Second Annual French Complex Systems Summer School
Institut des Systèmes Complexes, Lyon & Paris, July 15-August 9, 2008

Complex Systems Made Simple:

A Hands-on Exploration of Agent-Based Modeling

René Doursat, Ph.D.

http://doursat.free.fr

Instructor

René Doursat

Experience

- Guest Researcher, CNRS (ISC) & Ecole Polytechnique (CREA), 2006-Today
- Visiting Assistant Professor, University of Nevada, Reno, 2004-2006
- *Senior Software Engineer & Architect, Paris and San Francisco, 1995-2004*
- Research Associate, Ecole Polytechnique (CREA), Paris, 1996-1997
- Postdoctoral Fellow, Ruhr-Universität Bochum, Germany, 1991-1995

Education

- Ph.D. in applied math (computational neuroscience), Université Paris VI, 1991
- M.S. in physics, Ecole Normale Supérieure, Paris, 1987

Research interests

- computational modeling and simulation of complex systems, especially neural, biological and social, which can foster novel principles and applications in ICT
- self-organization of *reproducible* and *programmable* structures in (a) large-scale spiking neural dynamics, (b) developmental artificial life, (c) multi-agent networks

Course Info

Information

- Course schedule
  - Wednesday, July 16: 14:00-15:45 16:15-18:00
  - Thursday, July 17: 14:00-15:45 16:15-18:00
  - Friday, July 18: 14:00-15:45 16:15-18:00
- Summer school & course Web page
  - http://iscpif.csregistry.org/Summer+School+2008

    then click on *Complex systems made simple*
- Personal Web page
  - http://doursat.free.fr
- E-mail
  - rene.doursat@iscpif.fr

Course Contents

- **What this course is about** (dense preview, will be repeated)
  - ✓ an *exploration* of various complex systems *objects* (i.e., made of many agents, with simple or complex rules, and complex behavior):
    - cellular automata, pattern formation, swarm intelligence, complex networks, spatial communities, structured morphogenesis
  - ✓ and their common *questions*:
    - emergence, self-organization, positive feedback, decentralization, between simple and disordered, “more is different”, adaptation & evolution
  - ✓ by interactive *experimentation* (using NetLogo),
  - ✓ introducing *practical* complex systems *modeling* and simulation
  - ✓ from a *computational* viewpoint, as opposed to a “mathematical” one (i.e., formal or numerical resolution of symbolic equations),
  - ✓ based on discrete *agents* moving in discrete or quasi-continuous space, and *interacting* with each other and their environment

Complex Systems Made Simple

1. Introduction

2. A Complex Systems Sampler

3. Commonalities

4. NetLogo Tutorial

Complex Systems Made Simple

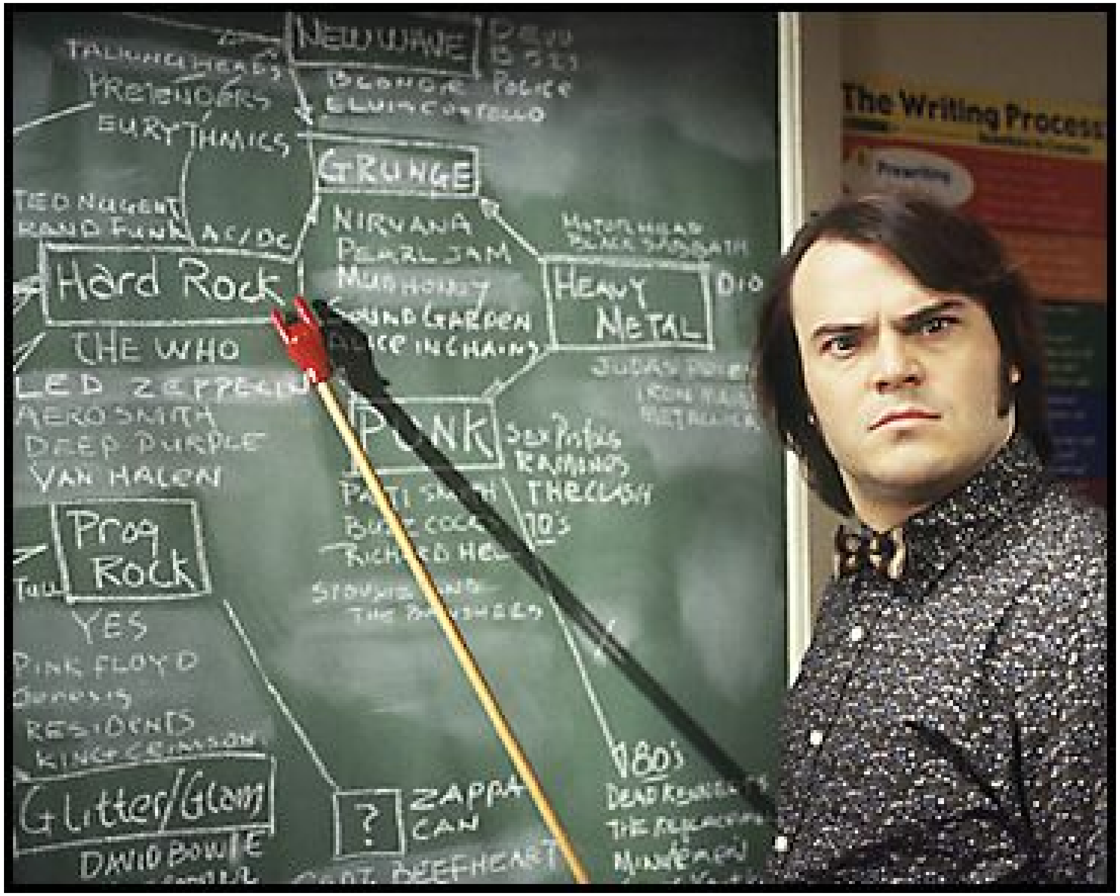
1. Introduction
    a. What are complex systems?
    b. A vast archipelago
    c. Computational modeling
2. A Complex Systems Sampler
3. Commonalities
4. NetLogo Tutorial

Complex Systems Made Simple

1. Introduction
   a. What are complex systems?
      - *Few agents*
      - *Many agents*
      - *CS in this course*
   b. A vast archipelago
   c. Computational modeling
2. A Complex Systems Sampler
3. Commonalities
4. NetLogo Tutorial

1. Introduction — a. What are complex systems?

➢ Any ideas?



The School of Rock (2003)
Jack Black, Paramount Pictures

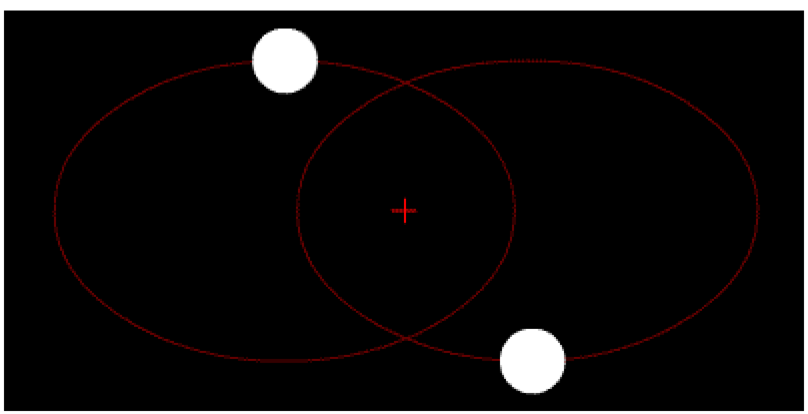
1. Introduction — a. What are complex systems?

➢ Few agents, simple emergent behavior

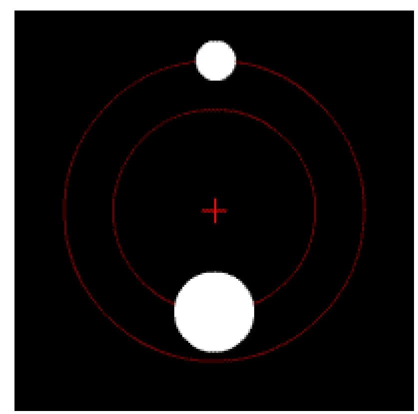
→ *ex: two-body problem*

✓ fully solvable and ***regular*** trajectories for inverse-square force laws (e.g., gravitational or electrostatic)

$$\begin{cases} \mathbf{F}_{12}(\mathbf{x}_1, \mathbf{x}_2) = m_1 \ddot{\mathbf{x}}_1 & (Equation\ 1) \\ \mathbf{F}_{21}(\mathbf{x}_1, \mathbf{x}_2) = m_2 \ddot{\mathbf{x}}_2 & (Equation\ 2) \end{cases}$$



