The Problem of Coordination in Self-Organizing Systems

Francis Heylighen

Evolution, Complexity and Cognition group (ECCO)

Free University of Brussels (VUB)

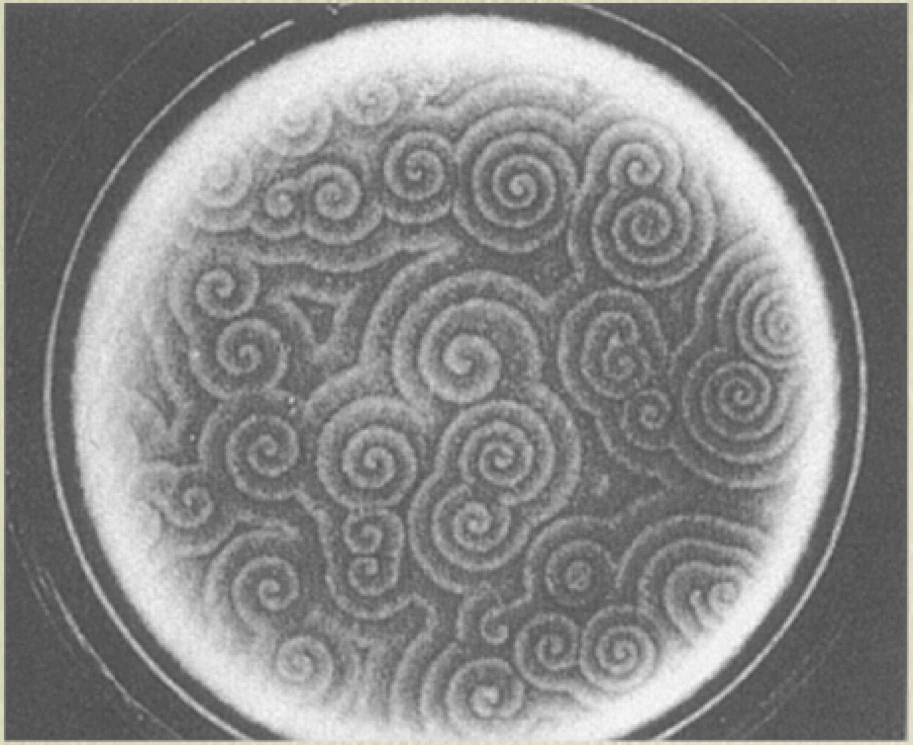
Self-organization

Spontaneous appearance of order or organization

Global **order from** ***local*** **interactions**

Organization ***distributed*** **over all the components**

- collective
- Robust



Agent

Elementary system component

- E.g. molecule, ant, person, robot, sensor, module

Acts on environment

- In response to experienced conditions

Goal-directed

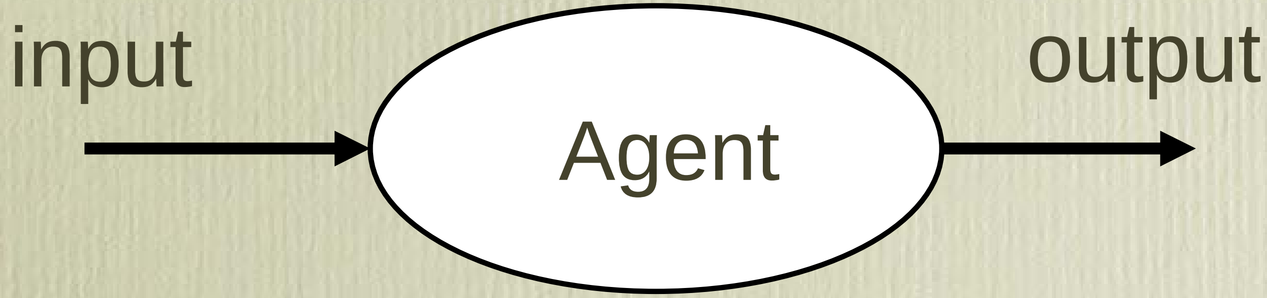
- Actions lead to an attractor or “high fitness” state

Agent as input-output system

Input: sensed state of the environment

Output: new state of environment resulting from action

Agent: “black box” transforming input into output



Self-Organizing System

Collection of many interacting agents

Interaction:

- action by agent A triggers action by B
- Which triggers new action by A, C, D…etc.

Interactions propagate throughout the system

- Local action may have global effects

Goal of System = Attractor of global dynamics

Function of a System

Function = what the system (robustly) does

- System itself determines its purpose

“Does” = transforms input into output

- Represented by input output-function
- f: I → O: i → f(i)

“robustly” = in a stable, invariant matter

- Invariant under further evolution or most perturbations
- The “attractor” of the system’s dynamics is the implicit goal of the system

Controlling Emergent Function

How can we make sure that the global function is as desired?

Input = problem or query

- initial information or situation to be dealt with by the system

Output = solution

- processed information or changed situation that answers the query

Example: neural network

- input = pixels to be recognized
- output= recognized character

Engineering SOS

Design a system consisting of interacting agents

- overall function specified by the designer
- agents initially specified by the designer
- interactions patterns *not* specified

How can the system self-organize to achieve the function?

