University of Nottingham
UK | CHINA | MALAYSIA

The Art Of Iterating Update-Strategies in ABS

—

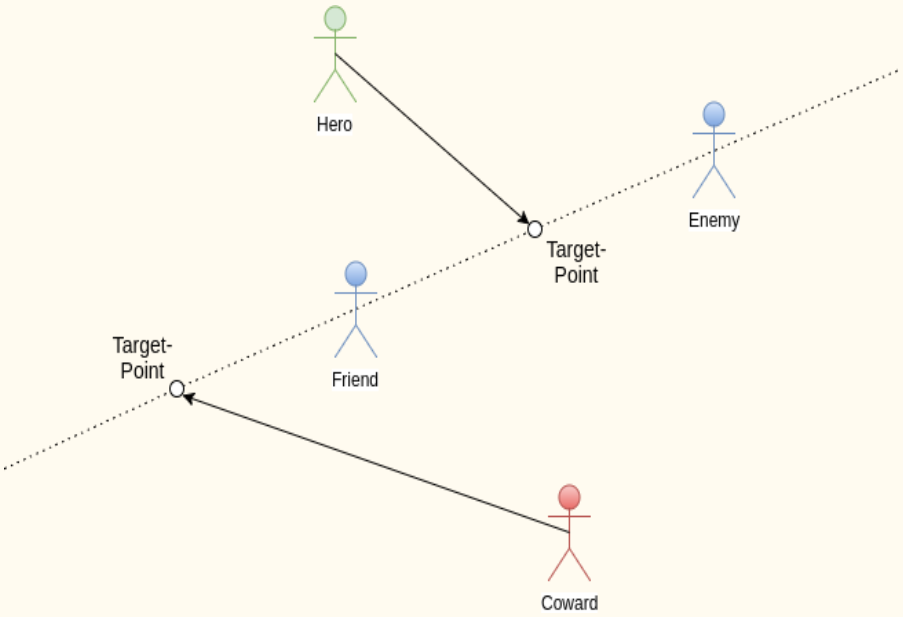
Jonathan Thaler
University of Nottingham

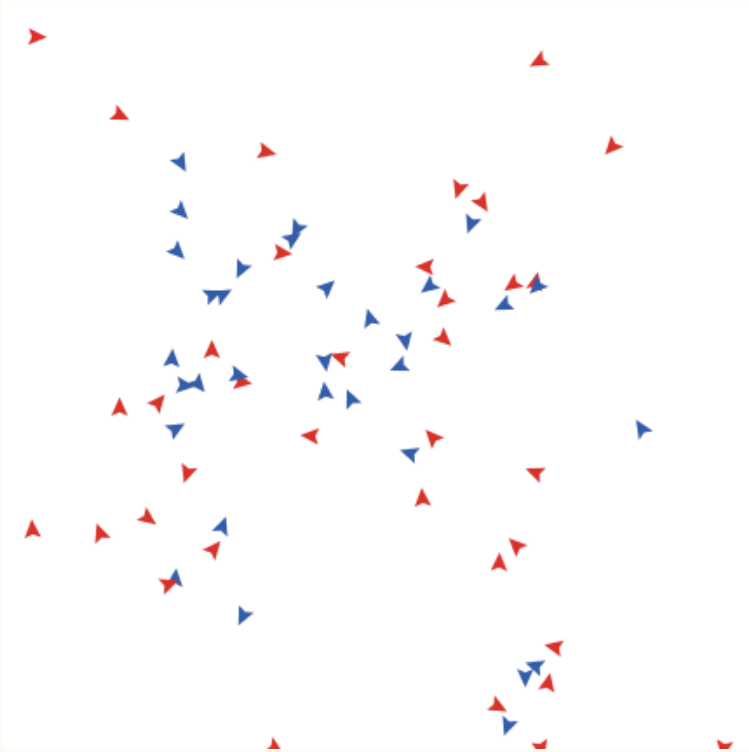
Agenda

- Motivating Examples
  - Heroes & Cowards
  - Prisoner's Dilemma 2D
- The Agent Metaphor
  - How can an Agent be awakened in a computational environment?
- Update-Strategies
  - Four ways of awakening Agents
- Conclusions

Motivating Examples

Heroes & Cowards (Wilensky & Rand, 2015)





Prisoner's Dilemma 2D (Nowak & May, 1992)

- Play Prisoner's Dilemma over multiple steps on 2D grid
- Two prisoners cooperate (say nothing)
  or defect (betray each other)
- 4 outcomes & rewards: $T > R > P > S$
  - R: both cooperate
  - P: both defect
  - T / S: one defects (T) other one cooperates (S)

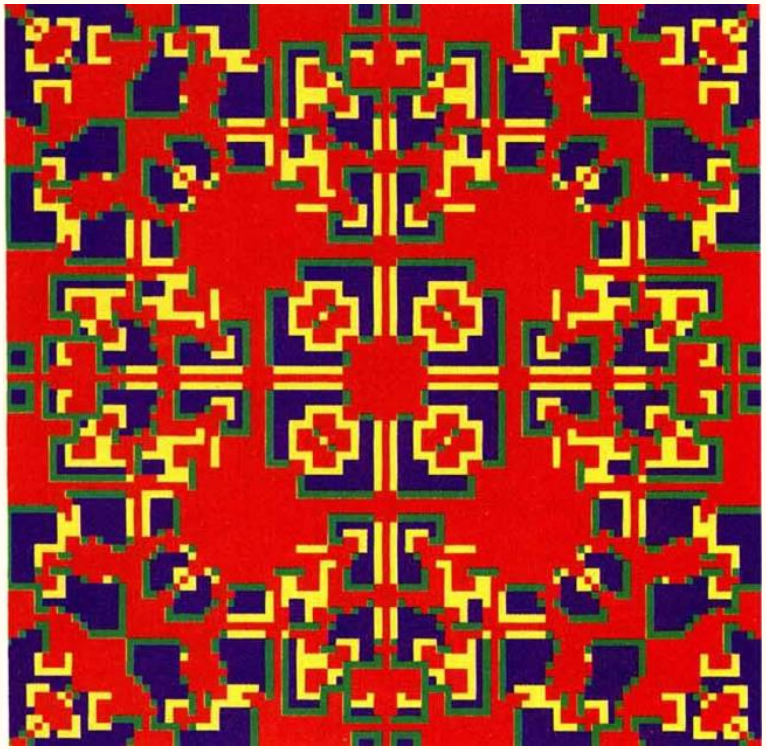
Prisoner's Dilemma 2D cont'd

99x99 Grid

single defector at center (50/50)

