Where do we go from here?

Project CyberSym Part I - Progress Report
Janosch Haber 07-05-2015

Scenario 1) - Emergent community spirit.
Assumptions: Myopic agents with Wissner-Gross' formula of intelligence

1) Let agents request the final Product as well as all its parts and basic Resources from the Agents in its environment.
2) Create a `Hashtable<List<Character>>, LinkedList<Agent>` as an individual Agent's field to propagate Demands through the network. This keeps track of the original Agent who demanded for a Resource or Product, gives an indication of necessary travelling distance and prevents circular chains of demand.
3) Add a discount for repetition of actions. Either all actions can become cheaper (the action in general or for given Resource types only) or the amount of resources and/or products produced by one action is increased. In that case, it must be decided whether Sources are depleted faster through this or remain unaffected (e.g. through higher extraction efficiency)
4) Create functions to generate or estimate the number of possible future states connected to an action
5) Let agents choose the action connected to the largest number of possible future states
6) Measure the number of dying Agents and depleted Resources. Does the system become stable? How long can it sustain itself? Measure the number and weighing of individual Agent's chosen actions: Do agents assume a certain role over time? If so, Efficiency in sustaining the community emerges through the individual Agent's intelligence.

   Potential problems:
   1) How to determine the number of possible states / a measurement of the effect of an action. The number of possible actions can become very large, future prediction probably is too computationally complex (and too difficult for the given project duration). Reasoning on consequences also can become highly complex: "If I extract the Resource I need from this Source, it might be exhausted in 5 ticks. If I instead get it from this one (which is larger), the other Source can regenerate and stay longer. If I however choose to request it from other Agents, both Sources in my neighborhood can regenerate. But then it might take longer to get it. In the meantime I however can extract another Resource requested by my neighbor, because he might die if he doesn't get it…" A feasible approach (probably using heuristics) is needed to tackle this issue.
   2) How to evaluate / compare the system? Maybe it can sustain itself without that "intelligent" decision making, just because there is no Resource scarcity. So maybe compare with a simulation where some of the Agents are assigned "egoistic" settings who do not forward Resources (or something like that).

Scenario 2) - Collective time savers

Assumptions: Myopic agents choosing for the best cost/reward ratios.
Steps 1-3 are the same as in scenario 1.

1) *Let agents request the final Product as well as all its parts and basic Resources from the Agents in its environment.*
2) *Create a `Hashtable<List<Character>>, LinkedList<Agent>` as an individual Agent's field to propagate Demands through the n etwork. This keeps track of the original Agent who demanded for a Resource or Product, gives an indication of necessary travelling distance and prevents circular chains of demand.*
3) *Add a discount for repetition of actions. Either all actions can become cheaper (the action in general or for given Resource types only) or the amount of resources and/or products produced by one action is increased. In that case, it must be decided whether Sources are depleted faster through this or remain unaffected (e.g. through higher extraction efficiency)*
4) Create functions to generate or estimate the cost of performing an action/ the cost of all actions necessary to reach the final goal of that action (extraction, assembly, transport…)
5) Let agents choose the action with the best cost/reward ratio. The reward is determined by the number of time units added to an Agent's lifespan when
6) Measure the number of dying Agents and depleted Resources. Does the system become stable? How long can it sustain itself? Measure the number and weighing of individual Agent's chosen actions: Do agents assume a certain role over time? If so, Efficiency in sustaining the community emerges through the selection of "cheapest" actions.

   Potential Problems:
   1) How to determine/estimate the total cost of actions necessary for producing a final Product? Maybe partial determinations are sufficient - extracting, assembly and transport cost still are readily available
   2) Maybe additional restrictions are necessary, otherwise might an Agent always choose the action that is cheapest to him - and for example deplete the Sources in his environment without forwarding the acquired Resources
   3) Is there "enough AI" in this system? Decisions are actually determined by the system settings - there seems to be no real self-regulating as in cybernetic systems like proposed by Beer

Scenario 3) - Time will tell

Assumptions: Myopic agents "learning" best behavior through ML in multiple iterations
Steps 1-3 are the same as in scenario 1 and 2.

1) *Let agents request the final Product as well as all its parts and basic Resources from the Agents in its environment.*
2) *Create a `Hashtable<List<Character>>, LinkedList<Agent>` as an individual Agent's field to propagate Demands through the n etwork. This keeps track of the original Agent who demanded for a Resource or Product, gives an indication of necessary travelling distance and prevents circular chains of demand.*
3) *Add a discount for repetition of actions. Either all actions can become cheaper (the action in general or for given Resource types only) or the amount of resources and/or products produced by one action is increased. In that case, it must be decided whether Sources are depleted faster through this or remain unaffected (e.g. through higher extraction efficiency)*
4) Give a static array of choices that can be selected from given a certain situation.
5) Let agents choose actions randomly and record their consequences (e.g. rewards obtained during the next few cycles, number of dying agents, number of depleted resources etc.
6) Put a ML layer on top of that and let it determine the best actions recorded in combination with the best consequences
7) (Automatically) alter the Agent's decision weighing in accordance with the results of the ML step
8) Measure if system performance improves over simulations. Possible features: System sustainability (duration of survival or stability); number of dying Agents; Number of depleted Sources; Total cost/value of the assembled products

   Potential Problems:
   1) Correlation vs. Causality. The chosen action not necessarily (or most probably really not) effects the distribution of rewards in consequent cycles as producing a final product takes a larger number of cycles. Thus, there might be no measurable causality between performed actions and the consequences thereof (except maybe depletion of resources)
   2) Difficulty of implementing ML that fits on the current system design