- 1) impose global function from the outside:
  - **Problem of training**
- 2) let agents discover effective pattern of interaction
  - **Problem of coordination**

Problem of Training

Approach in Evolutionary Algorithms

- Define fitness function
- Let agents and interactions undergo variations
- Retain and multiply fittest ones
- Eliminate least fit ones

Approach in Neural Networks/Reinforcement learning

- Reward /reinforce “good” connections/behavior
- Punish/suppress “bad” connections/behavior

Problem of Credit Allocation

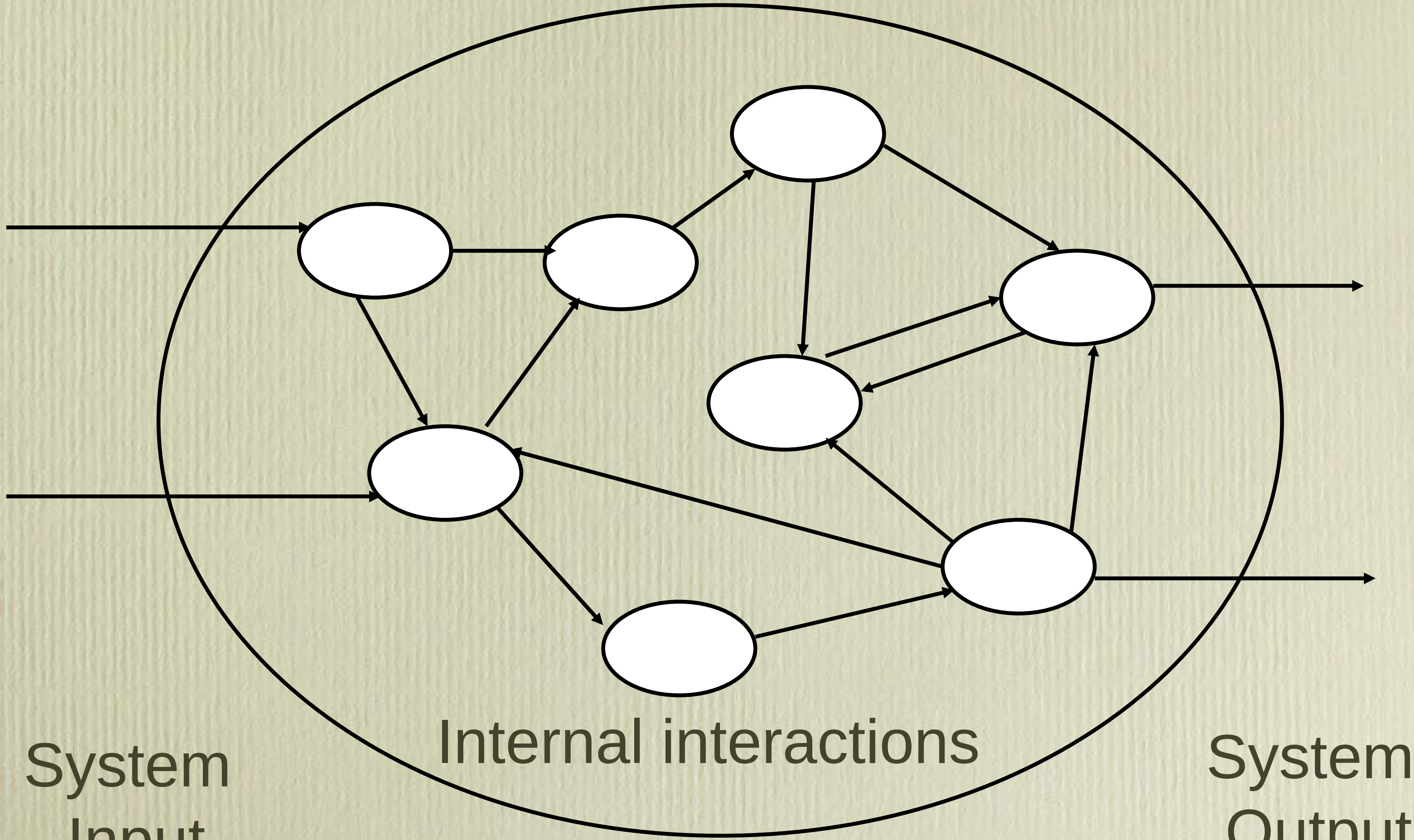
When several agents together perform the function, who should get which reward?

General method: **backpropagation**

- Last agent to deliver solution gets reward
- That agent passes on part of the reward to previous agent(s)
- Which in turn pass on reward to their suppliers, etc.

