Biologically-inspired Distributed Computing

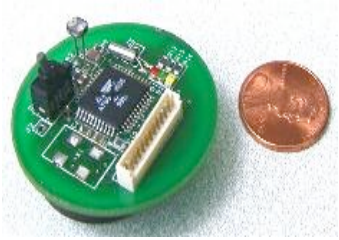
Radhika Nagpal
Harvard University, Computer Science
(Harvard Medical School, Systems Biology)

Robust Collective Behavior

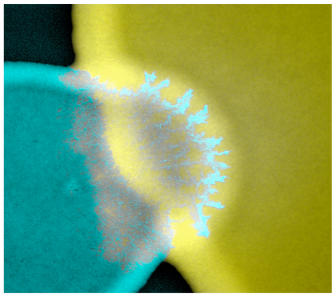
Design systems capable of self-organization, self-repair

Distributed Systems

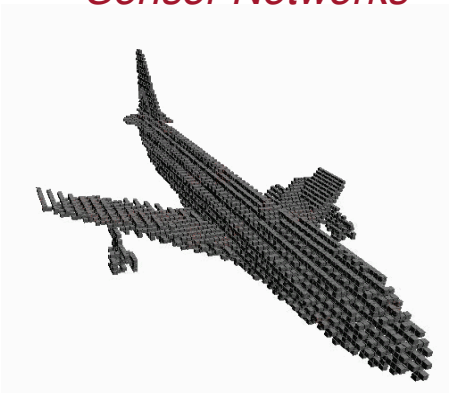
Swarm robotics



Sensor Networks

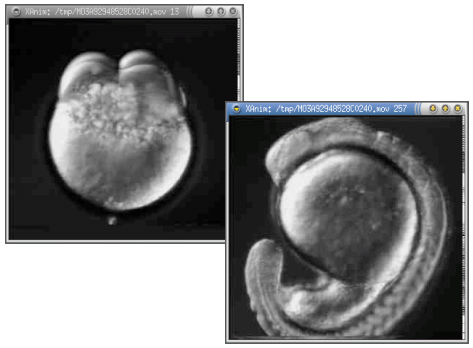


Synthetic Biology



Reconfigurable Robots

Inspiration: Biological Systems

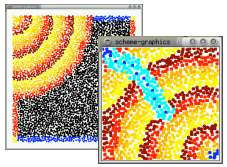


Programming Myriads

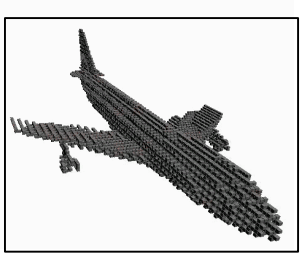
High-level Programming Languages
in terms of Goals / Tasks

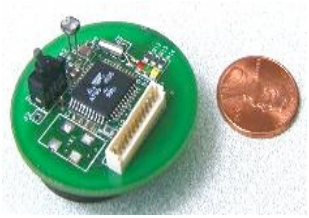
Global-to-Local COMPILER

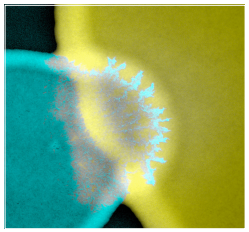
Biologically-inspired primitives for Robust Collective Behavior



