The Theory of Systems: An Information Oriented Approach

Geoff Sharman

Why?

- We **recognise** systems in our experience: we are surrounded by them:
    - physical (mechanical, electrical, electronic, software, etc)
    - cosmological
    - chemical
    - biological
    - social (ethnic, organisational, governmental, etc)
- General belief that a system is “**more than the sum of its parts**”
- But **what is a system?** What makes the difference and how do systems work? Are there any general principles? What can we learn about building systems?
- We want to present a **synthesis** based on known scientific principles

Dramatis Personae

- This synthesis is based on the work of, amongst others:
- Sir Isaac Newton
- Sir Benjamin Thompson/Count Rumford
- Ludwig Boltzmann
- Charles Darwin
- Alan Turing
- Richard Feynman
- and others …

Definition

- A **system is a collection of components** which are more **strongly bound** to each other than they are to their environment (***identity*** property)

- A **component** may be anything, including another **system** (***hierarchy*** property)

- **BUT** note that many of these words (e.g. “component”, “bound”, “environment”) are not precisely defined, so further explanation is needed

Why can Systems Exist?

- Ultimately, systems can exist because:
    - there are **stable, long lived components** with which to build them (e.g. protons, nuclei, atoms)
    - there are **forces** which can **bind** components together (e.g. strong, weak, electromagnetic, gravitation)
- BUT what are the equivalents of force in a biological or social system?

Locality

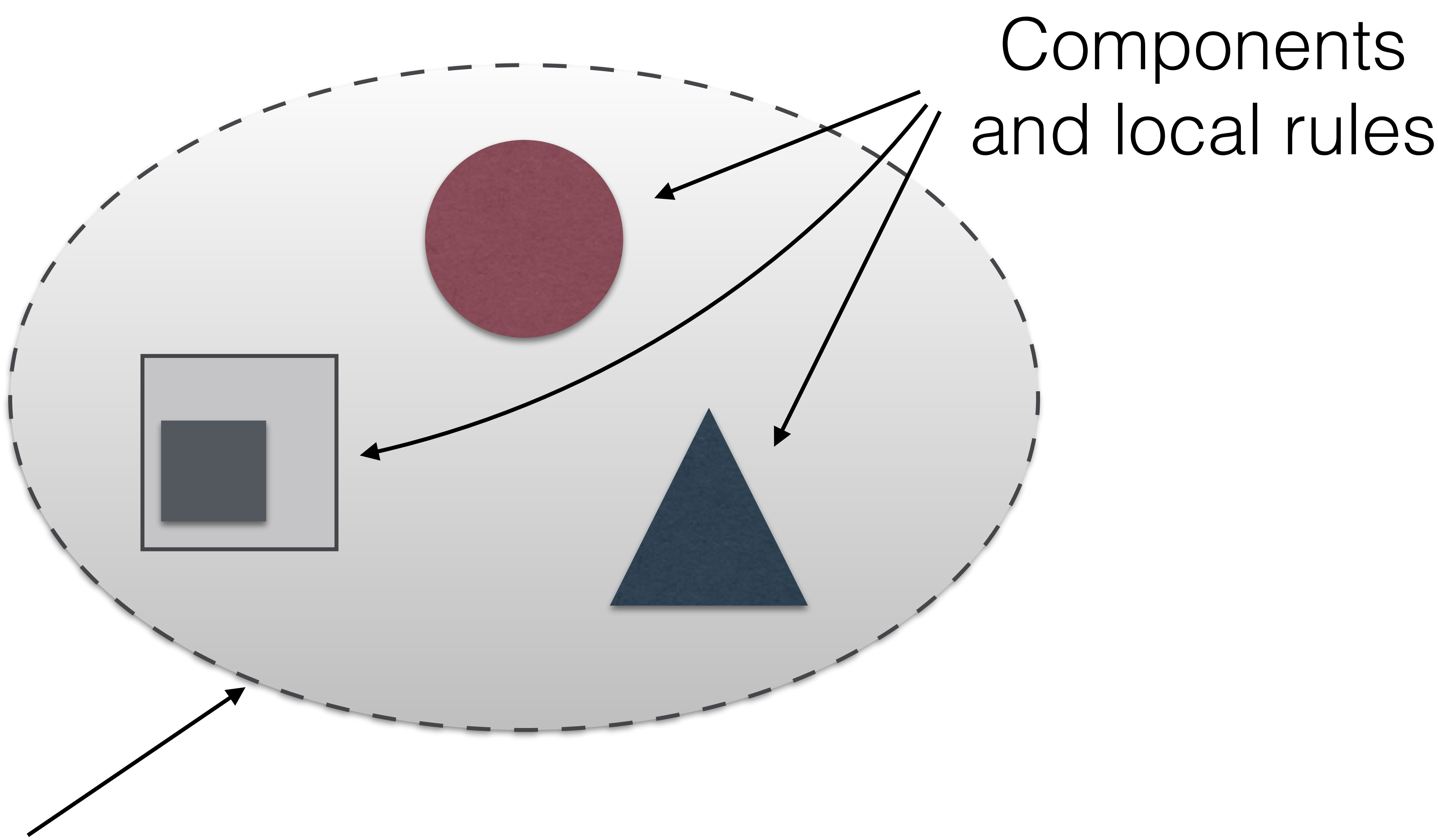
- The essence of a system is that it is **local** and its **rules** are local … as are scientific laws