Used in

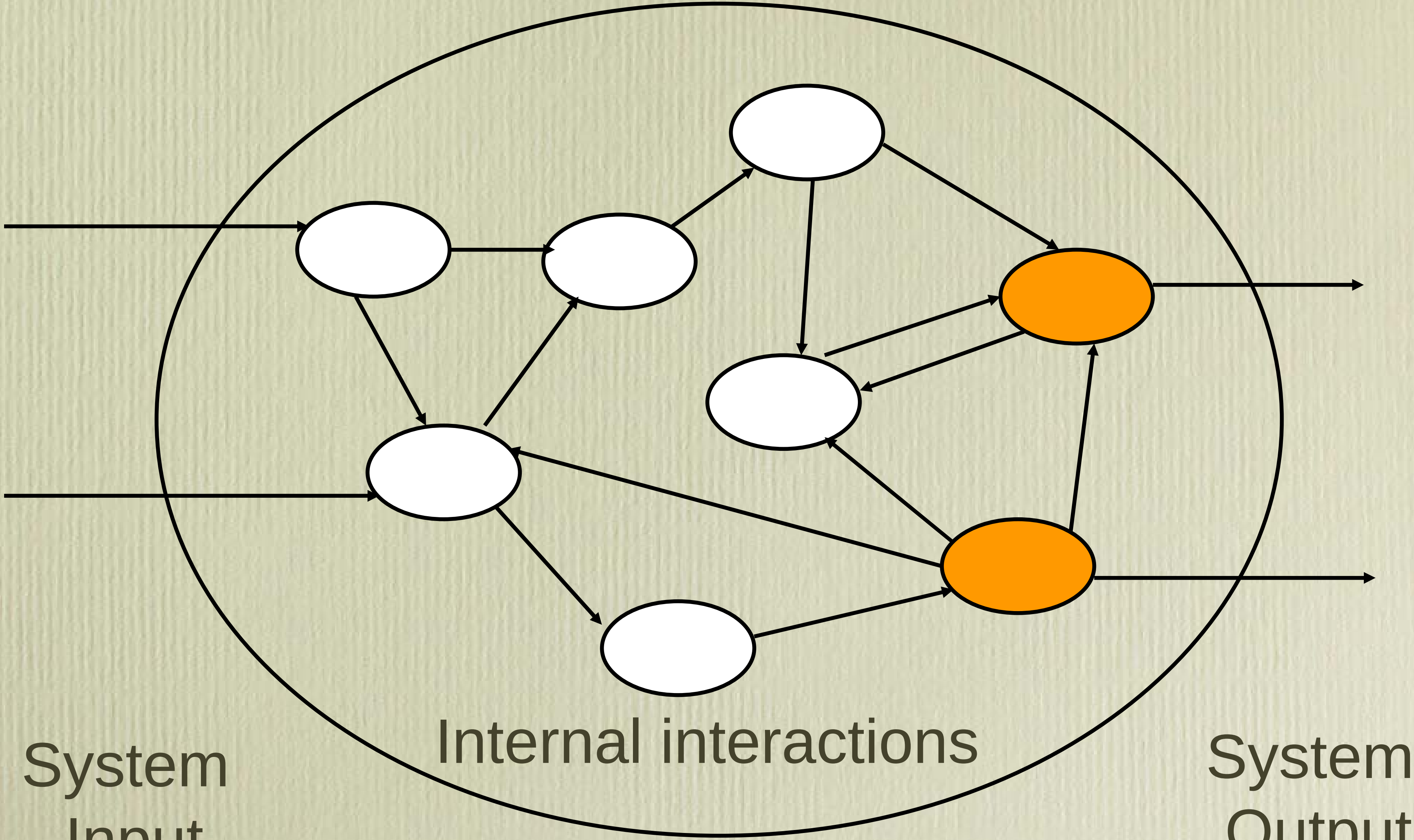
- neural networks
- Classifier Systems (agents undergoing evolutionary algorithms)
- Jack-in-the-Net SO agent service



