II

Model of Computation

- Definition: A mathematical description that has a syntax and rules for computation of the behavior described by the syntax (semantics). Used to specify the semantics of computation and concurrency.
  - Example: FSM, Differential Equations

- An MoC is useful for
  - Capture the functionality
  - Verification of the functional spec
  - Synthesize part of the spec
  - To use different tool

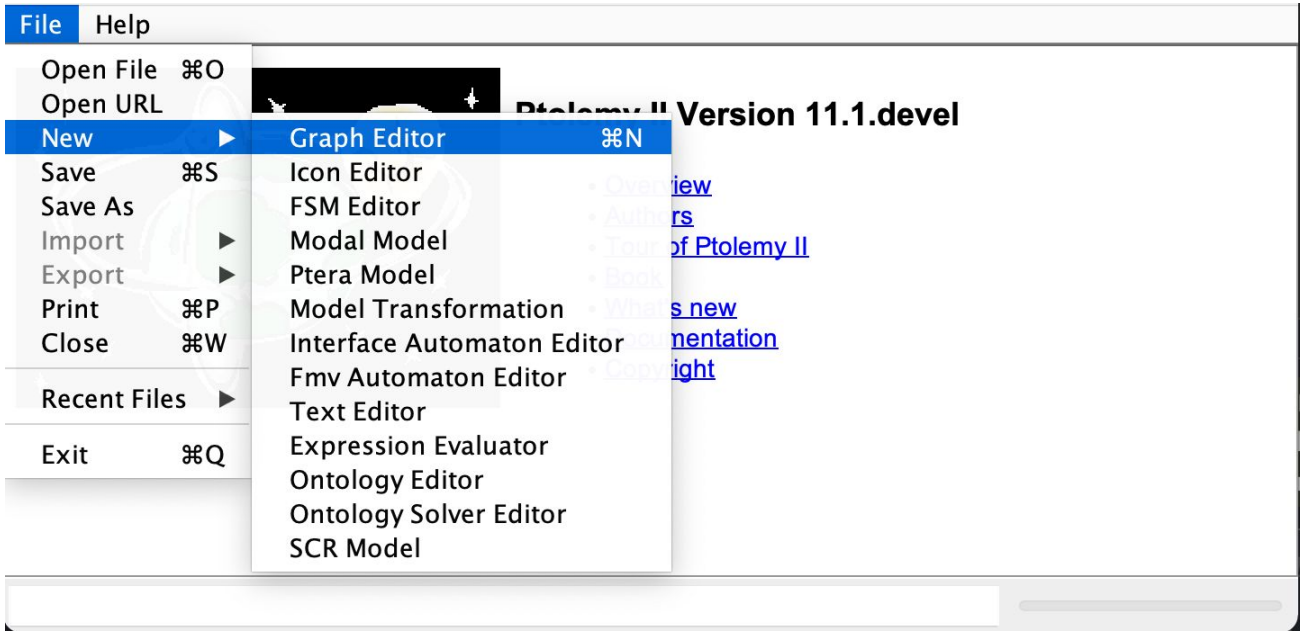
Ptolemy II

- An open-source software framework with actor-oriented design.
  - Actors: software components that execute concurrently and communicate through messages sent via interconnected ports.
- The semantics of a model (MoC) is not determined by the framework but by a component called “Director”
  - process networks (PN)
  - discrete-events (DE)
  - dataflow (SDF)
  - synchronous/reactive(SR)
  - rendezvous-based models
  - continuous-time models
- Allow hierarchical model
  - Each level can have its own director