- Every system has a **boundary**, which separates what is "in the system" from what is "outside the system"

- Every system has an "**environment**", which is just another name for "everything outside the system"

- So we can draw a **diagram** to show a system and its boundary

Schematic of a System



- Boundary
- Environment = everything outside system

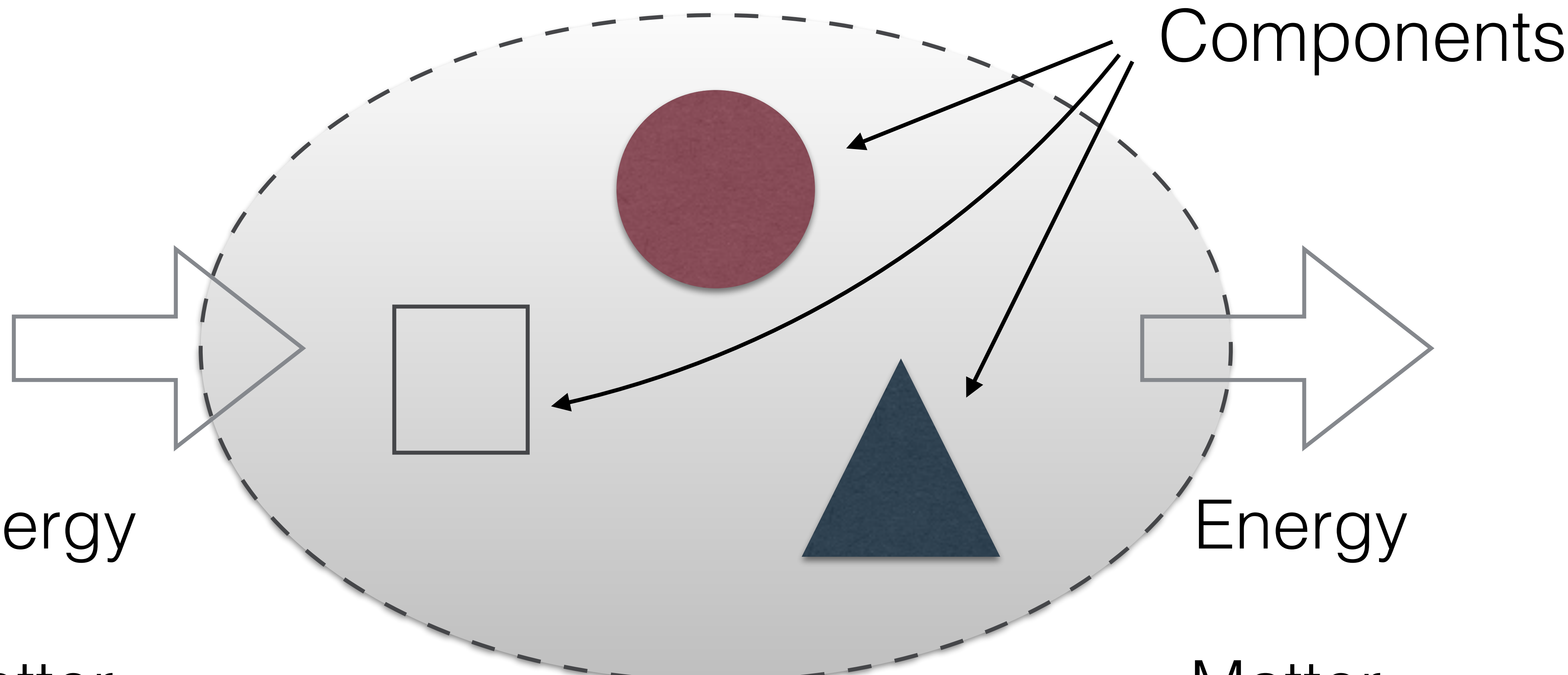
What is the Environment?