Internal interactions

System Input

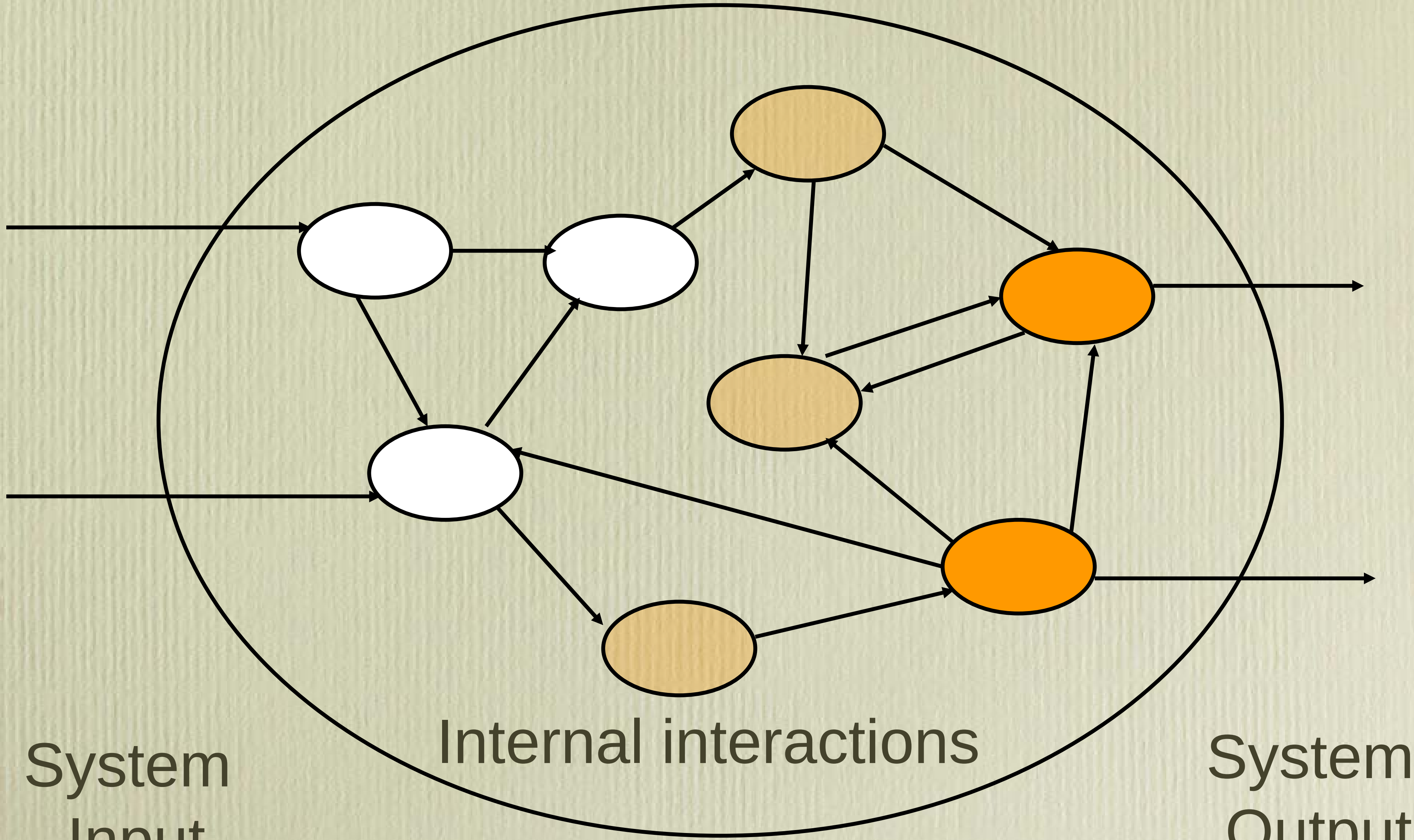
System Output



Internal interactions

System Input

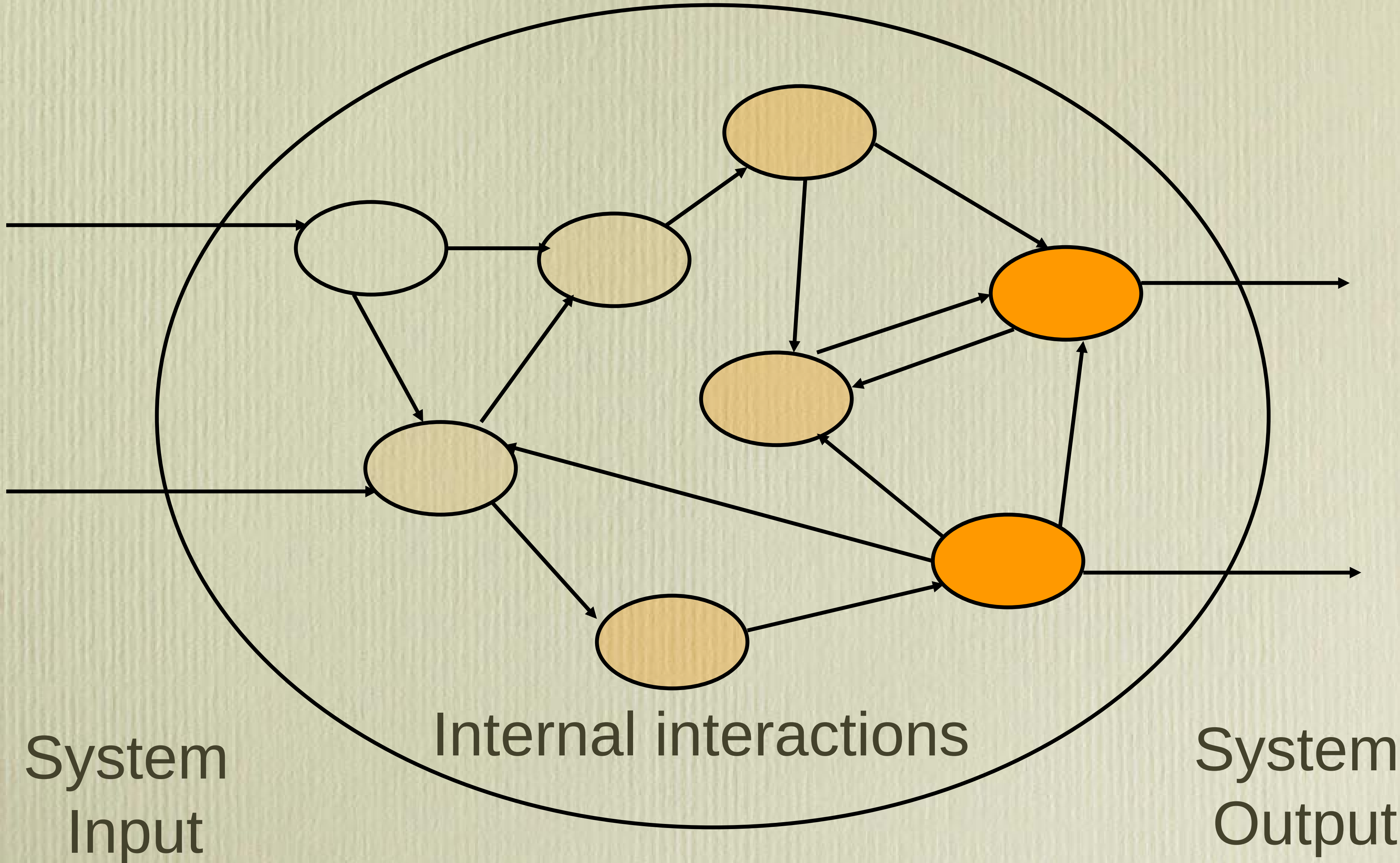
System Output



Internal interactions

System Input

System Output



Internal interactions

System Input

System Output

Problem of Coordination

Credit allocation is non-trivial when “geometry” is variable

- Different agent perform different functions at different times and places

“Intelligent” self-organization implies specialization/division of labor among agents