1. Play game with neighbours & self, sum reward
2. Compare payoffs, adopt role of best reward

Blue: Cooperators now & before
Red: Defectors now & before
Yellow: Cooperate now, defected before
Green: Defect now, cooperated before

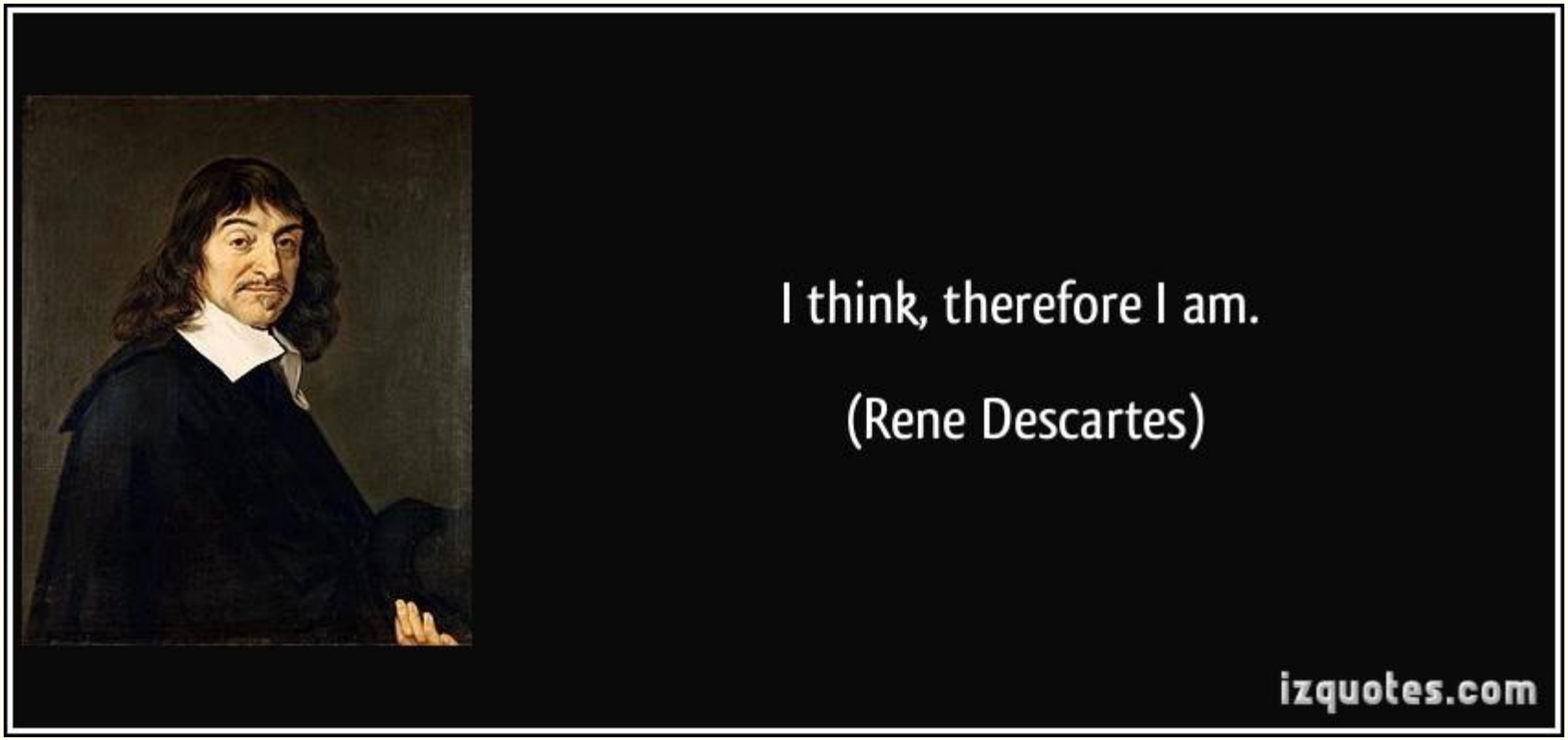


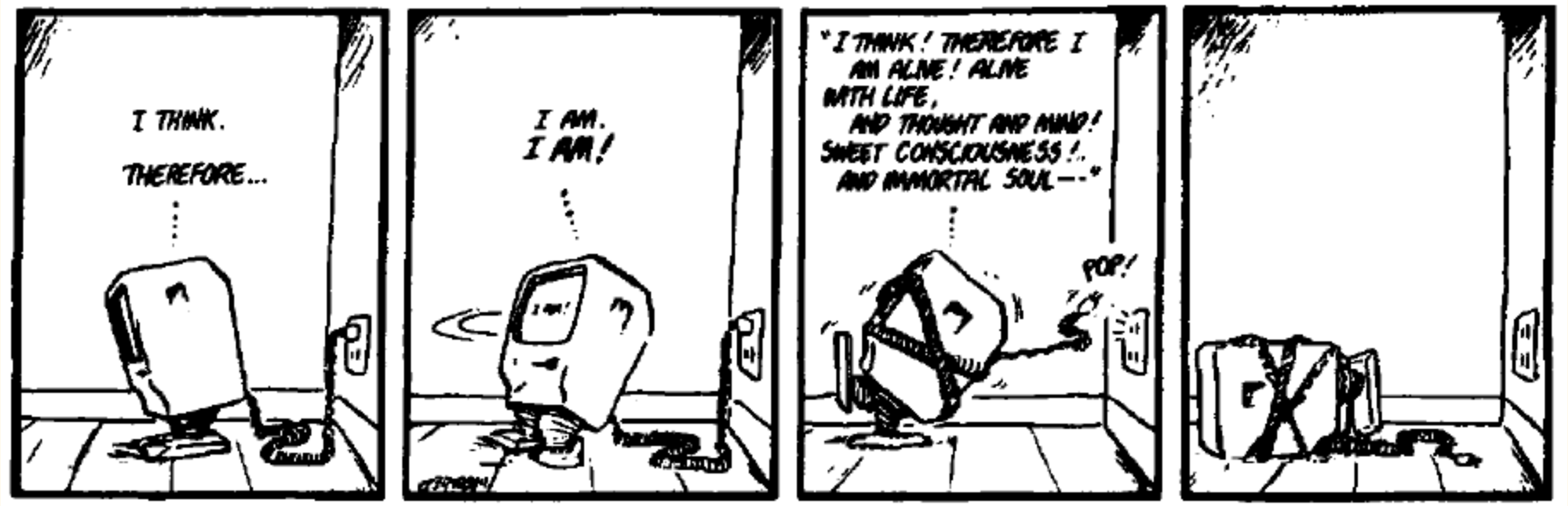
The Agent Metaphor

Agent

- Unique entity with internal state
- Agent-Environment interaction
- Agent-Agent interaction
- Pro-Active

How can we raise Agents to existence in a computational environment?


I think, therefore I am.
(Rene Descartes)
izquotes.com


I THINK.
THEREFORE...
I AM.
I AM!
"I THINK! THEREFORE I AM ALIVE! ALIVE WITH LIFE, AND THOUGHT AND MIND! SWEET CONSCIOUSNESS!.. AND IMMORTAL SOUL—"
POP!

Pro-Activity

- Need *independent* stimulus
- Independent stimulus allows to perceive change
- Time = Change = Time

- Time in a computational environment:
  being executed / updated

- Time a variable: increasing between updates

Agent Updating

- How execute a set of agents?

- When is message M
  Agent A to Agent B
  visible to Agent B?

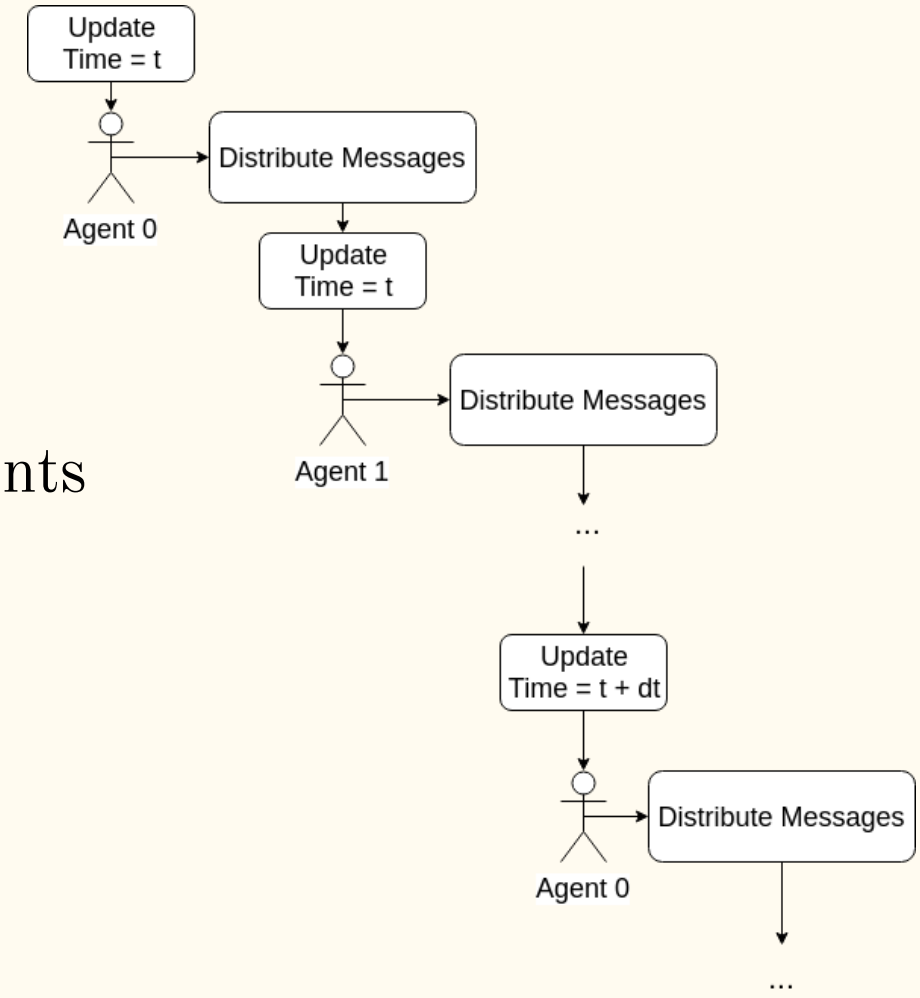
- When is message M
  processed by Agent B?