- The "environment" is **everything else in the Universe** and it has an effect on any system:
  - E.g. all systems **exchange heat** with their surroundings and are subject to **gravitational** forces
  - When the environmental effects are small, systems tend towards **equilibrium**
- Every system also **affects** its environment and leaves its "footprint" on it

Binding vs. Containment

- How does a system maintain its **identity**?
- It must have:
    - **internal binding forces** which are stronger than those exerted by the environment, **OR**
    - a **physical envelope** which contains components and separates them from the environment
    - or both

Inputs and Outputs



- Energy
- Matter
- Information

Energy

Matter

Information

Systems Principle # 1

- **Outputs from a system must balance inputs**
  - In Physics, this is known as the **conservation of mass/energy**
  - Systems move and change, but you **cannot build a perpetual motion machine**!
  - Sometimes also known as the **First Law of Thermodynamics**

How Systems Change

- In small scale systems, change is **deterministic** and **reversible** (e.g. collisions between particles)
- Given the **state** of the system **at a given time**, we can calculate a **subsequent** state or an **earlier** state using **equations of motion**
- By “state”, we mean a complete specification of the **momentum** (energy & direction) of each component (e.g. particle) in the system
- This state is known as **information**; momentum is conserved, therefore information is conserved (usually)

Large Scale Systems

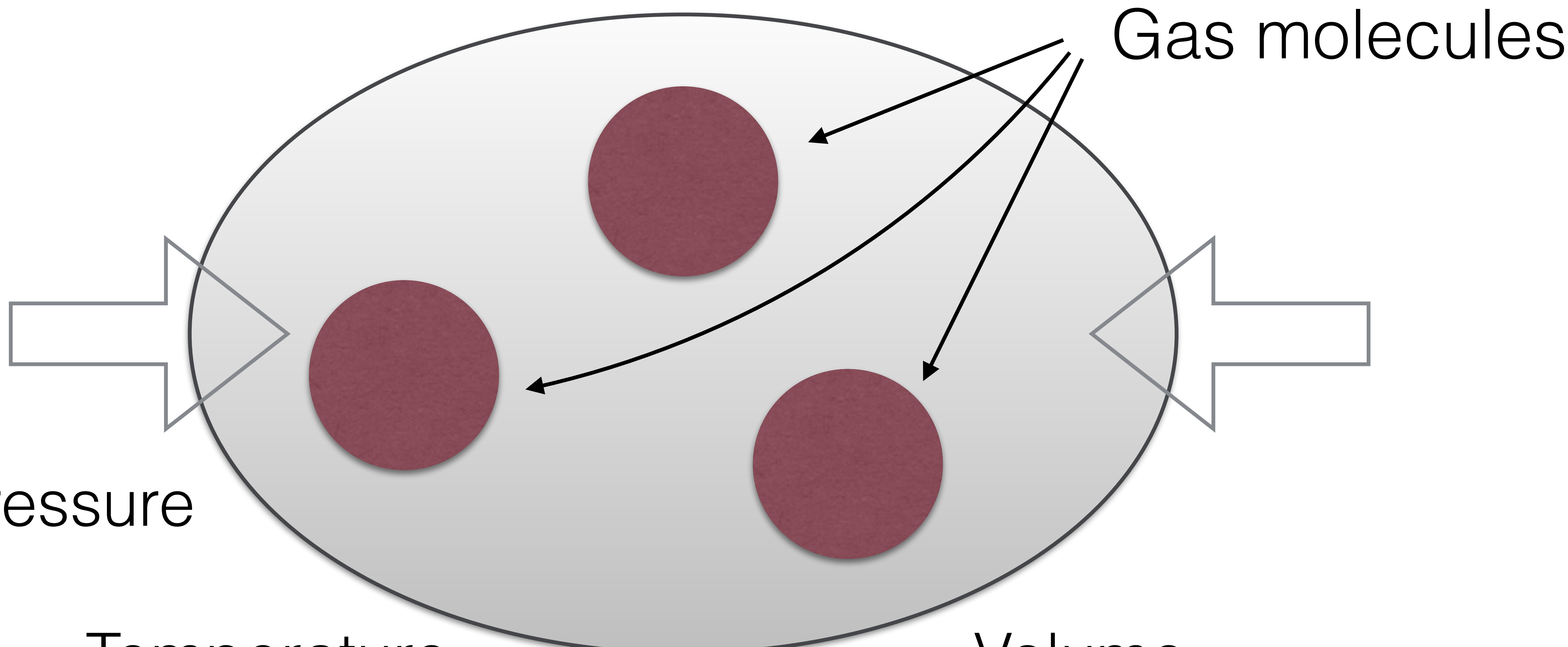
- By **large scale**, we mean systems with **many components** and, especially, **many similar components** (e.g. molecules of gas)