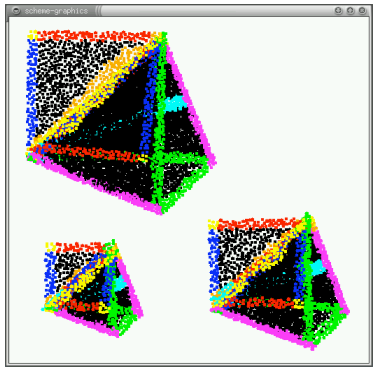
Program for Myriads of Agents

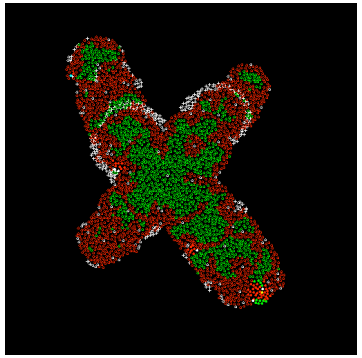


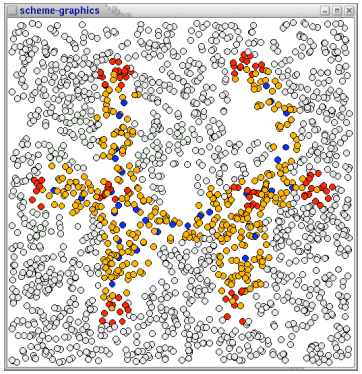


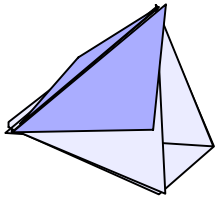


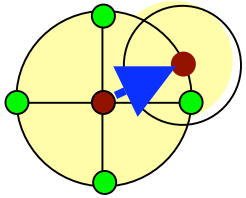
Languages for Pattern/Self-Assembly

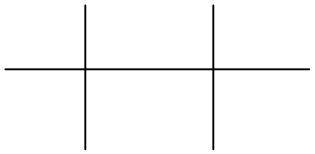










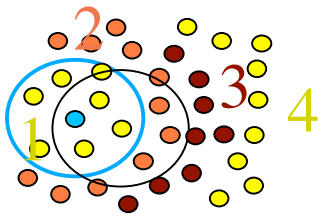


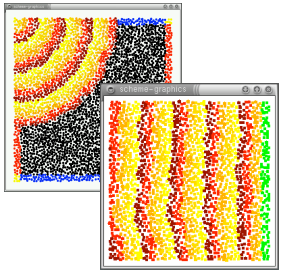
Grammars for "constructing" pattern

COMPILER

Behavior of an individual element

Primitives for local behavior





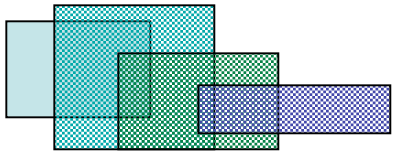
Lessons

- Traditional advantages of compilation
  - Easily program structure at an abstract level
  - Optimize and reason at the global level
  - Conservation of low-level code (and effort)
  - Complexity proportional to structure, not the number of nodes.

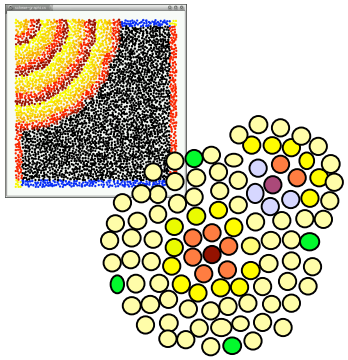
- Robust
  - Small number of robust local primitives
  - Robust to random placement, node failure and message loss, asynchronous nodes
  - Best-effort (but theoretically analyzable)

Example: Programmable Self-Assembly

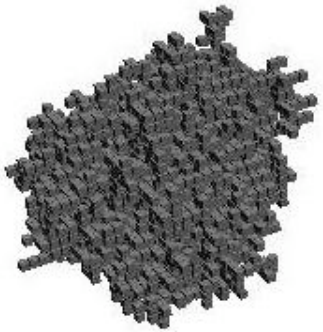
Goal Shape:
described using
block-construction representation

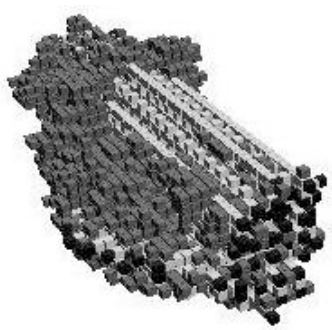


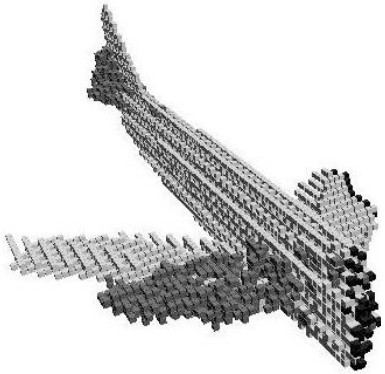
COMPILED

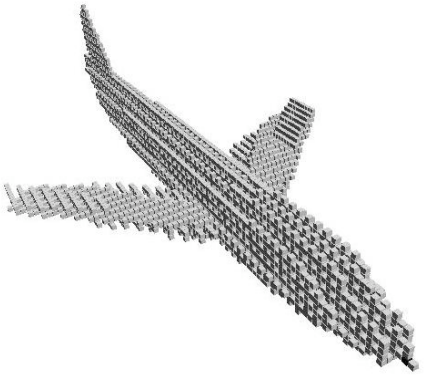


Program run by
identically-programmed mobile modules,
based on biologically-inspired primitives