Update-Strategies

I: Sequential

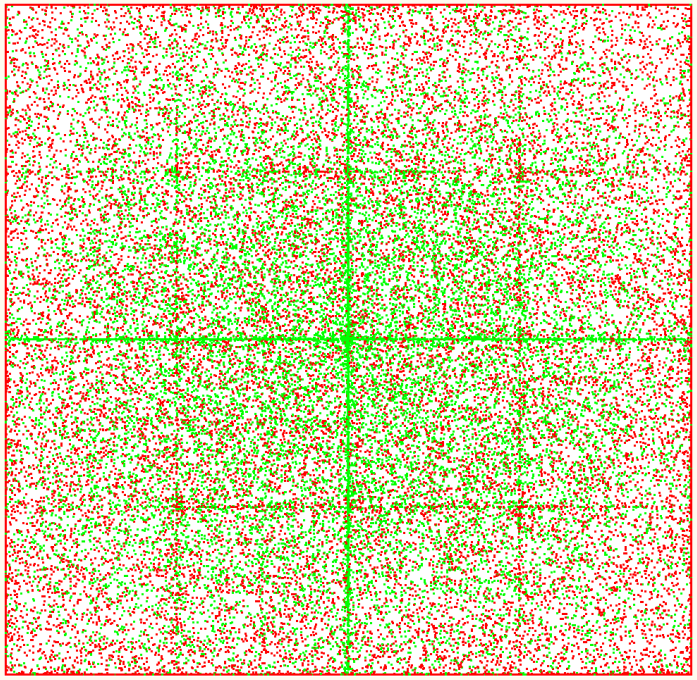
Agents act after another

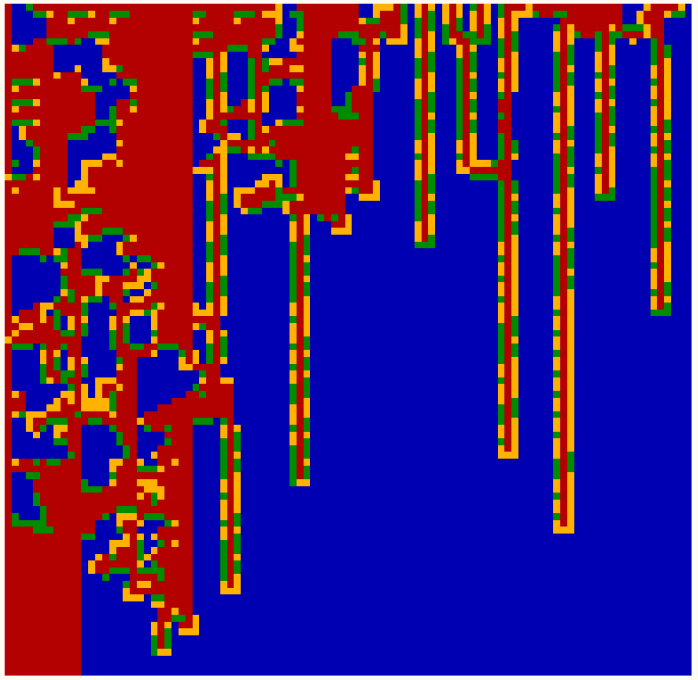
- Global, absolute Time
- Agents see changes of Agents before
- Single, shared Thread
- Deterministic



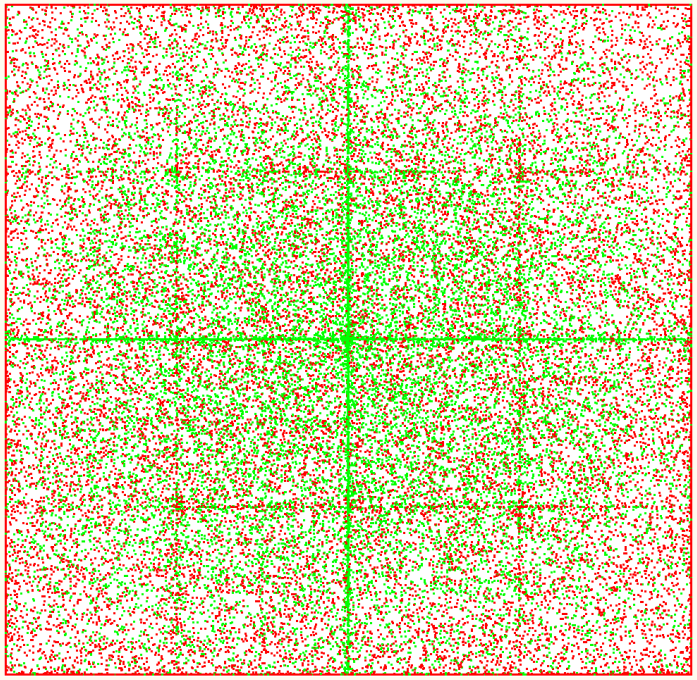
- Imperative / OO languages (Java, C++) strong