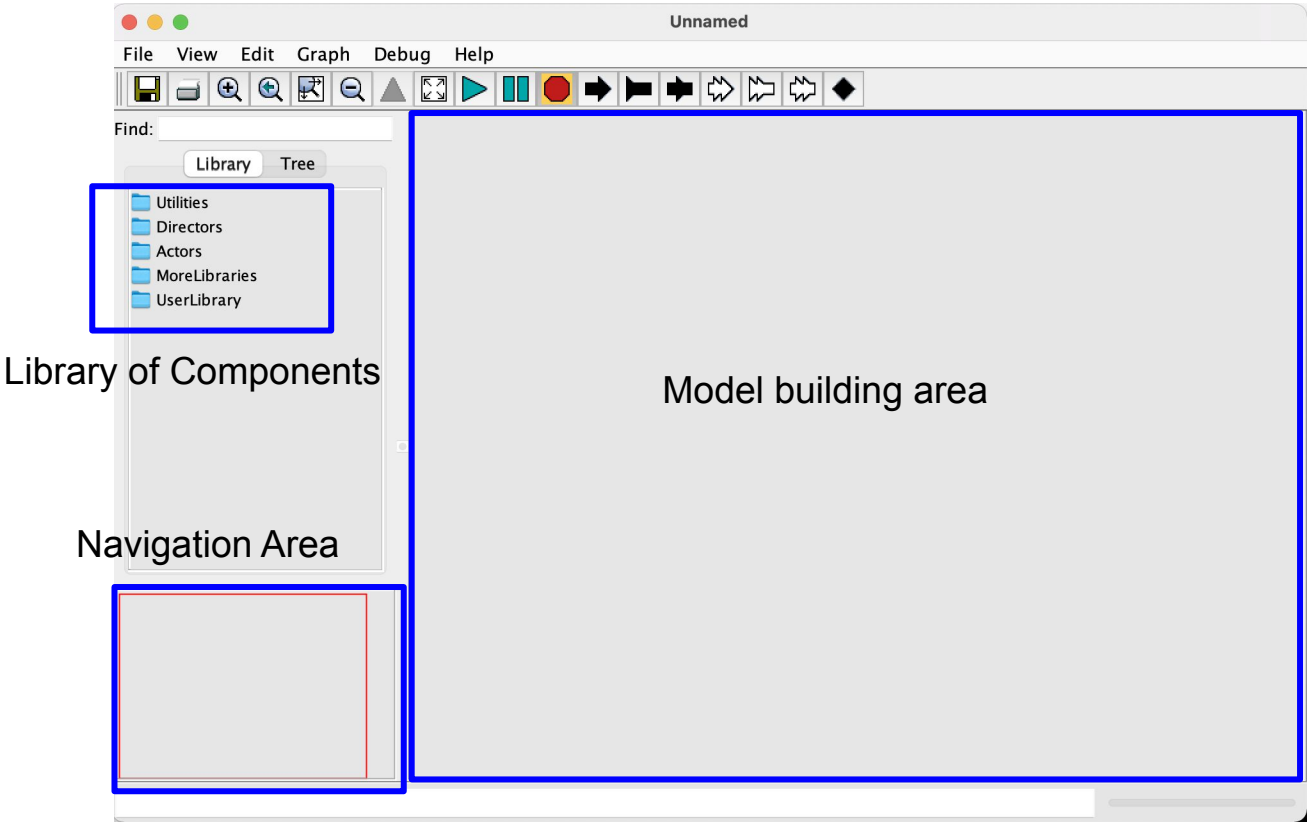
https://ptolemy.berkeley.edu/ptolemyII/index.htm

Introduction on Ptolemy II

- Turn on the Vergil application
  - File -> Graph Editor



The Graph Editor



Example: SDF

Resource

- Tutorial on more MoCs and how to use Ptolemy II
  - Claudius Ptolemaeus, Editor, "System Design, Modeling, and Simulation Using Ptolemy II," Ptolemy.org, 2014.

Lab Problem

Model and Design an automatic windshield wiper

- Visibility Requirements
  - Definition for visibility
  - Statement of the requirements based on the definition
- Weather
  - The environment that affects the visibility
- Rain Sensor
  - Interact with the environment and the windshield
- Windshield
  - An actuator that also changes the visibility

Target

- Make a good Abstraction
  - Prevent unnecessary information
  - Still faithful to reflect the design
- Requirements should be stated in the abstraction
  - Do not use vague natural language expression
  - Bad example: Too much water on the window
  - Potential way 1: The rain drops fall on the window are over 100 per second
  - Potential way 2: Over 50 ml water remained on the window
- The requirement and abstraction affect your design!

Deliverables

- Presentation on March 30th
  - How do you define the visibility requirement
  - How do you model the components/environments
  - What is the designs
  - Difficulties experienced during the lab
- Submission
  - Deadline: April 1st 23:59
  - Design files (.xml)
  - A PDF report answering the questions on the lab problem
  - Please compress the both files as .zip file and upload to bcourse

Lab Questions

1. What is your visibility requirement?
2. Explain your model. What is realistic about it? What is not? Why is it a useful model?
    a. Describe your weather model.
    b. Describe your rain sensor model.
    c. Describe your windshield wiper model.
3. Which models of computation did you use? How well do they match the dynamics of your components?
4. Which (component) parameters did you have to tweak to let the system satisfy the visibility requirement under the given environment assumptions?