# Towards a RTC agent algorithm – some initial rules of thumb

**It's complex and evolving but here is my take on some heuristics:**

- Observe the environment carefully
- Suppress desire to act spontaneously (not to act is also to act)
- Consider at least 4 actions you can take and consider each in relation to the existing environment
- Your action may:
  - reinforce (copy / add to / continue) an existing pattern in the environment (evidenced from at least two previous actions)
  - Begin a pattern suggested by a first action (make a second action)
  - Start a new action
- Try to make actions clear to observers of the environment
- Try to avoid starting a new action unless the environment suggests it: such as looping or a physical constraint
- Be creative in continuing a pattern with the materials available and the environmental constraints
- Limit your communication to actions within the environment

# An outline algorithm

- Given environment E = {f1,f2,f3..} where f are observable features (or patterns in) E
- A set of agents P = {p1,p2,p3..} have access to E
- For each time step t
  - each agent observes E and forms a set S = {null,a1,a2,a3..} of potential actions, where null = no action, up to some maximum
  - Each action may extended an existing feature (or pattern) or create a new one with a bias toward extending [tricky bit]
  - By some maximum time t+delta each agent selects an action from S and performs it on E changing the feature set of E
- Repeat next time step

*[note: each agent may / should have several different / overlapping feature (pattern) recognition modules and action repertoires]*

# Trivial example

- E = a network of nodes undirected links (graph), f = each node and it's links

- P = population of agents all the same

- Potential actions = copy a random f or add a new node and link to one randomly chosen existing node (create a new f)

- Repeat

  *[from ANY initial E – including nothing - we get something like....]*