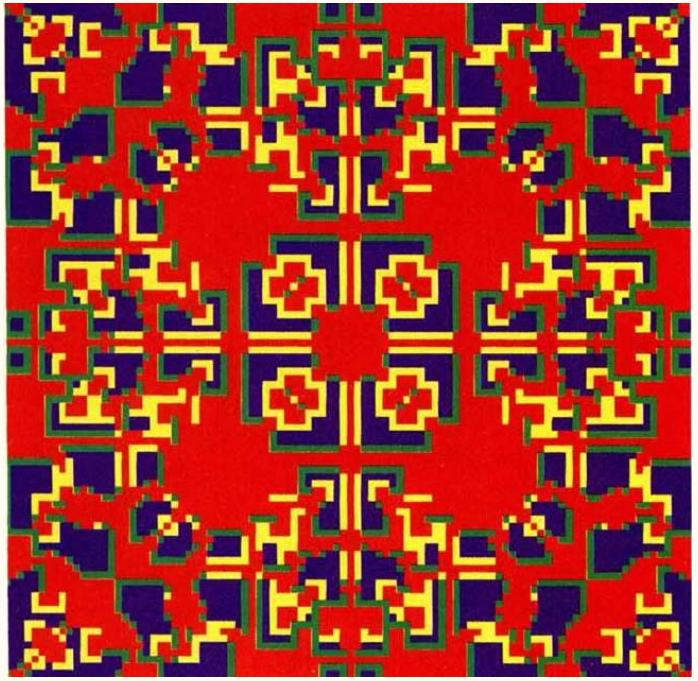
I: Sequential





I: Sequential

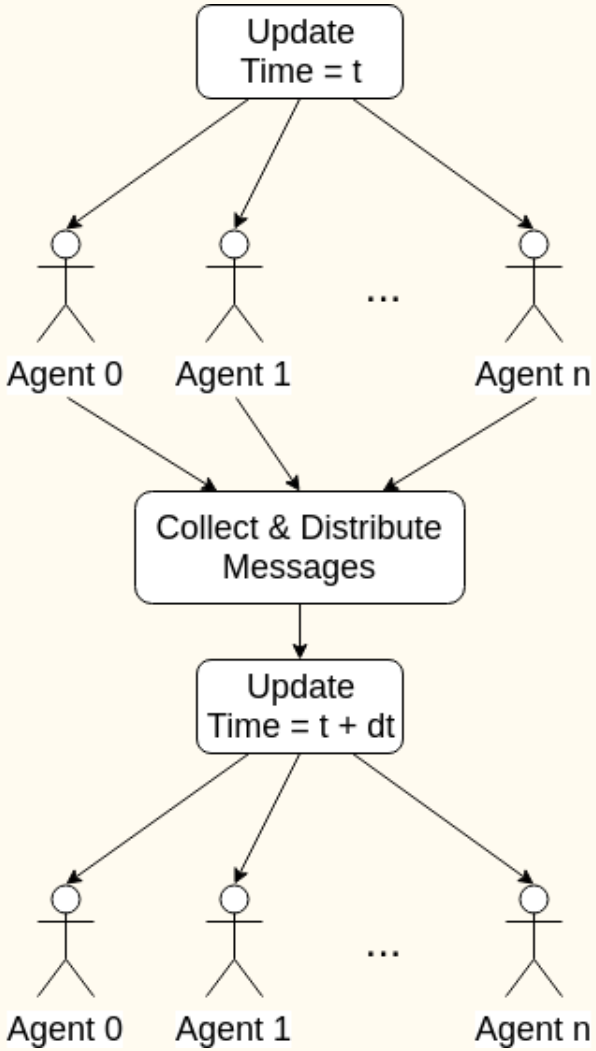




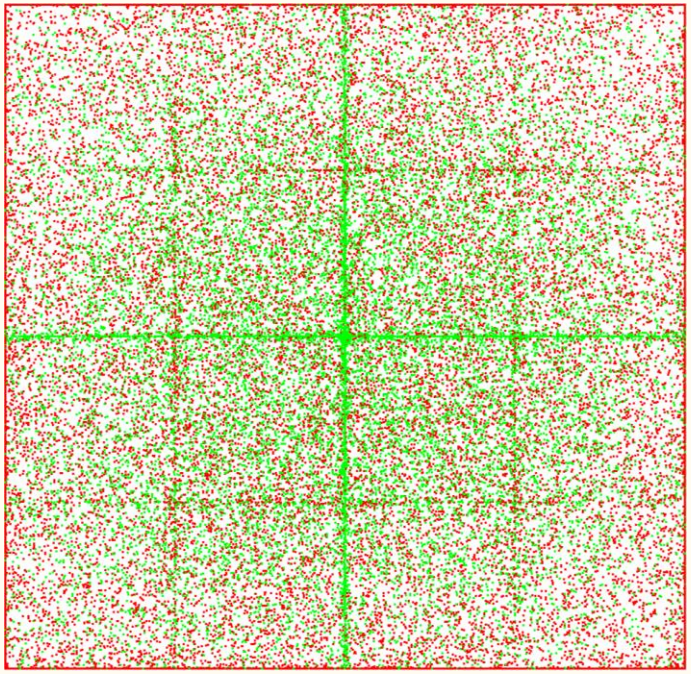
II: Parallel

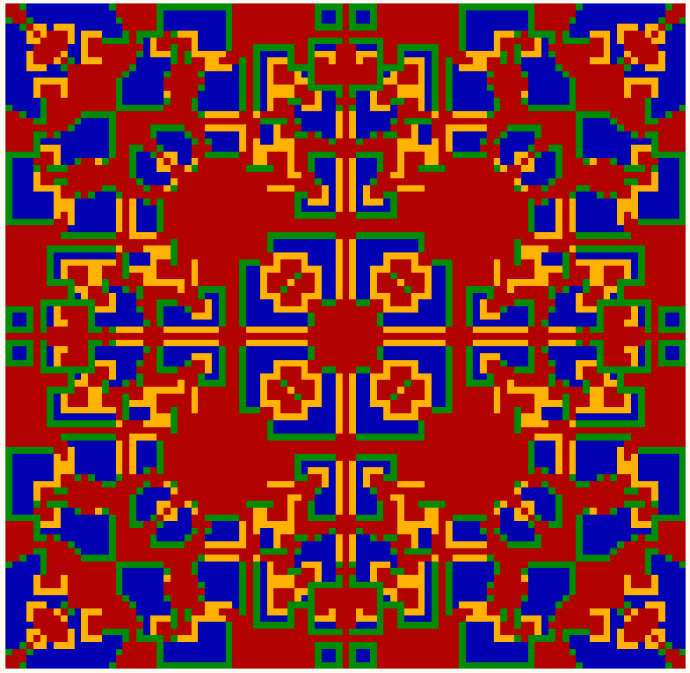
Agents act at the same time in lockstep

- Global, absolute Time
- Actions **not** visible during update-step
- Single shared / separate Threads
- Deterministic
- Functional languages (Haskell, Clojure) strong