Two bodies with similar mass
Wikimedia Commons

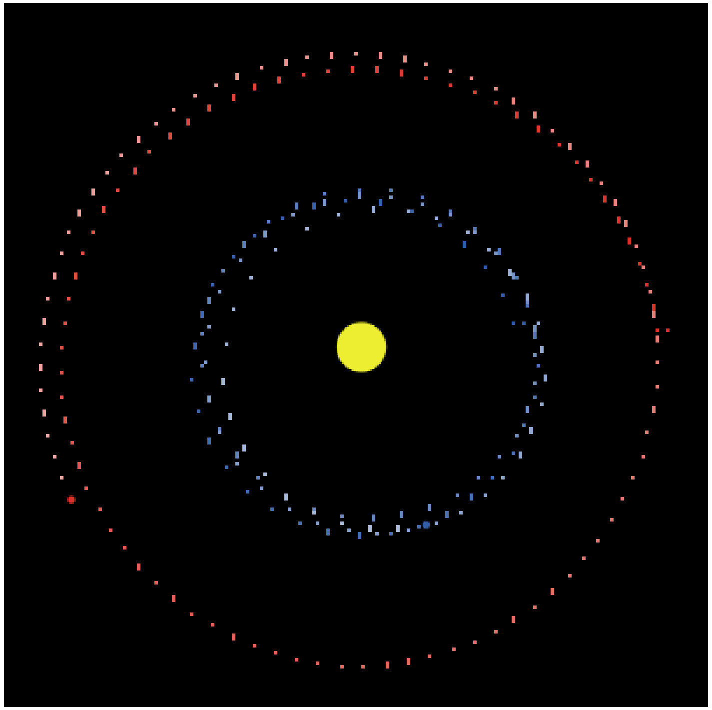


Two bodies with different mass
Wikimedia Commons

1. Introduction — a. What are complex systems?

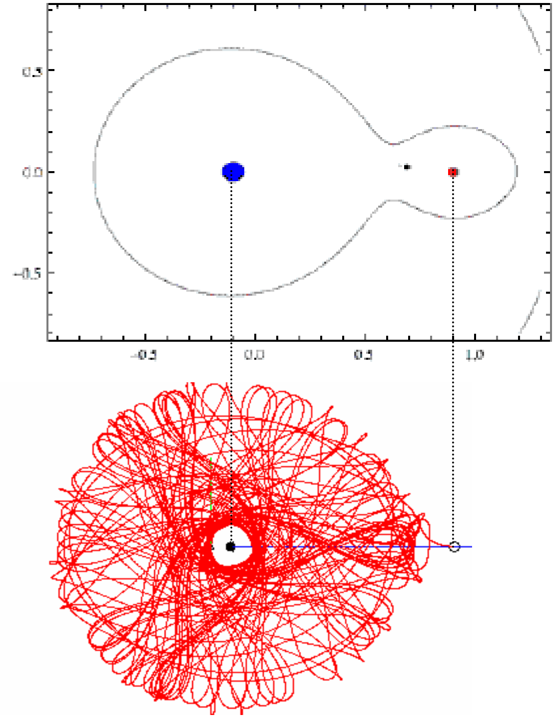
- **Few agents, complex emergent behavior**
  - → *ex: three-body problem*
  - ✓ generally no exact mathematical solution (even in "restricted" case $m_1 \langle\langle\ m_2 \approx m_3$): must be solved numerically → ***chaotic*** trajectories

NetLogo model: /Chemistry & Physics/Mechanics/Unverified



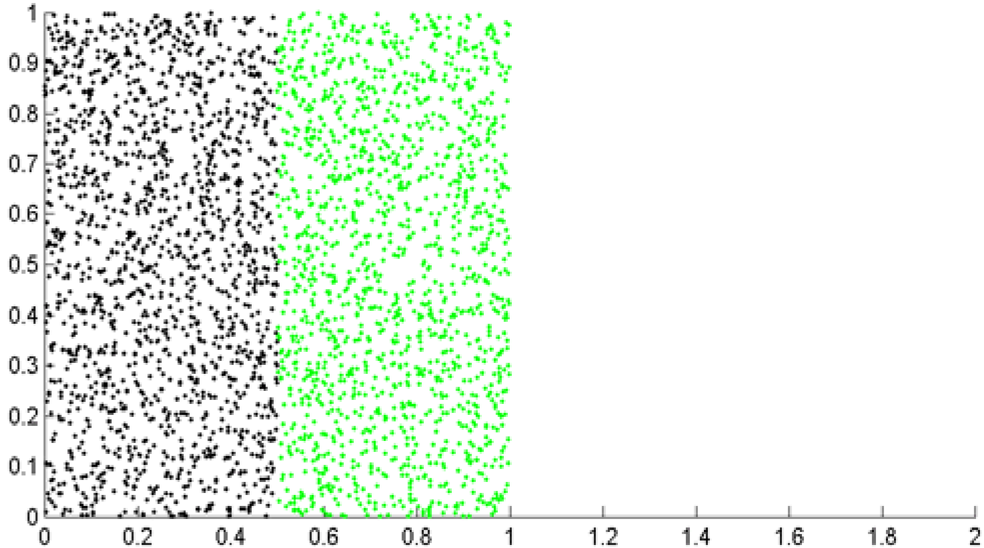
Transit orbit of the planar circular restricted problem

Scholarpedia: Three Body Problem & Joachim Köppen Kiel's applet



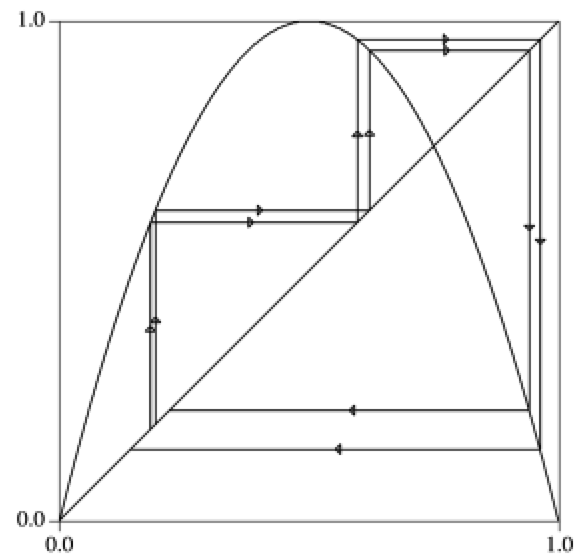
1. Introduction — a. What are complex systems?

- **Few agents, complex emergent behavior**
  - → *ex: more chaos (baker's/horseshoe maps, logistic map, etc.)*
  - ✓ ***chaos*** generally means a ***bounded***, ***deterministic*** process that is ***aperiodic*** and ***sensitive on initial conditions*** → small fluctuations create large variations ("butterfly effect")
  - ✓ even one-variable iterative functions: $x_{n+1} = f(x_n)$ can be "complex"



Baker's transformation

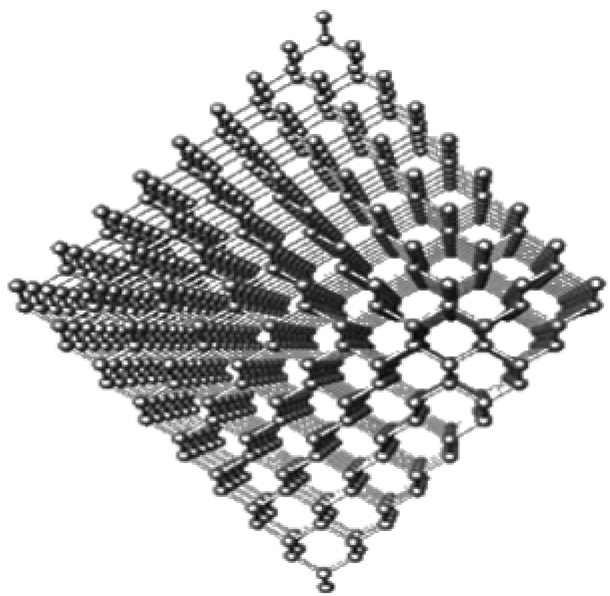
Craig L. Zirbel, Bowling Green State University, OH



Logistic map

1. Introduction — a. What are complex systems?

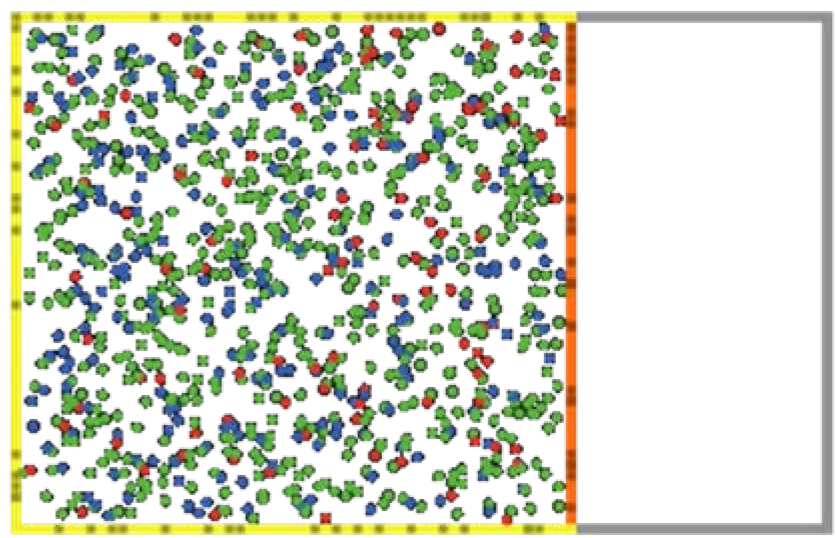
- **Many agents, simple rules, simple emergent behavior**
  - → *ex: crystal and gas (covalent bonds or electrostatic forces)*
  - ✓ either highly ordered, ***regular*** states (crystal)
  - ✓ or disordered, random, statistically ***homogeneous*** states (gas): a few global variables (P, V, T) suffice to describe the system