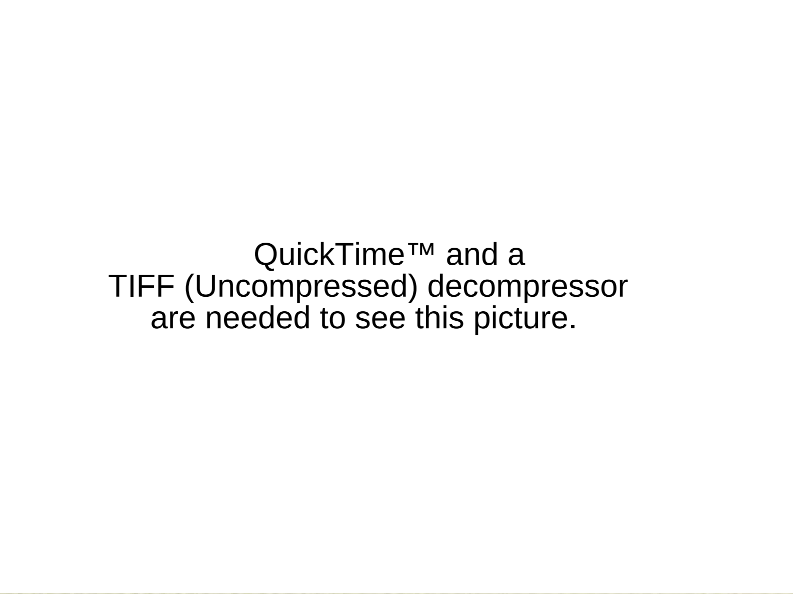
- Different agents have different abilities / experiences
- Together they can be smarter than alone

But how do we achieve such collective intelligence?

Collective Intelligence

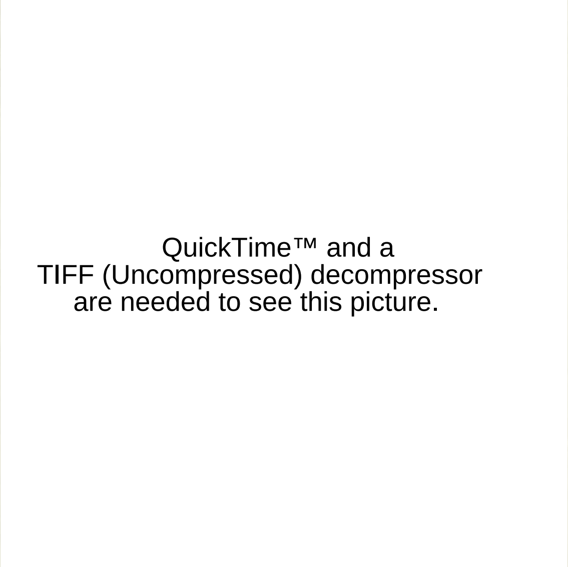
Together we can be more intelligent than individually

- E.g. ants, bees, termites
- group can solve more problems than its members


QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

James Surowiecki:

- "The Wisdom of Crowds”
- Summary of requirements for collective intelligence


QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Requirements for Collective Intelligence

Diversity

- the more varied the individual expertise, the more comprehensive the collective knowledge

Aggregation

- individual contributions must be synthesized into global solution

Independence

- no individuals or subgroups should be able to bias the result

Decentralization (distribution)

- Different individuals should be able to work on different parts of the problem

Intelligent SOS

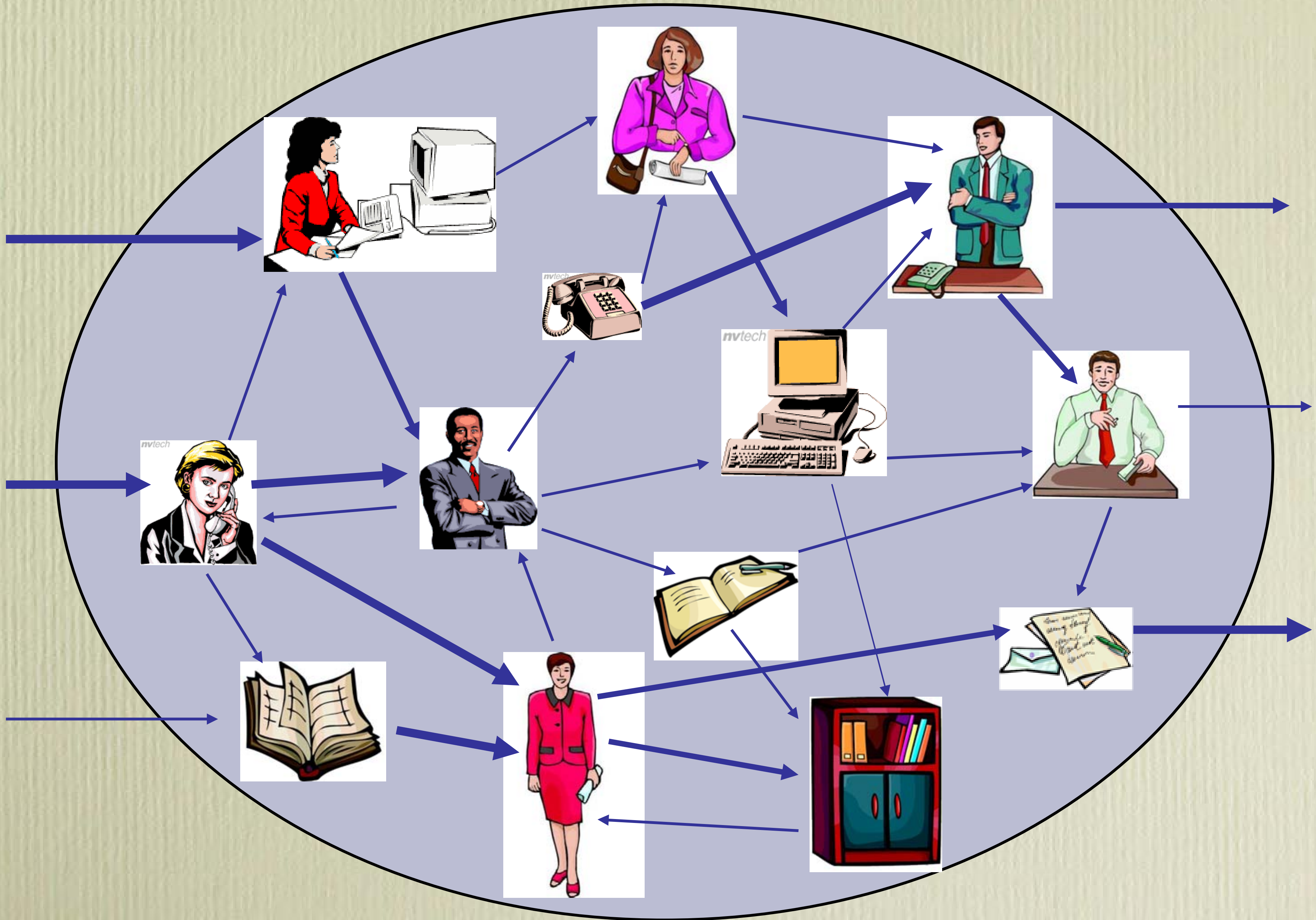
Emergence of a "superagent"

