Protocols

Protocols

- Protocol: rules for sending and receiving messages
  - programming with agents

- Examples
  - broadcasting messages to group of agents
  - choosing an agent

- Important properties of protocols
  - deadlock free

Broadcast

- Just send message M to all agents As

```
proc {Broadcast As M}
   for A in As do {A M} end
end
```

Broadcast

- Let us take the agent-oriented view
- We define a Broadcaster Agent
- Any Agent that wants to broadcast sends a broadcast request to the Broadcaster

```
proc {Broadcast As M}
   for A in As do {A M} end
end
```

Broadcast

- Broadcaster function BF

```
fun {BF S M}
   case M
   of init(As) then state(agents:As)
   [] broadcast(M1) then
      for A in S.agents do {A M1} end
      S
   else S end
end
```

Broadcaster Scenario

- Broadcaster function BrowseF

```
fun {BrowseF S M}
   {Browse (S.name) # M } S
end
Broadcaster = {NewAgent BF state}
As = for I in 1..10 collect:C do
        {C {NewAgent BrowseF state(name:I)}}
     end
{Broadcaster init(As)}
{Broadcaster broadcast(a)}
```