Diamond crystal structure

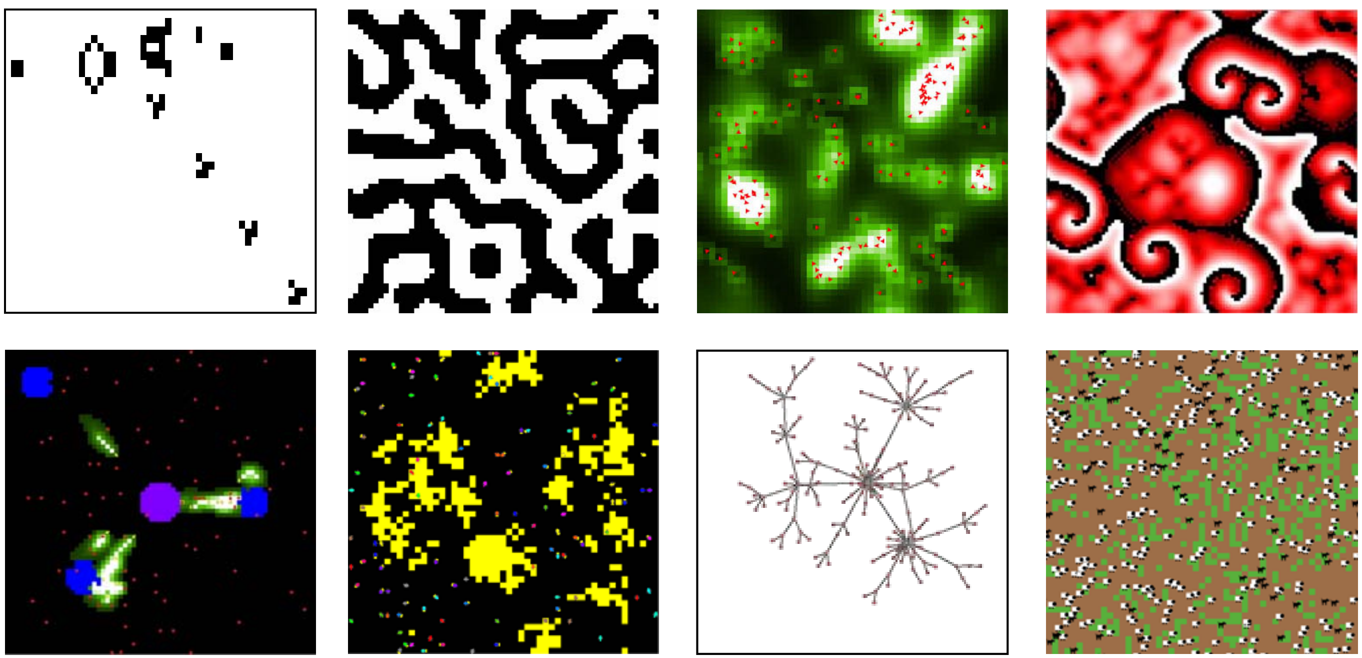
Tonci Balic-Zunic, University of Copenhagen

NetLogo model: /Chemistry & Physics/GasLab Isothermal Piston



1. Introduction — a. What are complex systems?

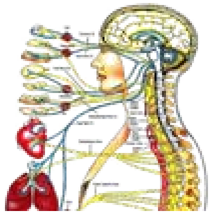
- **Many agents, simple rules, complex emergent behavior**
  - → *ex: cellular automata, pattern formation, swarm intelligence (insect colonies, neural networks), complex networks, spatial communities*
  - ✓ the "clichés" of complex systems: a major part of this course and NetLogo models



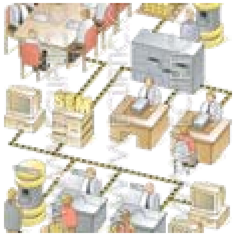
1. Introduction — a. What are complex systems?

- **Many agents, complex rules, complex emergent behavior**
  - → *natural ex: organisms (cells), societies (individuals + techniques)*
  - ✓ agent rules become more “sophisticated”, e.g., ***heterogeneous*** depending on the element’s ***type*** and/or ***position*** in the system
  - ✓ behavior is also complex but, paradoxically, can become more ***controlled***, e.g., ***reproducible*** and ***programmable***

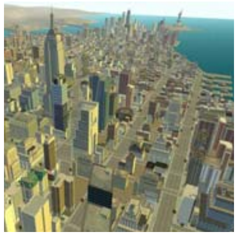



termite mounds


companies


techno-networks


cities

1. Introduction — a. What are complex systems?

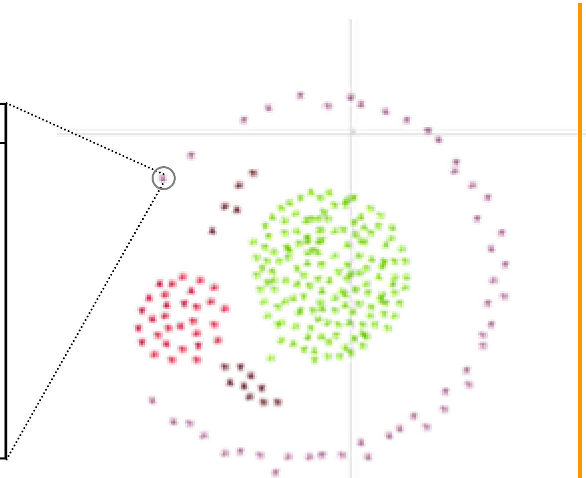
➢ Many agents, complex rules, complex emergent behavior

→ *ex: self-organized "artificial life": swarm chemistry, morphogenesis*

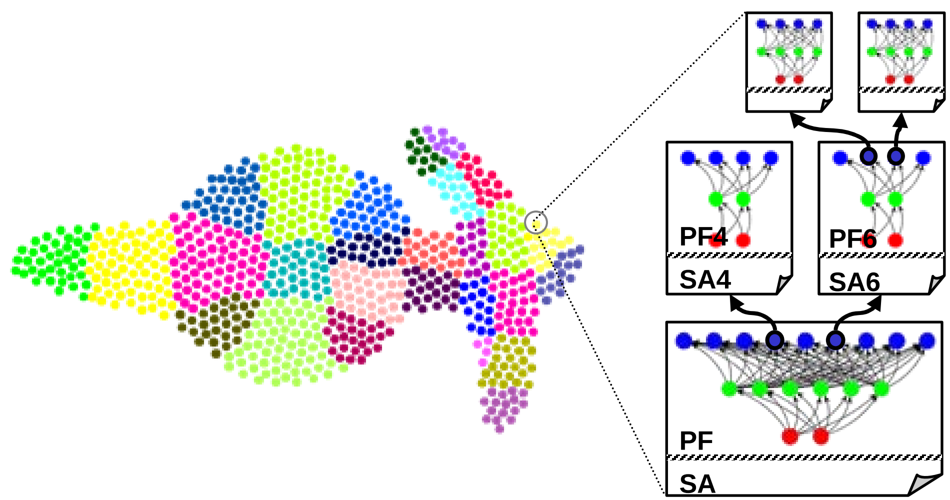
✓ in swarm chemistry (Sayama 2007), mixed self-propelled particles with different flocking parameters create nontrivial formations

✓ in embryomorphism (Doursat 2006), cells contain the same genetic program, but differentiate while self-assembling into specific shapes

Name	Min	Max
R^i	0	300
V_n^i	0	20
V_m^i	0	40
c_1^i	0	1
c_2^i	0	1
c_3^i	0	100
c_4^i	0	0.5
c_5^i	0	1



Swarm chemistry
Hiroki Sayama, Binghamton University SUNY



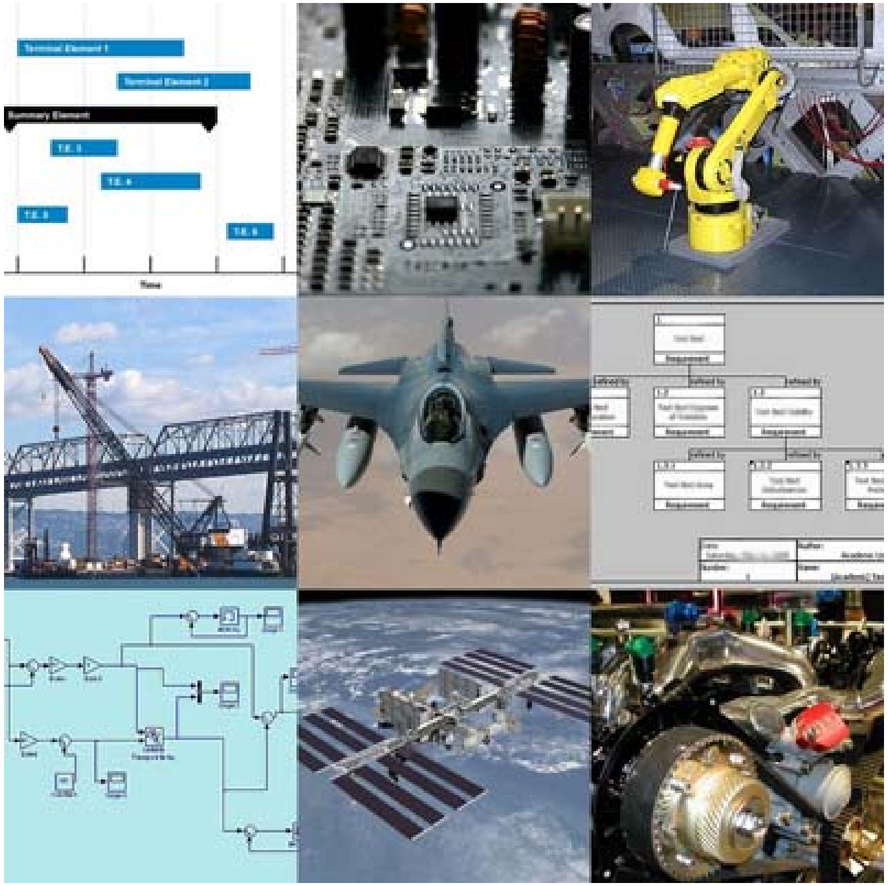
Embryomorphic engineering
René Doursat, Insitut des Systèmes Complexes, Paris