- coordinated, goal-directed collective
- characterized by *collective intelligence*

Distributed cognition

- different agents contribute different results at different times and places
- results are integrated
- together, they solve the global problem

Distributed Cognition



Distributed Cognition

Complex problems must be decomposed into a network of subproblems

- With each subproblem tackled by the appropriate agent
- The different contributions must be integrated into a coherent whole

This requires coordinated action

- *Alignment*
- *Division of labor*
- *Workflow*
- *Aggregation*

Alignment

Simplest case of coordination

Agents should not obstruct or hinder each other

They should act with a common “intention”

- work towards the same goal
- Speak same “language”
- Refer to the same things

Functions of Alignment

Development of shared standards

- agents must agree about doing things the same way
- so as to avoid confusion, friction or conflict

Examples

- *Synchronization*: agreeing about the time to do something
- driving on the same side of the road
- using compatible technological formats

SO of Alignment

Via trial-and-error two agents discover how to act synergetically:

- 1-1 alignment

Basic mechanism

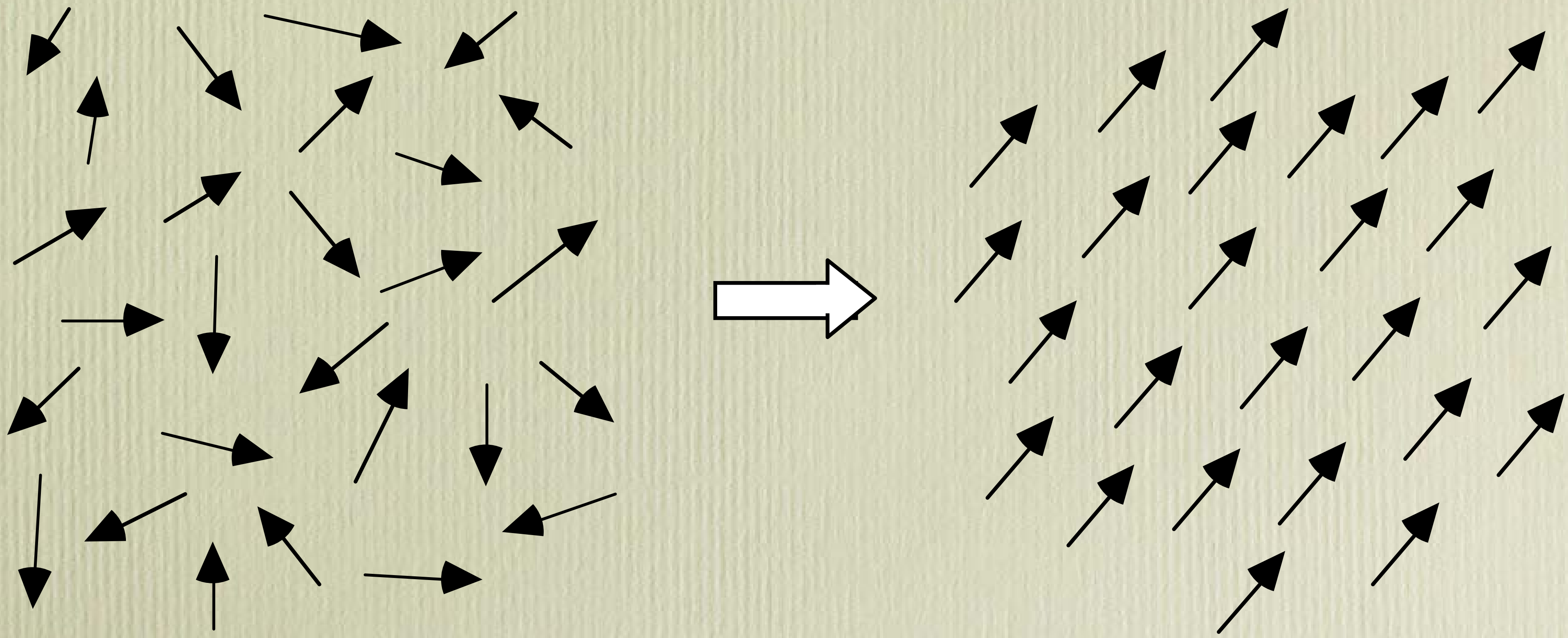
- reinforcement of synergetic interactions
- suppression of frictional interactions