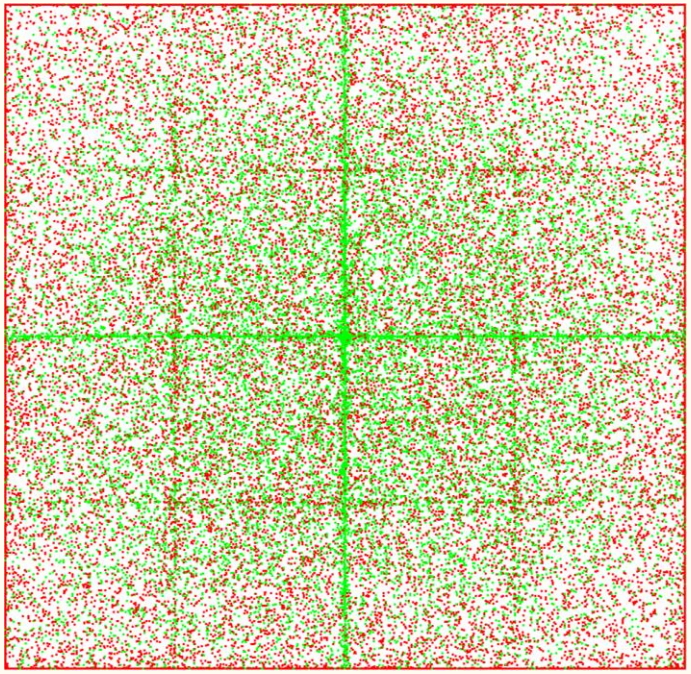


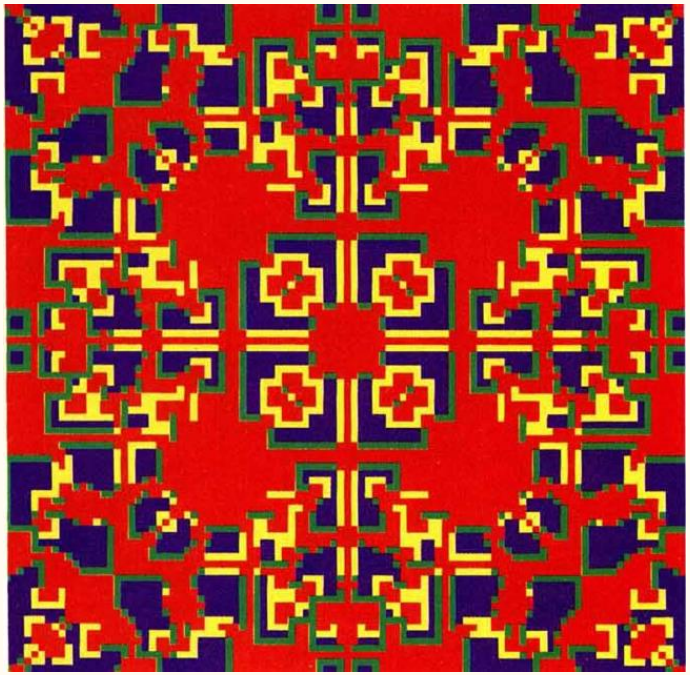
II: Parallel





II: Parallel

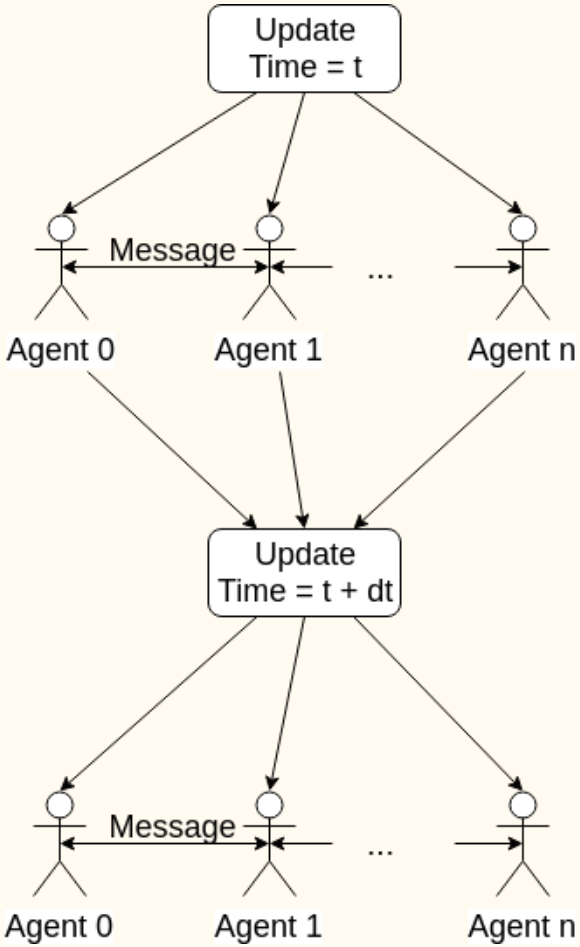




III: Concurrent

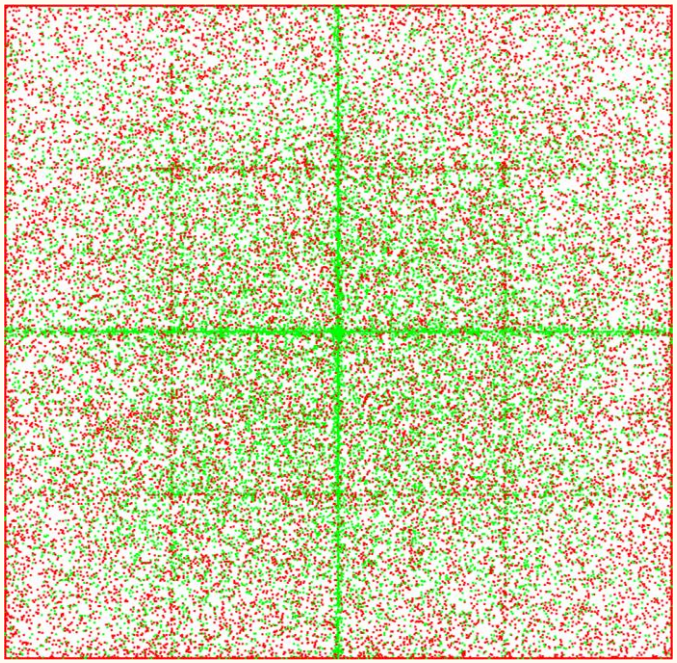
Agents act concurrently at the same time in lockstep

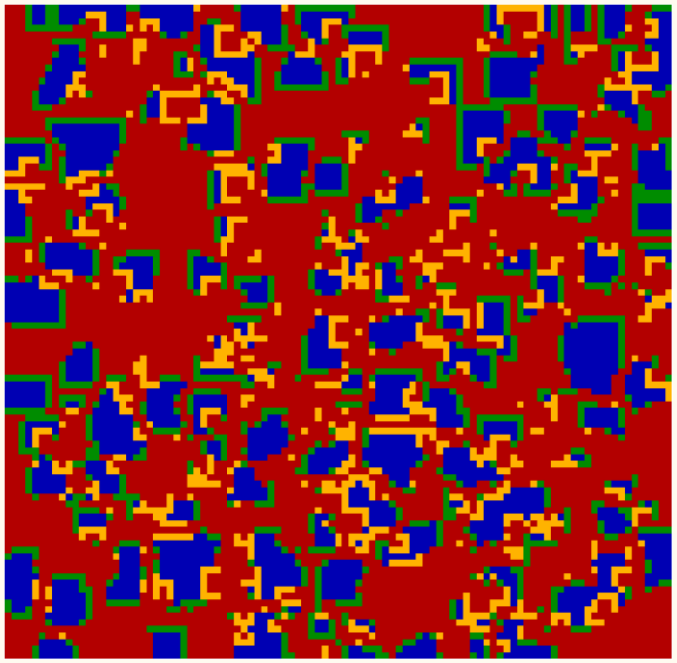
- Global, absolute Time
- Changes visible **within** update-step
- Separate Threads
- Non-Deterministic