1. Introduction — a. What are complex systems?

- Many agents, complex rules, “simple” emergent behavior
  - → *human ex: crowds, orchestras, armies*
  - ✓ humans reacting similarly and/or simultaneously to a complicated set of stimuli coming from a *centralized* leader, plan or event
  - → ***absence of self-organization*** *(or only little)*

1. Introduction — a. What are complex systems?

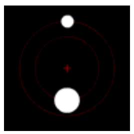
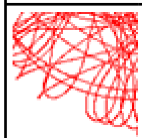
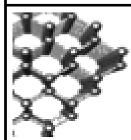
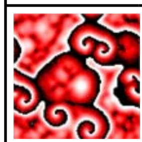
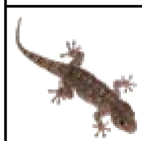
- **Many agents, complex rules, "simple" emergent behavior**
  - → *artificial ex: electronics, machines, aircrafts, civil constructions*
  - ✓ complicated, multi-part devices designed by engineers to behave in a limited and ***predictable*** (reliable, controllable) number of ways "I don't want my airplane to be creatively emergent"
  - → ***absence of self-organization*** (components do not assemble or evolve by themselves)



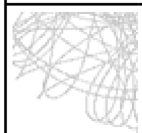
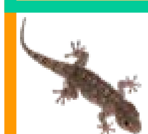
Systems engineering
Wikimedia Commons

1. Introduction — a. What are complex systems?

➢ Recap: complex systems in this course

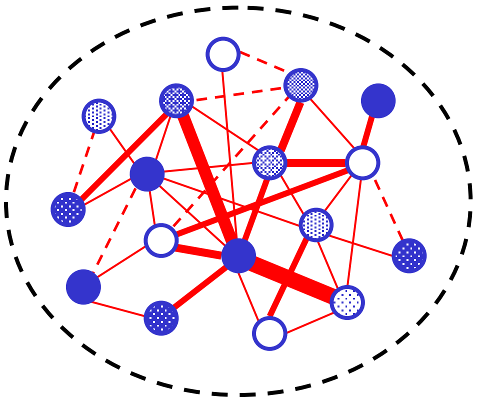
Example	Agents	Rules	Emergent Behavior	A "Complex System"?
 two-body	*few*	*simple*	*simple*	*NO*
 chaos	*few*	*simple*	*complex*	*NOT HERE*
 crystal, gas	*many*	*simple*	*simple*	*NO*
 patterns, swarms, colonies, networks	*many*	*simple*	*complex*	*YES*
 structured morphogenesis	*many*	*complex*	*complex*	*YES*
 crowds, devices	*many*	*complex*	*"simple"*	*NOT HERE*

➢ Recap: complex systems in this course

Example	*Agents*	*Rules*	*Emergent Behavior*	*A "complex system"?*
two-body	*few*	*simple*	*simple*	*NO*
 chaos	*few*	*simple*	*complex*	*NOT HERE*
 crystal, gas	*many*	*simple*	*simple*	*NO*
 patterns, swarms, colonies, networks	*many*	*simple*	*complex*	*YES*
 structured morphogenesis	*many*	*complex*	*complex*	*YES*
 crowds, devices	*many*	*complex*	*"simple"*	*NOT HERE*

1. Introduction — a. What are complex systems?

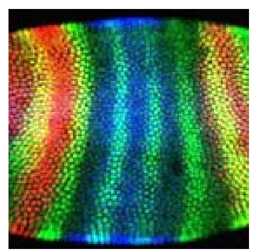
➢ Complex systems in this course



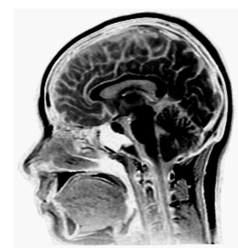
- **large** number of elementary **agents** interacting **locally**
- more or less simple individual agent behaviors creating a complex emergent **self-organized** behavior
- **decentralized** dynamics: no master blueprint or grand architect

✓ physical, biological, technical, social systems (natural or artificial)

pattern formation
○ = matter

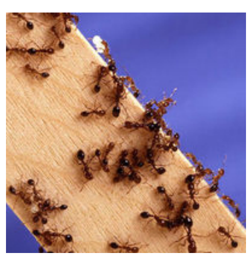


biological development
○ = cell

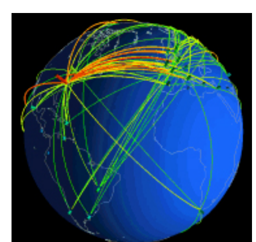


the brain & cognition
○ = neuron

insect colonies
○ = ant



Internet & Web
○ = host/page



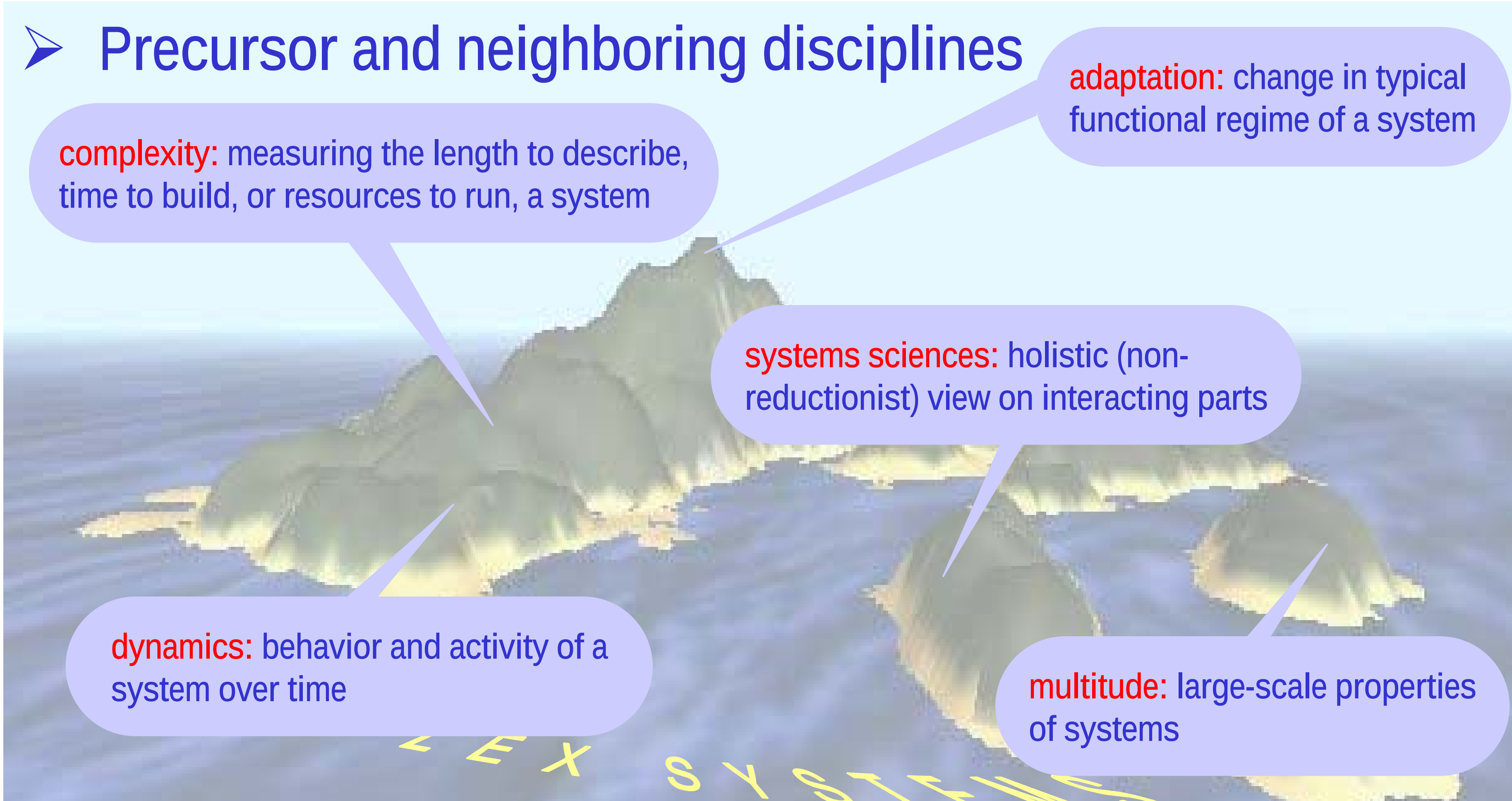
social networks
○ = person

Complex Systems Made Simple

1. Introduction
    a. What are complex systems?
    b. A vast archipelago
        - *Related disciplines*
        - *Big questions × big objects*
        - *Exporting CS to ICT*
    c. Computational modeling
2. A Complex Systems Sampler
3. Commonalities
4. NetLogo Tutorial

1. Introduction — b. A vast archipelago

➢ Precursor and neighboring disciplines



- ✓ different families of disciplines *focus* on different aspects
- ✓ naturally, they intersect a lot: don't take this taxonomy too seriously!

