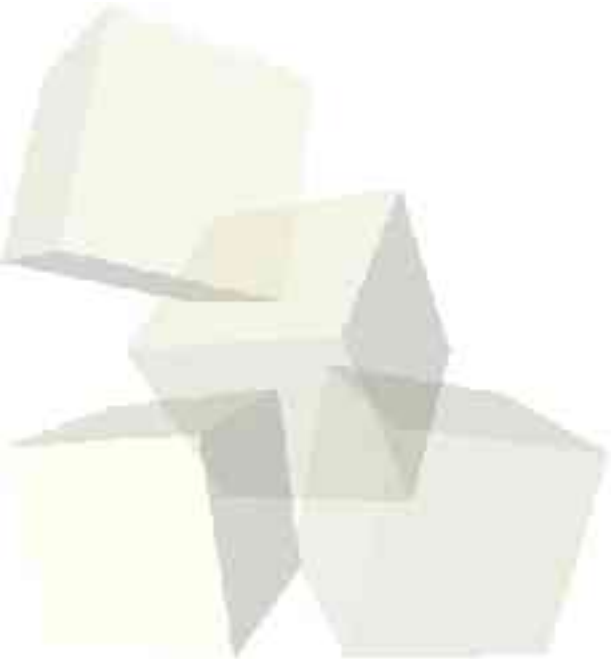
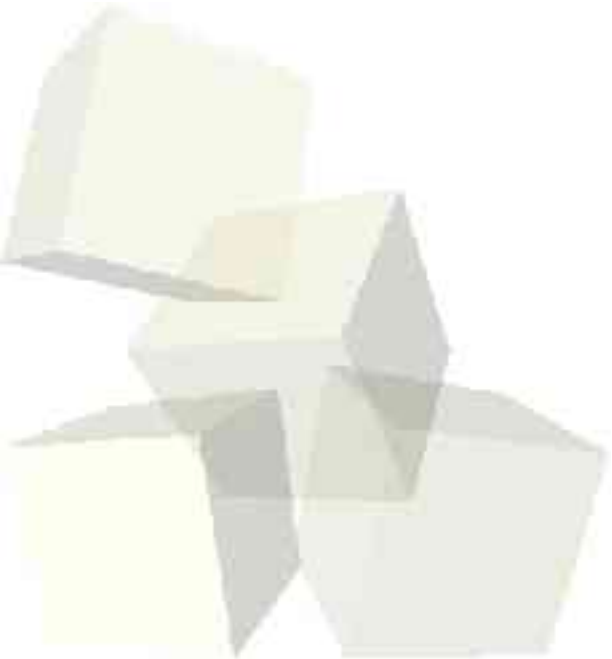
Event and Hybrid Declarative Modeling

2-8-2005



Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment? It's due a week from today.



Finite Event Automata

- Declarative models can also be done where we focus on the events that happen in moving from one state to another. For these we draw something very similar to a FSA only the circles are events and our model has the system move from one event to another.
- This style makes sense for some types of systems that are truly event based and what we care about is the events that are happening.

Hybrid Methods

- We can create other forms of diagrams that combine event and state information.
- One form of this is to have an FEA that has been enhanced with other symbols to denote aspects that store state information. Your book gives an example of this in figure 4.36 though I would almost swear that one of the arrows is backwards.
- Notice that in this usage the diagram is not specifying the whole system. It is a higher level view, like UML diagrams.

Petri-Nets

- Another form of hybrid model is the petri-net. These are a completely different type of formal model that is often used in research on parallel processing.
- A petri-net is defined by a set of places (circles), a set of transitions (lines), and a set of arrows that connect places to transitions or transitions to places.
- We also typically put markers in the places though formally each place just has a count for how many markers are in it.

How a Petri-Net Works

- A petri-net operates by having markers move from place to place, through transitions. In order for a transition to fire, there must be at least one marker in each of the places with an arrow to that transition. When a transition fires, one marker is removed from each place going into to and one marker is added to each place it goes out to.
- We can make a simulation out of a petri-net by attaching delay times to the transitions.

Running a Basic Petri-Net

- Your standard petri-net is run rather simply. At any instant, you look at all the transitions and from the set of transitions that can fire, you pick one at random to fire. If at any time you can't fire any transitions it typically means you have come to a deadlock though it is possible you just finished doing what you were supposed to do.
- The latter makes more sense in simulations than in operating system resource scheduling.

Writing a Petri-Net Simulation

- With time attached to a petri-net, the running of the net becomes a problem of event scheduling in a queue. Transitions take some time interval to fire between when the markers are removed from the input places and when they are placed in the output places. If many transitions could fire at any instant, we pick as many as we can do at the same time and randomly resolve conflicts. That schedules an event for when the transition should end.

Minute Essay

- Draw a petri-net that shows the process of making hot dogs at a cookout.