Static Protocols

- Merely "scripts" that the agent follows
  - Typical notation is state diagrams
  - Petri nets are a generalization of state diagrams

Ad-hoc Protocols: Petri Nets

Buyer: Request to Buy
Seller: Request Payment
Seller: Accepting to Sell
Buyer: Accepting to Pay
Buyer: Paying for item
Seller: Accepting Payment
Seller: Handing out item
Buyer: Accepting item

Success

Simple Petri Nets

- Directed arcs only
- Strictly bipartite
- "Tokens" occupy places (a place can contain any number of tokens)
- A transition can fire only when there is sufficient tokens at all of the incoming places.
- The transition firing consumes tokens at the incoming side and emits tokens to the places on the outgoing side.

Transitions

- If ALL of a transition’s incoming places have tokens, it MAY fire
- So if there’s a single token available for more than one transition, only one of them MAY fire

- A transition will place tokens on all of its outgoing places IF it fires
Transitions

• EVERY incoming place must have sufficient tokens available in order for it to fire

• Places can contain any number of tokens

Lots of other kinds of Petri Nets
– Weighted Petri nets
– Coloured Petri nets
– Hierarchy
– Vector addition system with states
– Prioritised Petri nets
– Timed Petri nets
– Stochastic Petri nets
– etc.

• Could do the whole course on Petri Nets!

Example: “call for proposals”


Example: A produce Protocol

This is an extension of colored Petri nets called “reference nets”

Issues

• Rigid and Brittle
• Hard to account for all eventualities
• See class exercise to convince yourself of this