Self-reinforcing growth of the synergetic assembly

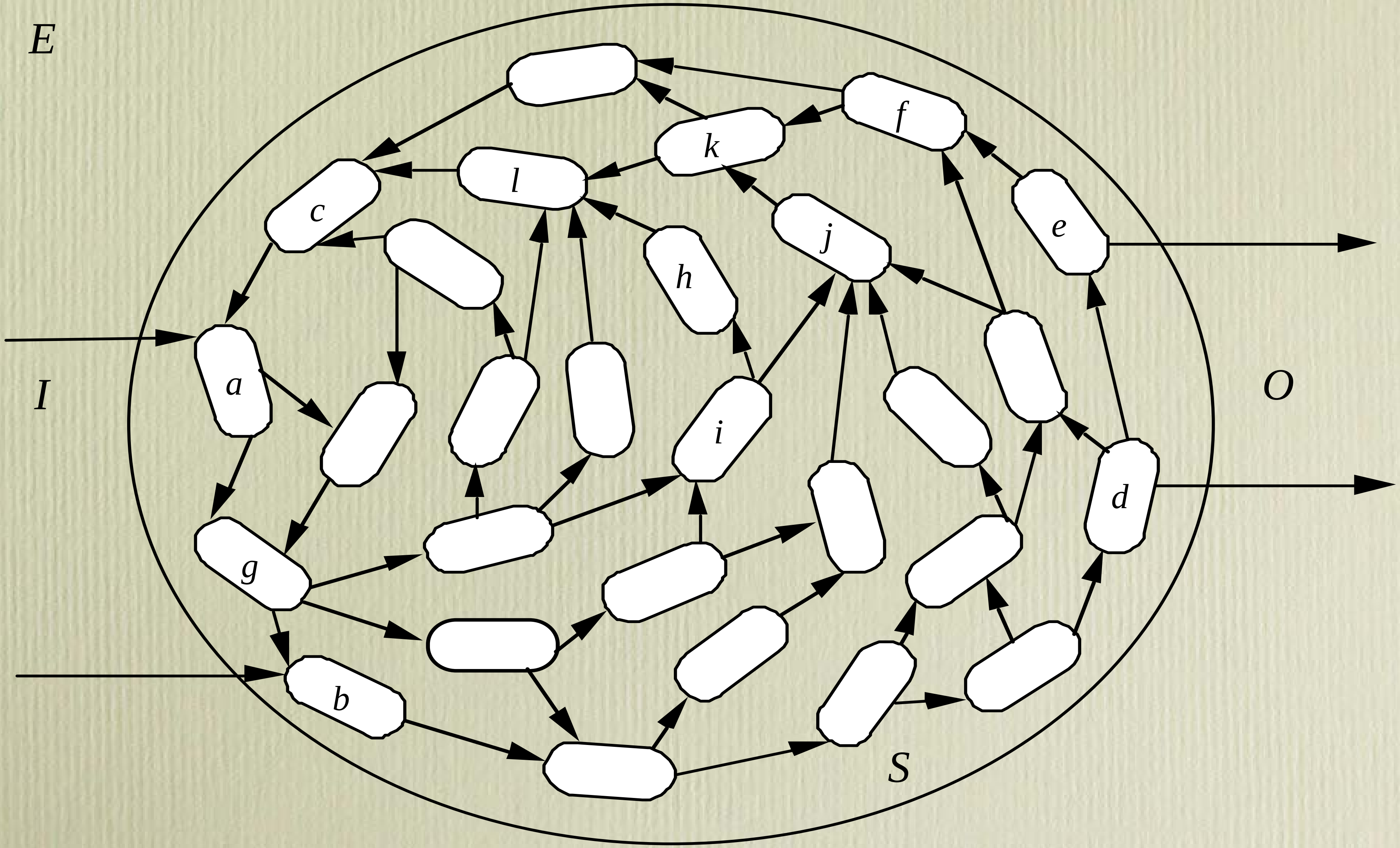
- speeds up process

Eventually, the whole system become “aligned” or “coordinated”