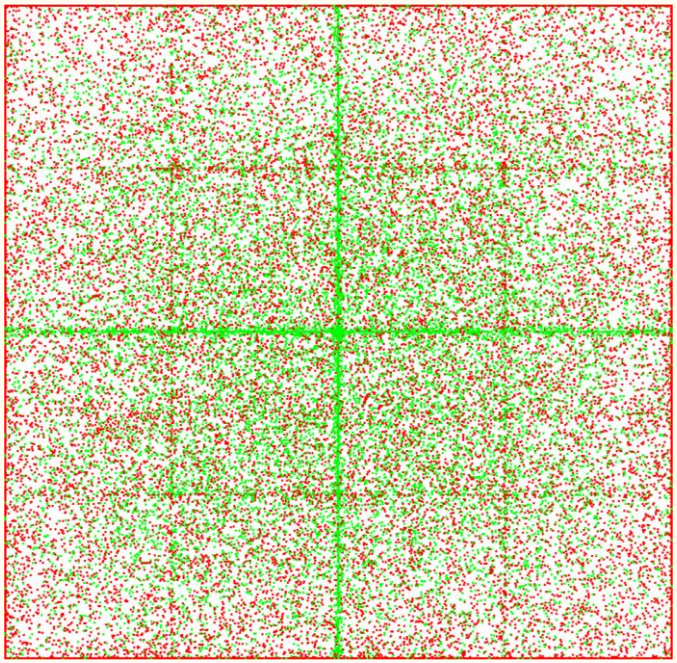
- Languages with concurrency features (Java, Haskell) strong