1. Introduction — b. A vast archipelago

➢ Precursor and neighboring disciplines

complexity: measuring the length to describe, time to build, or resources to run, a system
- information theory (Shannon; entropy)
- computational complexity (P, NP)
- Turing machines & cellular automata

adaptation: change in typical functional regime of a system
- evolutionary methods
- genetic algorithms
- machine learning

systems sciences: holistic (non-reductionist) view on interacting parts
- systems theory (von Bertalanffy)
- systems engineering (design)
- cybernetics (Wiener; goals & feedback)
- control theory (negative feedback)

dynamics: behavior and activity of a system over time
- nonlinear dynamics & chaos
- stochastic processes
- systems dynamics (macro variables)

multitude: large-scale properties of systems
- graph theory & networks
- statistical physics
- agent-based modeling
- distributed AI systems

1. Introduction — b. A vast archipelago

- Sorry, there is no general “complex systems science” or “complexity theory”...
  - ✓ there are a lot of theories and results in related disciplines (“systems theory”, “computational complexity”, etc.), yet
    - such generic names often come from one researcher with one particular view
    - there is no unified viewpoint on *complex systems, especially autonomous*
    - in fact, there is not even any agreement on their ***definition***
  - ✓ we are currently dealing with an intuitive set of criteria, more or less shared by researchers, but still hard to formalize and quantify:
    - complexity
    - emergence
    - self-organization
    - multitude / decentralization
    - adaptation

... but don’t go packing yet!

1. Introduction — b. A vast archipelago

➢ The French "roadmap" toward complex systems science

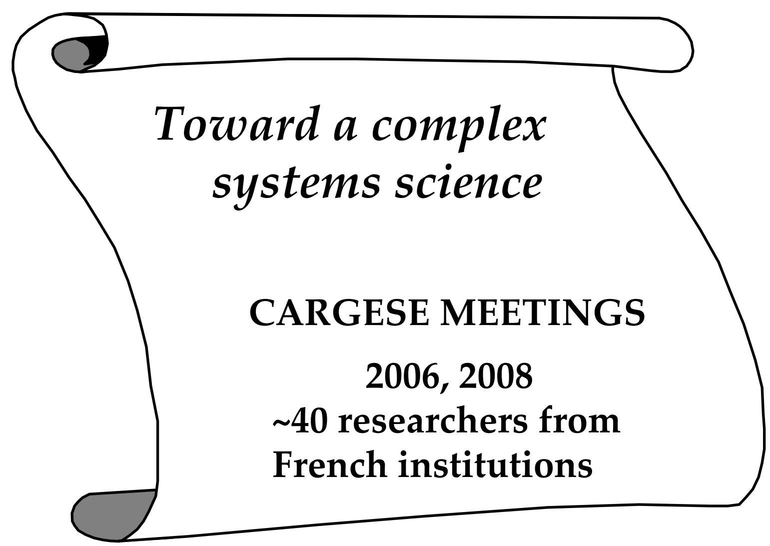
✓ another way to circumscribe complex systems is to list "big (horizontal) questions" and "big (vertical) objects", and cross them

Big objects

- subcellular
- multicellular development
- physiology
- indiv. & social cognition
- ecosystems
- territorial intell. & sustainability
- ubiquitous computing

Big questions

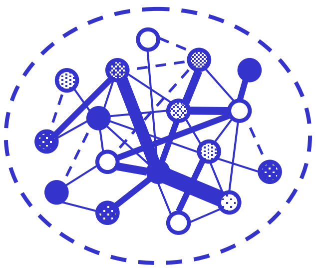
- reconstruct multiscale dynam.
- emergence & immergence
- spatiotemp. morphodynamics
- optimal control & steering
- artificial design
- fluctuations out-of-equilib.
- adaptation, learning, evolution



1. Introduction — b. A vast archipelago

➢ The central challenges of complex systems (CS) research

✓ complex systems pervade nature and human structures: similarities among phenomena can create many cross-disciplinary exchanges

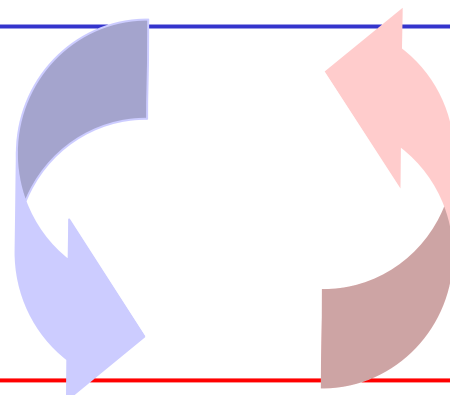


CS science: understand "natural" CS

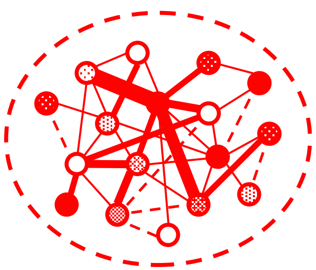
(i.e. spontaneously emergent, including human activity)

my own keen interest

Exports
- decentralization
- autonomy, homeostasis
- learning, evolution



Imports
- observe, model
- control, harness
- design, use



CS engineering: design a new generation of "artificial" CS (i.e. harnessed, including nature)

1. Introduction — b. A vast archipelago

- Exporting natural CS to artificial disciplines, such as ICT

ex: brain	specific natural or societal complex system	*ex: bird flocks*
↓	↓	↓
biological neural models	model simulating this system	*3-D collective flight simulation*
↓	↓	↓
binary neuron, linear synapse	generic principles and mechanisms (schematization, caricature)	*“boid”, separation, alignment, cohesion*
↓	↓	↓
artificial neural networks applied to association and classification	new computational discipline exploiting these principles to solve ICT problems	*particle swarm optimization (PSO) “flying over” solutions in high-D spaces*

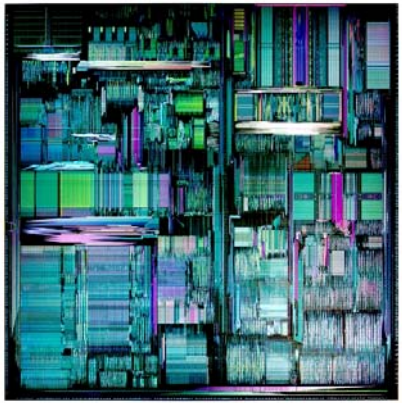
1. Introduction — b. A vast archipelago

➢ Exporting natural CS to artificial disciplines, such as ICT

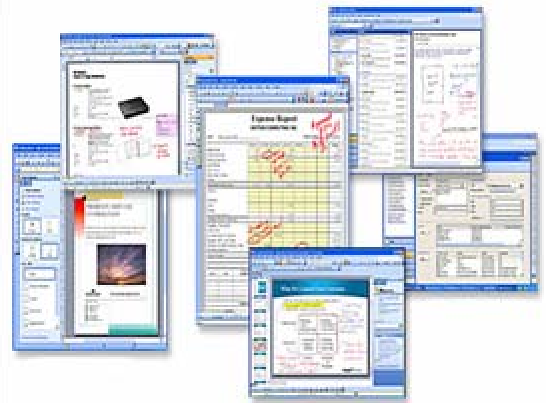
ex: genes & evolution	specific natural or societal complex system	*ex: ant colonies*
↓	↓	↓
laws of genetics	model simulating this system	*trail formation, swarming*
↓	↓	↓
genetic program, binary code, mutation	generic principles and mechanisms (schematization, caricature)	*agents that move, deposit & follow "pheromone"*
↓	↓	↓
genetic algorithms, evolutionary programming for search & optimization	new computational discipline exploiting these principles to solve ICT problems	*ant colony optimization (ACO) applied to graph theoretic & networking problems*

1. Introduction — b. A vast archipelago

- The rapid growth in size & complexity of ICT systems,

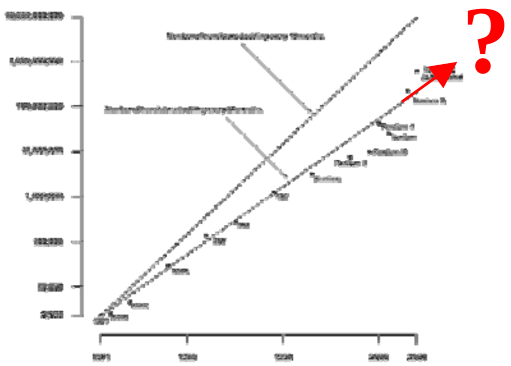


whether hardware,

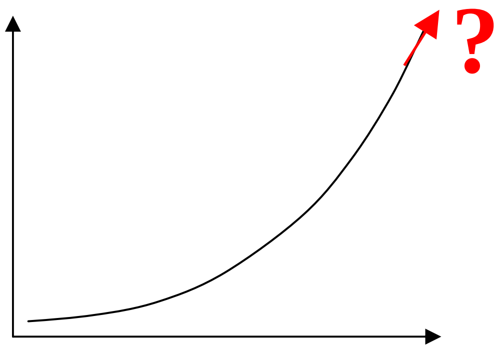


software,

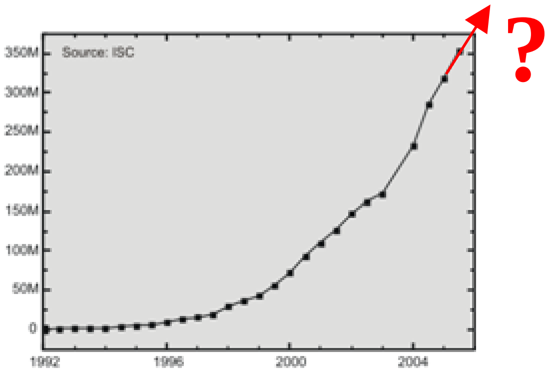
or (info) networks, ...