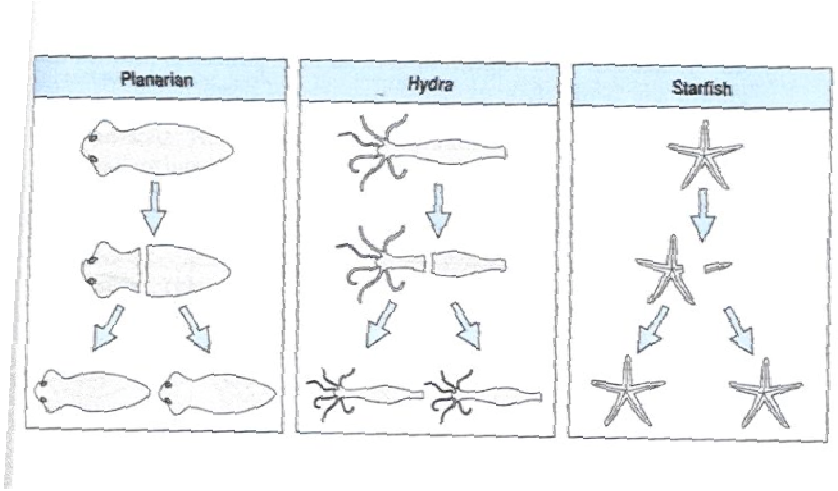


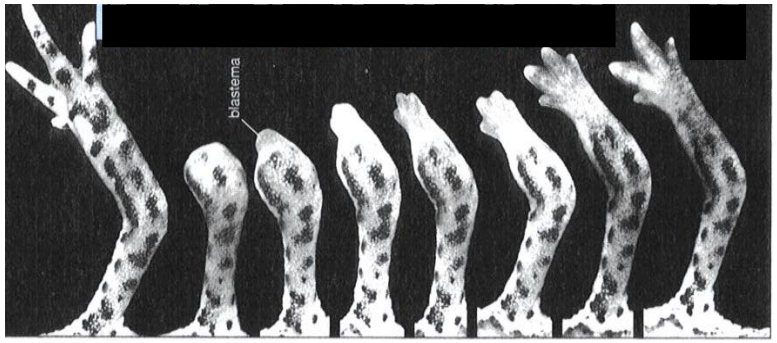




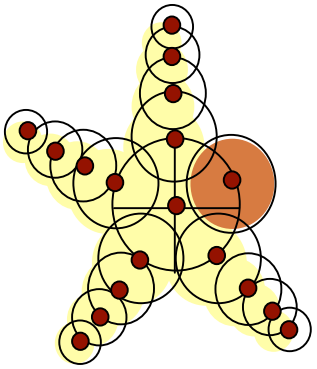


Self-Repair and Regeneration



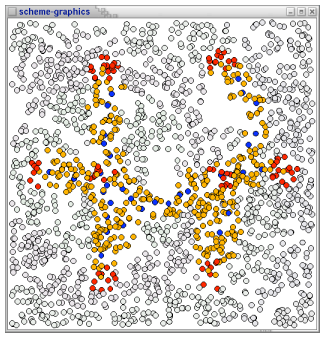


Regenerating structures



Absence of neighbor causes circle to recreate its neighbor, which in turn recreates its neighbor - thus regenerating the broken structure

Self-repairing patterns



If a line is broken, one part dies off and the other regrows

Research Agenda

- How do we obtain a *robust behavior* from the cooperation of vast numbers of unreliable parts?
- How do we engineer *pre-specified global* behavior from local interactions?

- Organizational principles
- Algorithms
  - Design and analysis
- Programming models and languages
  - Catalogues of primitives
  - Means of combination
  - Means of abstraction
  - Compilation technology targeted to appropriate substrates