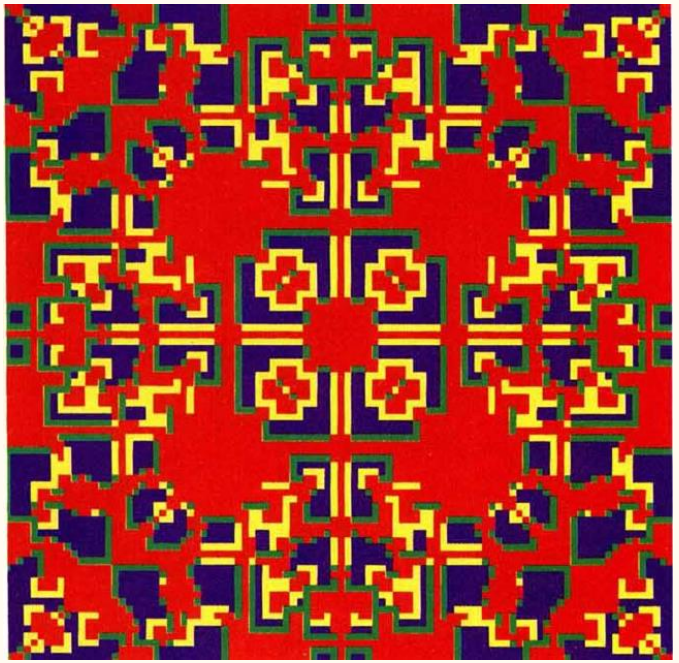
III: Concurrent





III: Concurrent

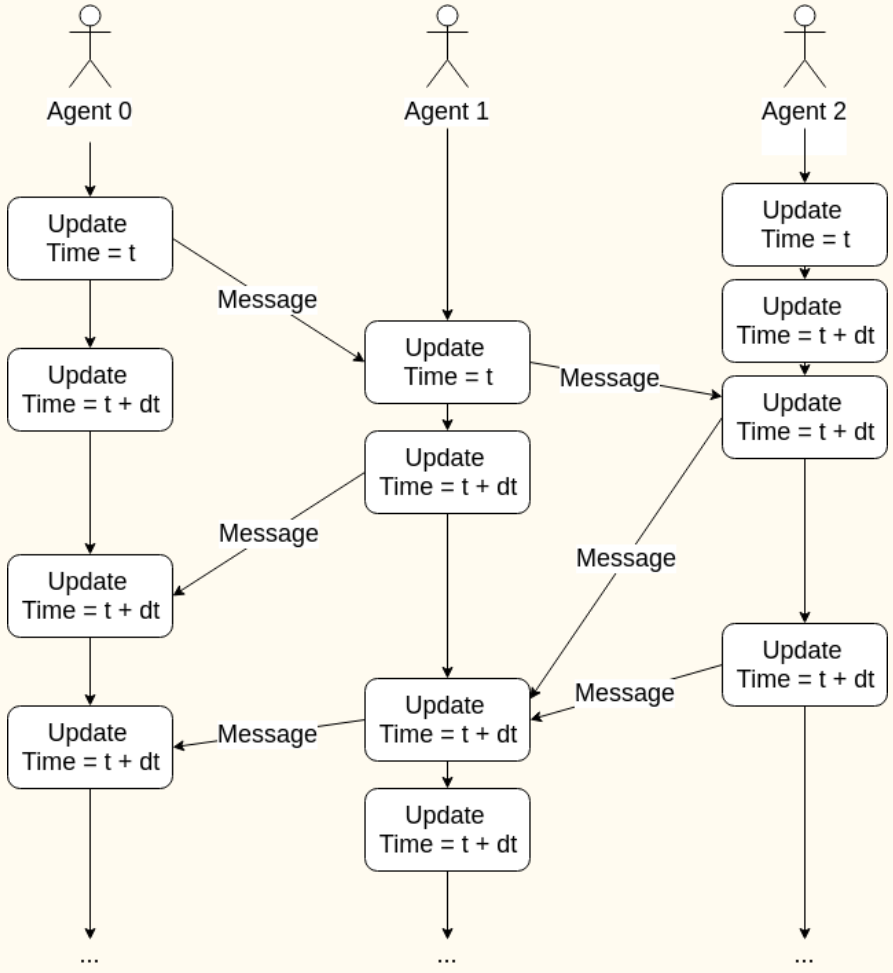




IV: Actor

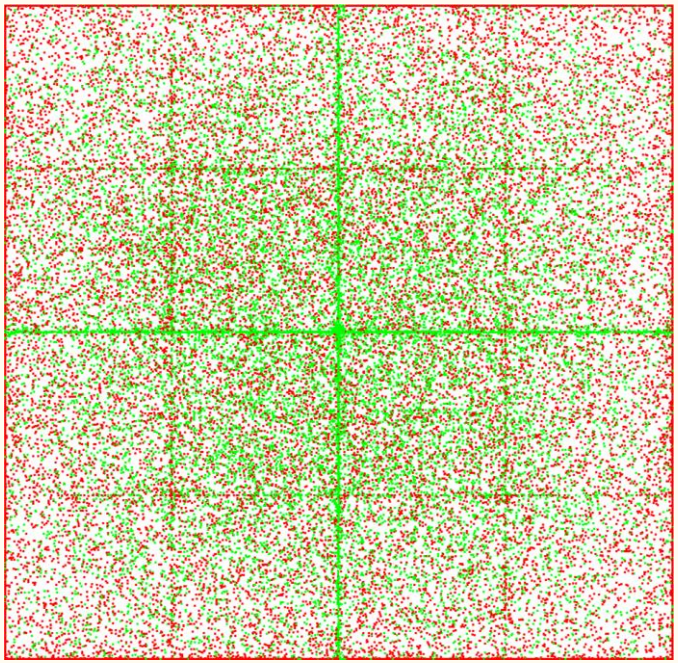
Agents act at the same time in their own relative universe

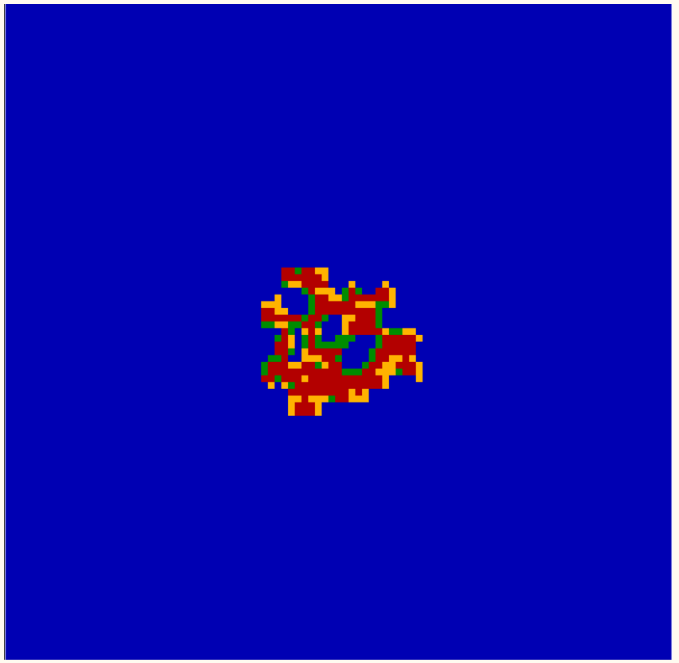
- Local time
- Separate Threads
- Non-Deterministic



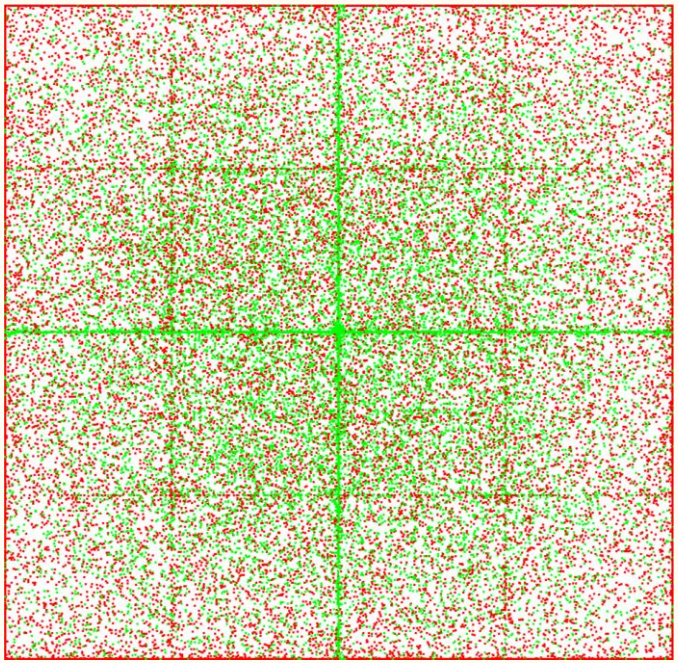