in number of transistors/year



in number of O/S lines of code/year



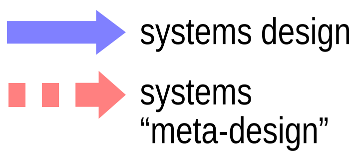
in number of network hosts/year

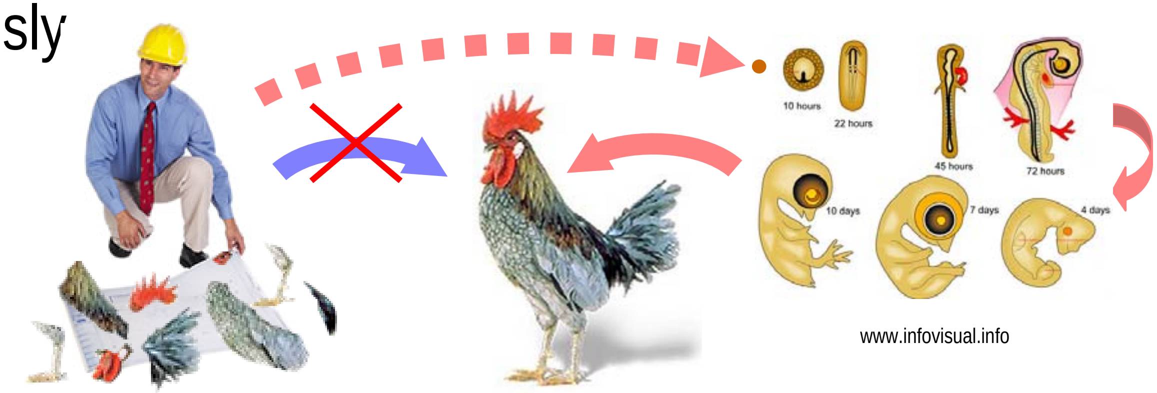
... is already pushing us to rethink ICT in terms of CS

1. Introduction — b. A vast archipelago

➢ Exporting self-assembly: from design to “meta-design”

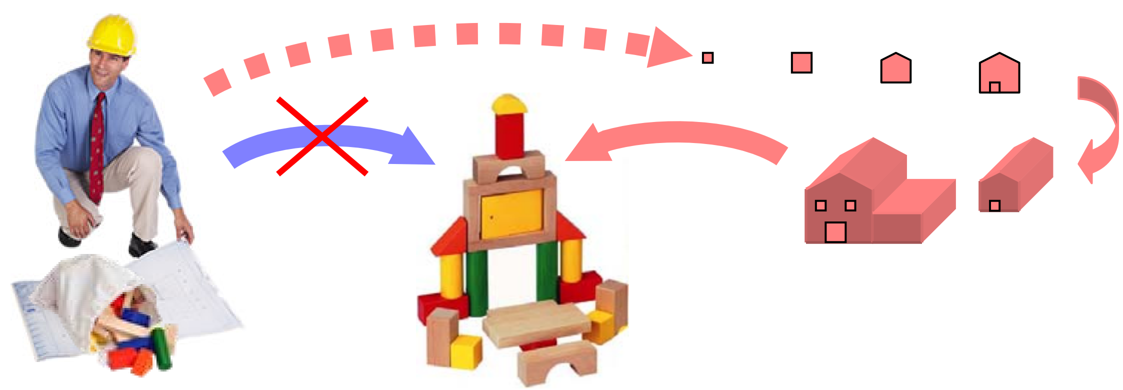
- ✓ organisms endogenously *grow* but artificial systems *are built* exogenously





- ✓ future engineers should “step back” from their creation and only set *generic* conditions for systems to self-assemble and evolve

don’t build the system (phenotype), <u>program the agents</u> (developmental genotype)



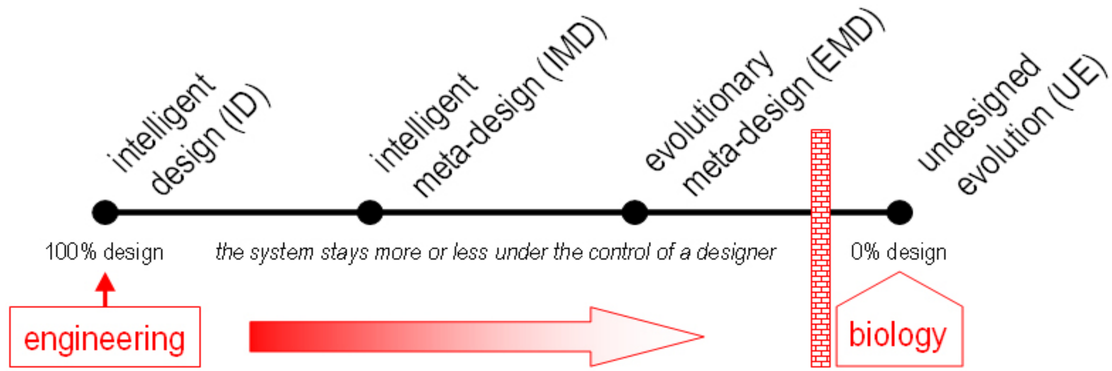
1. Introduction — b. A vast archipelago

➢ Natural C(A)S as a new paradigm for ICT engineering

- ✓ natural complex (adaptive) systems, biological or social, can become a new and powerful ***source of inspiration*** for future information & communication technologies (ICT) in their transition toward autonomy
- ✓ “emergent engineering” will be less about direct design and more about developmental and evolutionary ***meta-design***
- ✓ decentralized, unplanned “complex systems” are probably the most ***pervasive***, ***efficient*** and ***robust*** type of systems—maybe, in fact, the “simplest”?
- ✓ it is centralized, planned systems (computers, aircrafts, orchestras, armies) that are ***singular***, ***costly*** and ***fragile***, as they require another intelligent system (humans or other machines) to ***be*** organized, built, operated and/or controlled

1. Introduction — b. A vast archipelago

➢ Pushing ICT engineering toward evo-devo biology



intelligent (deliberate) design		intelligent & evolutionary "meta-design"
heteronomous order	▪	autonomous order
centralized control	▪	decentralized control
manual, extensional design	▪	automated, intentional design
engineer as a micromanager	▪	engineer as a lawmaker
rigidly placing components	▪	allowing fuzzy self-placement
tightly optimized systems	▪	hyperdistributed & redundant systems
sensitive to part failures	▪	insensitive to part failures
need to control	▪	prepare to adapt & self-regulate
need to redesign	▪	prepare to learn & evolve
complicated systems: planes, ~~computers~~	▪	*complex* systems: Web, market ... computers?

Complex Systems Made Simple

1. Introduction
   a. What are complex systems?
   b. A vast archipelago
   c. Computational modeling
2. A Complex Systems Sampler
3. Commonalities
4. NetLogo Tutorial

1. Introduction — c. Computational modeling

What this course is about

- an *exploration* of various complex systems *objects* (i.e., made of many agents, with simple or complex rules, and complex behavior):
  - cellular automata, pattern formation, swarm intelligence, complex networks, spatial communities, structured morphogenesis
- and their common *questions*:
  - emergence, self-organization, positive feedback, decentralization, between simple and disordered, “more is different”, adaptation & evolution
- by interactive *experimentation* (using NetLogo),
- introducing *practical* complex systems *modeling* and simulation
- from a *computational* viewpoint, as opposed to a “mathematical” one (i.e., formal or numerical resolution of symbolic equations),
- based on discrete *agents* moving in discrete or quasi-continuous space, and *interacting* with each other and their environment

➢ What this course is *not*

- ✓ a technical course about the archipelago of related disciplines
  - an information theory / computational complexity class
  - a dynamical systems / chaos / fractals / stochastic processes class
  - a systems engineering / control theory class
  - a graph theory / networks / statistical physics class
- ✓ a technical course about big questions × big objects
  - a fluid dynamics class
  - a condensed matter class
  - an embryology class
  - a neuroscience class
  - an entomology class
  - a sociology class
  - an economics class
  - ...