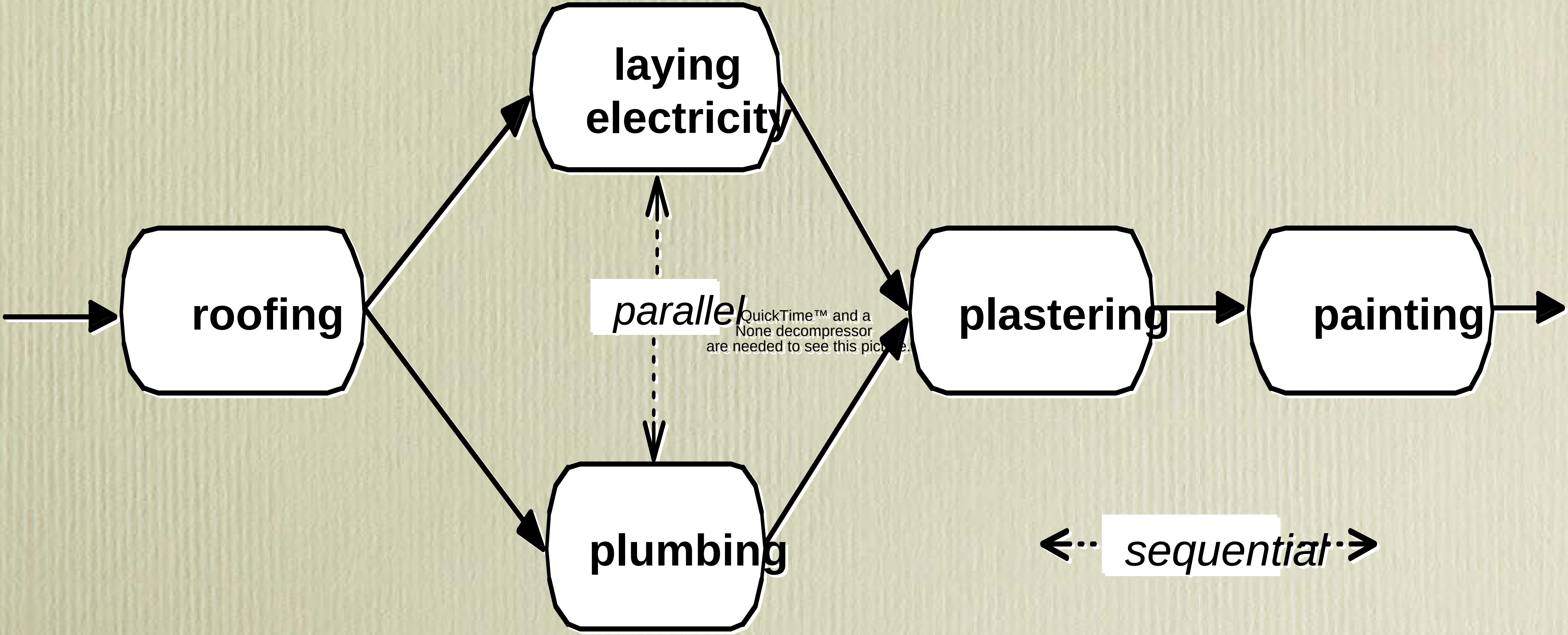
Example: Magnetization



System as Network of Agents

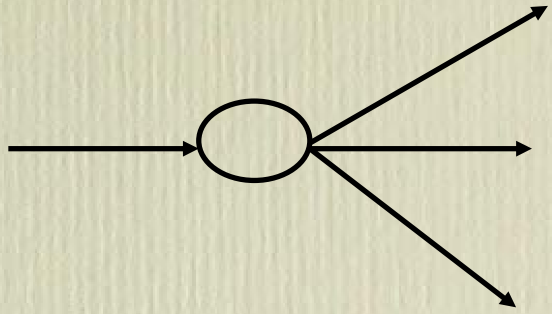


Parallel and Sequential Coordination

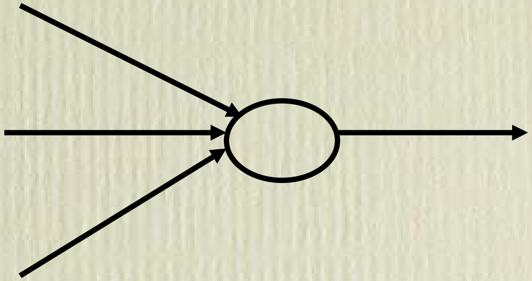


Parallel coordination

Input parallel = division of labor



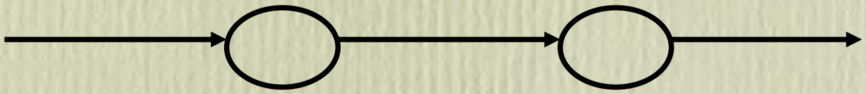
- who does what?
- Different parts of the task allocated to different agents
- allows specialization



Output parallel = aggregation of results

- How are the different results put together again?

Sequential coordination



Ordering of tasks

- second task can only be started when first is finished
- E.g. plasterer -> painter
- together they can do things they cannot do in separation

Workflow

- who does what when?
- How does the “work” move from agent to agent?
- partially solved problem must be passed on to next in line

SO coordination mechanisms

Stigmergy

- Agents deposit partially finished work in shared medium
- Agents pick up tasks from medium

Hebbian learning

- Agents interact preferentially with frequent partner
- Creates links between agents -> “social” network

Backpropagation

- Agents interact with the one that rewarded them

Stigmergy

Division of Labor

- Each agent picks up task it can do best

Workflow

- Partially completed tasks are put back in the medium to be picked up by next agent

Aggregation

- Medium accumulates all results

Examples:

- Wikipedia, ant pheromone traces

Hebbian Learning

Division of Labor

- May require lateral inhibition
    - Active agent suppresses activity in parallel agents
    - Thus forcing them to differentiate in specialization

Workflow

- Agents order themselves in a sequence of activity

Aggregation

- Output agents aggregate output they receive from their input agents

Backpropagation

Division of Labor

- Agents that successfully performed a task will get more of these tasks delegated to them

Workflow

- Agents learn to recognize who is next in line?

Aggregation

- Like in Hebbian learning

Integrated coordination mechanisms

Classifier systems (Holland):

- Stigmergy + Backprop

Jack-in-the-Net agent architecture (Itao et al.):

- Hebbian + Backprop

Ant algorithms:

- Stigmergy + Hebbian

Ideal system?:

- Stigmergy + Hebbian + Backprop?
- Formally represented by hypernetworks?