- Actor-based languages (Erlang, Scala) strong

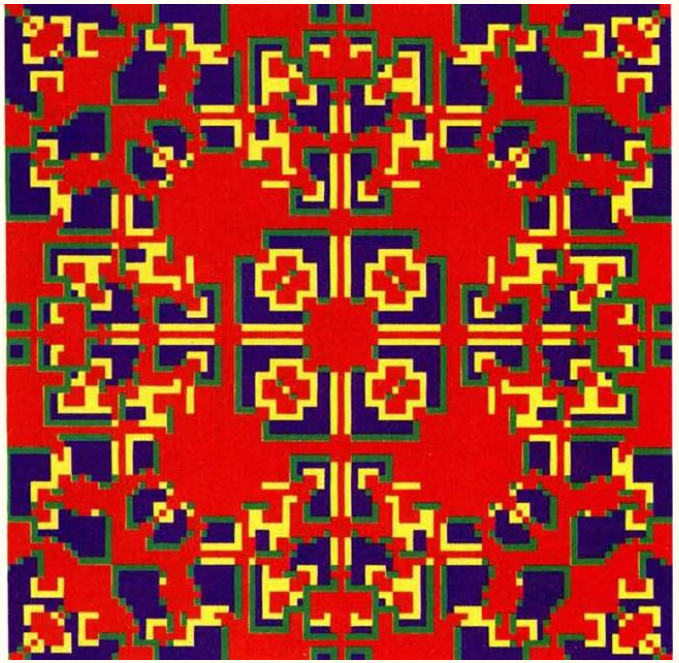
IV: Actor





IV: Actor





Conclusions

- Pro-activity requires independent stimulus
- Independent stimulus:
  Change = Time = being executed / updated
- Match update-strategy to semantics of the model
- Programming paradigms make a difference

Q & A