... you can wake up now

Existence of macro-equations for some dynamic systems

- ✓ we are typically interested in obtaining an explicit description or expression of the behavior of a whole system over time
- ✓ in the case of dynamical systems, this means *solving* their evolution rules, traditionally a set of ***differential equations*** (DEs)
- ✓ either *ordinary* (O)DEs of *macro-variables* in *well-mixed* systems
  - ex: in chemical kinetics, the law of mass action governing concentrations:

    $\alpha A + \beta B \rightarrow \gamma C$ described by $d[A]/dt = -\alpha k\,[A]^\alpha\,[B]^\beta$
  - ex: in economics, (simplistic) laws of gross domestic product (GDP) change:

    $dG(t)/dt = \rho\,G(t)$
- ✓ or *partial* (P)DEs of *local variables* in *spatially extended* systems
  - ex: heat equation: $\partial u/\partial t = \alpha\nabla^2 u$, wave equation: $\partial^2 u/\partial t^2 = c^2\nabla^2 u$
  - ex: Navier-Stokes in fluid dynamics, Maxwell in electromagnetism, etc.

1. Introduction — c. Computational modeling

- **Existence of macro-equations and an analytical solution**
  - ✓ in some cases, the explicit formulation of an exact solution can be found by calculus, i.e., the *symbolic manipulation of expressions*
    - ex: geometric GDP growth ⇒ exponential function

      $$dG(t)/dt = \rho\, G(t) \quad \Rightarrow \quad G(t) = G(0)\, e^{-\rho t}$$

    - ex: heat equation ⇒ linear in 1D borders; widening Gaussian around Dirac

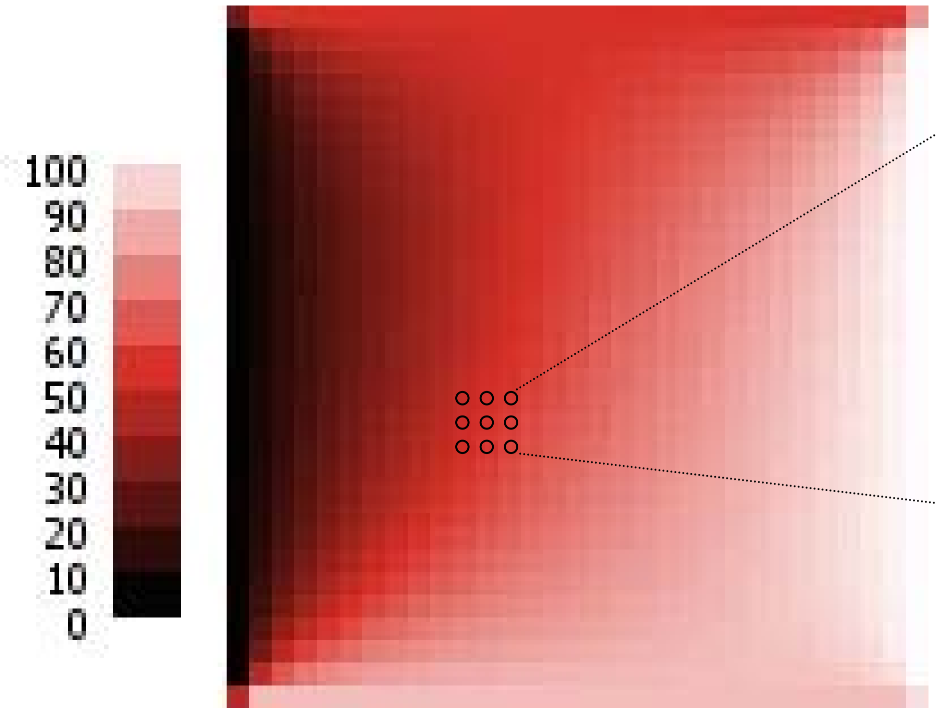
      $$\partial u/\partial t = \alpha\, \partial^2 u/\partial^2 x \text{ and } u(x,0) = \delta(x) \Rightarrow u(x,t) = \frac{1}{\sqrt{4\pi kt}} \exp\left(-\frac{x^2}{4kt}\right)$$

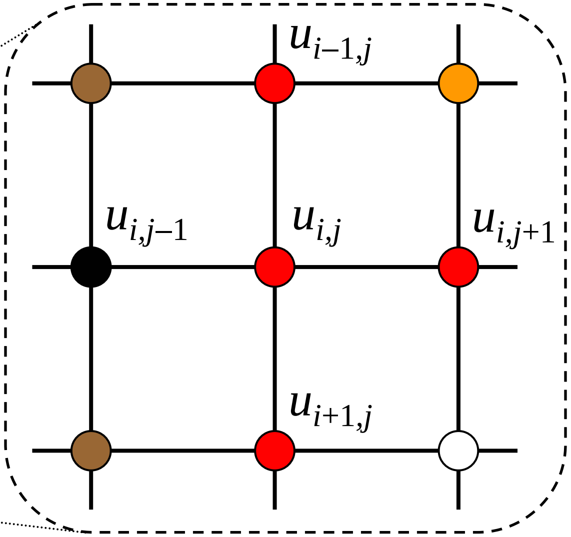
  - ✓ calculus (or analysis) relies on known shortcuts in the world of mathematical “regularities”, i.e., the family of continuous, derivable and integrable functions that can be expressed symbolically
  - ✓ unfortunately, although vast, this family is in fact very small compared to the immense range of dynamical behaviors that natural complex systems can exhibit!

1. Introduction — c. Computational modeling

- **Existence of macro-equations but no analytical solution**
  - ✓ when there is no symbolic resolution of an equation, *numerical analysis* involving algorithms (step-by-step recipes) can be used
  - ✓ it involves the discretization of space into cells, and time into steps

NetLogo model: /Chemistry & Physics/Heat/Unverified/Heat Diffusion





$\partial u/\partial t = \alpha\nabla^2 u$ by forward Euler $\Rightarrow$

$$\Delta u_{i,j} = \alpha(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j})$$

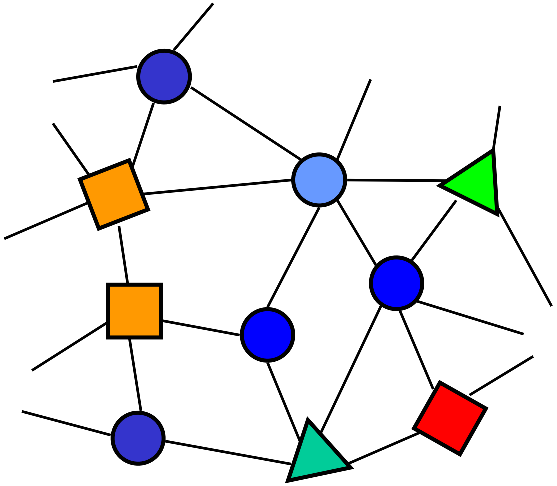
Absence of macro-equations

- *"The study of non-linear physics is like the study of non-elephant biology."* —Stanislaw Ulam
  - the physical world is a fundamentally *non-linear* and *out-of-equilibrium* process
  - focusing on linear approximations and stable points is missing the big picture in most cases
- let's push this quip: *"The study of non-analytical complex systems is like the study of non-elephant biology."* —??
  - complex systems have their own "elephant" species, too: dynamical systems that can be described by differential equations
  - most real-world complex systems do not obey such neat, macroscopic laws

1. Introduction — c. Computational modeling

➢ Where global ODEs and spatial PDEs break down...

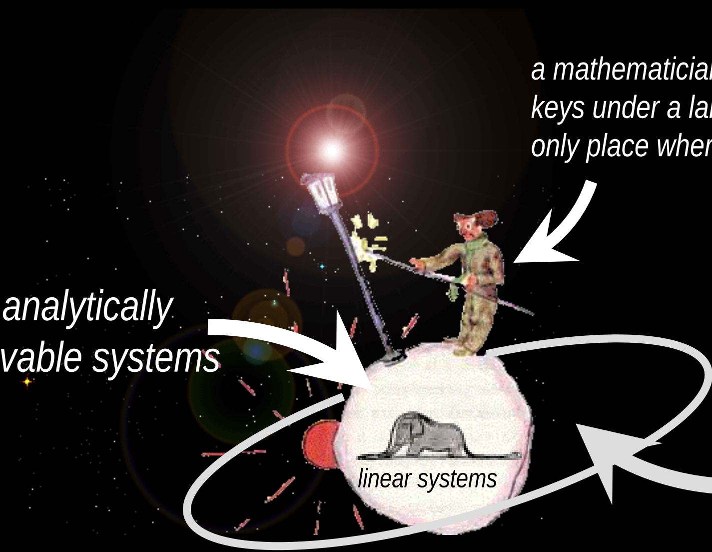
- ✓ systems that *no macroscopic quantity* suffices to explain (O)
  - no magical law of “concentration”, “pressure”, or “gross domestic product”
  - even if global metrics can be designed to give an indication about the system’s dynamical regimes, they rarely obey a given differential equation
- ✓ systems that require a *non-Cartesian* decomposition of space (P)
  - network of irregularly placed or mobile *agents*
- ✓ systems that contain *heterogeneity*
  - segmentation into different *types of agents*
  - at a fine grain, this would require a “patchwork” of regional equations (ex: embryo)
- ✓ systems that are *adaptive* (learn, evolve)
  - the topology and strength of the interactions depend on the short-term activity of the agents and long-term “fitness” of the system in its environment



➢ The world of complex systems modeling

a mathematician (physicist?) looking for his keys under a lamp post, because this is the only place where there is (analytical) light

analytically solvable systems



analytically expressable, numerically solvable systems

all the rest:

non-analytically expressable systems

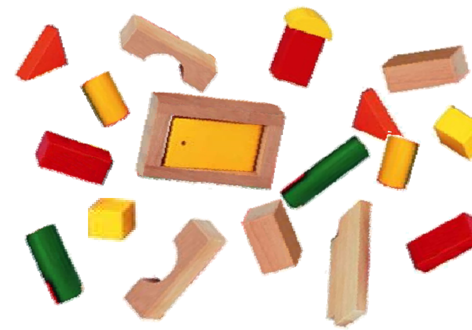
⇒ computational models

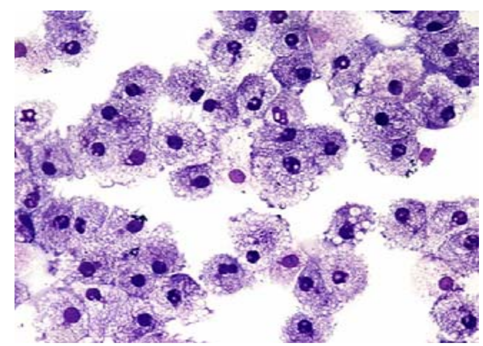
The Lamplighter & the Elephant-Digesting Boa, from "The Little Prince"
Antoine de Saint-Exupéry (born in Lyon)

➢ The world of computational modeling

- ✓ not a cold and dark place!... it is teeming with myriads of *agents* that carry (micro-)*rules*

a computer scientist (physicist?) populating the world with agents

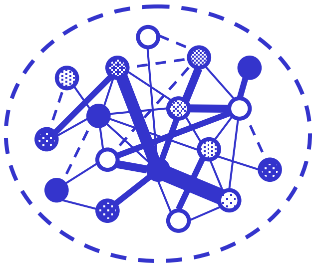




- ✓ the operational concept of “agent” is inspired from “social” groups: people, insects, cells, modules: agents have *goals* and *interactions*

1. Introduction — c. Computational modeling

➢ ABM meets MAS: two (slightly) different perspectives

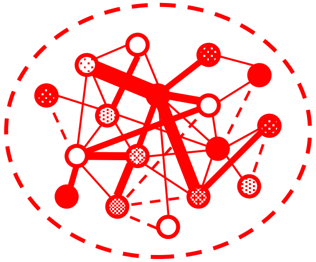


CS science: understand "natural" CS

→ *Agent-Based Modeling (ABM)*

... "Multi Agent-Based Modeling and Simulation Systems" (MABMSS)??

computational complex systems



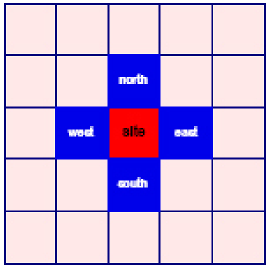
CS engineering: design a new generation of "artificial" CS → *Multi-Agent Systems (MAS)*

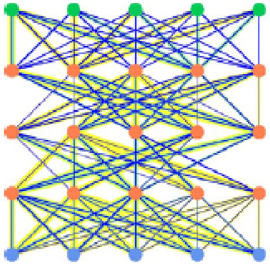
✓ but again, don't take this distinction too seriously! they overlap a lot

➢ ABM: the perspective from CA and social sciences

- ✓ *agent-* (or individual-) ***based modeling*** arose from the need to model systems that were too complex for analytical descriptions
- ✓ one time line going through cellular automata (CA)
  - von Neumann self-replicating machines → Ulam's "paper" abstraction into CAs → Conway's *Game of Life*
  - based on ***grid*** topology
- ✓ other origins rooted in economics and social sciences
  - related to "methodological individualism"
  - mostly based on grid and ***network*** topologies
- ✓ later extended to ecology, biology and physics
  - based on grid, network and 2D/3D ***Euclidean*** topologies

→ the rise of fast computing made ABM a practical tool



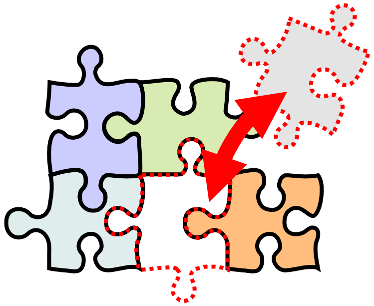


Macal & North
Argonne National Laboratory

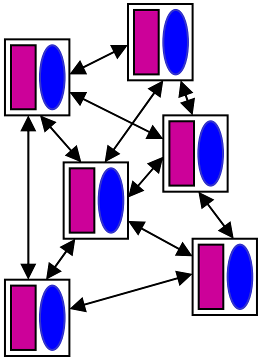
1. Introduction — c. Computational modeling

➢ MAS: the perspective from computer science and AI

- ✓ in software engineering, the need for clean *architectures*
  - historical trend: breaking up big monolithic code into *layers*, *modules* or *objects* that communicate via application programming *interfaces* (APIs)
  - this allows fixing, upgrading, or replacing parts without disturbing the rest
- ✓ in AI, the need for *distribution* (formerly "DAI")
  - break up big "intelligent" systems into smaller, less exhaustive units: *software / intelligent agents*



- ✓ difference with object-oriented programming:
  - agents are "proactive" / autonomously threaded
- ✓ difference with distributed (operating) systems:
  - agents don't appear transparently as one coherent system



→ the rise of pervasive networking made MAS a practical technology

1. Introduction — c. Computational modeling

➢ MAS: the perspective from computer science and AI

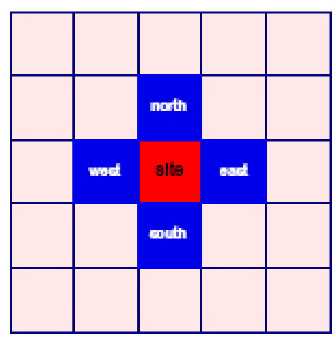
- ✓ emphasis on software agent as a *proxy* representing human users and their interests; users state their prefs, agents try to statisfy them
  - ex: internet agents searching information
  - ex: electronic broker agents competing / cooperating to reach an agreement
  - ex: automation agents controlling and monitoring devices
- ✓ main tasks of MAS programming: agent design and society design
  - an agent can be ± reactive, proactive, deliberative, social (Wooldridge)
  - an agent is caught between (a) its own (sophisticated) goals and (b) the constraints from the environment and exchanges with the other agents
- ✓ differences with the ABM philosophy
  - focus on few "heavy-weight" (big program), "selfish", intelligent agents, as opposed to many "light-weight" (few rules), highly "social", simple agents
  - focus on game theoretic gains, as opposed to collective emergent behavior

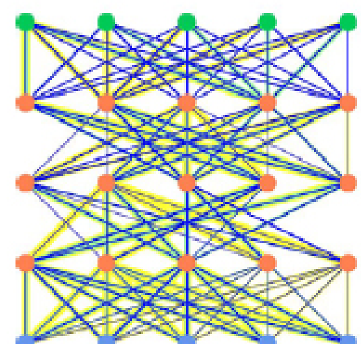
➢ An agent in this course

- ✓ a (small) program deemed "local" or "autonomous" because it has
  - its own scheduling (execution process or thread)
  - its own memory (data encapsulation)
  - ... generally simulated in a virtual machine
- ✓ this agent-level program can consist of
  - a set of dynamical equations ("reactive")
  - a set of logical rules (AI)... or a mix of both

Hugo Weaving as Agent Smith
The Matrix Revolutions, Warner Bros.

- ✓ peer-to-peer interactions among agents under different topologies

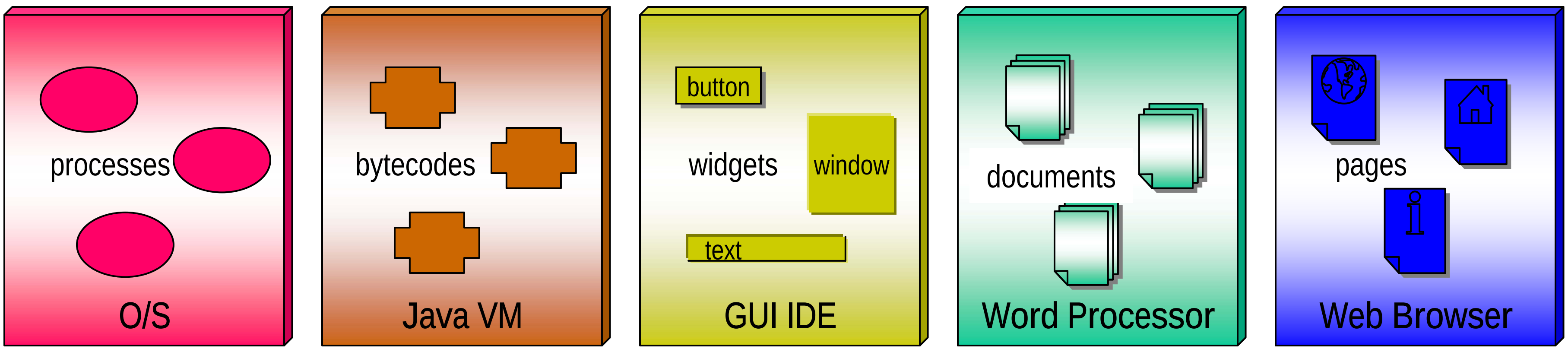




1. Introduction — c. Computational modeling

➢ Agent virtual machines or "platforms"

✓ just like there are various middleware-componentware frameworks...



✓ ... there are also ABM platforms, e.g., *NetLogo*, *Swarm*, or *Repast*