- The equations of motion for a large system **cannot be solved,** so motion appears random

- The effect of **many interactions** between components (e.g. collisions between gas molecules) is to **share energy** between the components, leading to an equilibrium **state** (e.g. energy profile) with many indistinguishable “**microstates**”

- We characterise this by **statistical measures**, e.g. **temperature** and **pressure**

Ideal Gas as a System



- Pressure
- Temperature Volume
- $PV = NkT$ (Ideal Gas Law)

Systems Principle # 2

- In large scale systems, **change is irreversible**
  - Systems are fragile!
- This occurs because **all microstates are equally likely** but only a few are simple and "useful"
- In Physics, this is known as the **Second Law of Thermodynamics**

Irreversibility!

Lack of Information

- It's impossible to have **full knowledge** of the state of a large scale system: calculation/measurement could take longer than the age of the universe
- We **perceive** this as "disorder"
- The Second Law of Thermodynamics is a consequence of this lack of information, usually called **entropy**, which always increases
- $S = k \log W$ (W = number of micro states)

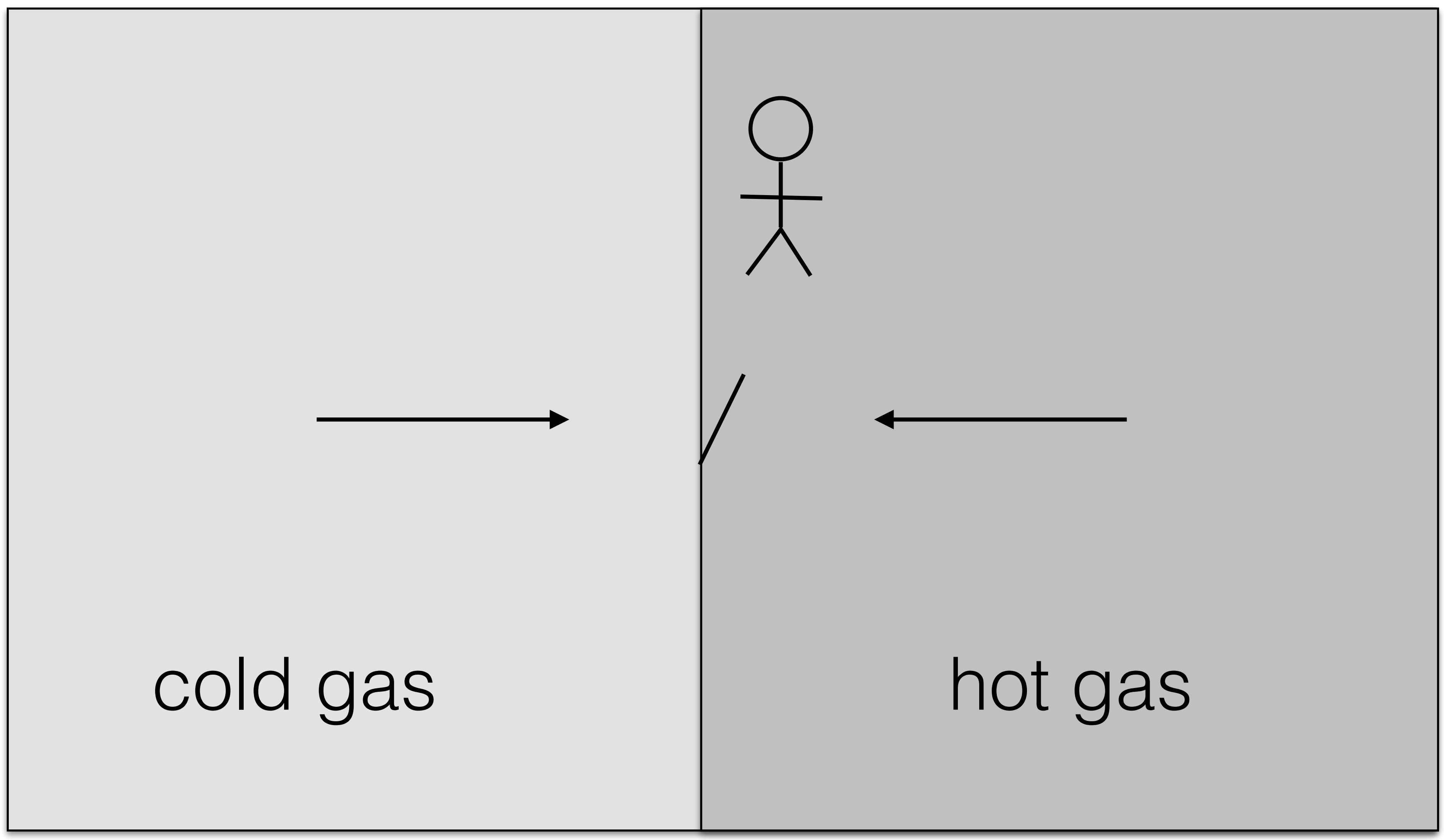
Systems Encode Information

- All large scale systems have **information encoded within them**:
  - at the **low level** (e.g. direction of motion, magnetisation, nuclear spin, etc.)
  - at larger scale, a **record of irreversible change**
- This is what makes geology, archaeology, forensic science and memory devices possible

Maxwell’s Sorting Demon

- Maxwell suggested that the supposedly irreversible changes to a gas could be reversed by the action of a “**sorting demon**”
    - the demon would **manipulate** a “trap door” separating two chambers of gas
    - **open** it to allow high velocity molecules into one chamber, but **close** it to low velocity molecules
    - gradually **accumulate** high velocity (**hot**) molecules in one chamber and slow (**cold**) molecules in the other
- Is this possible? What would it take to implement it?

The Sorting Demon



Resolving the Paradox

- For almost 100 years, the sorting demon was thought to be **paradoxical** because it suggested that entropy could be decreased
- We now know **in principle** how to implement the demon with sensors and a memory device
  - But operating the demon requires **energy**
  - And **resetting** the memory after each particle detection has an energy cost
- This tells us that **total entropy increases** in the environment

Open vs. Closed Systems

- An **open system** accepts inputs and outputs
- A **closed system** is **isolated** from its environment, so does not accept inputs or outputs
- But both these are **contradictory**:
  - a fully open system would **lose its identity**
  - a fully closed system is **impossible**
  - so most systems are partially open, partially closed

Systems Principle #3

- **There are no closed systems**
- In Physics, this is known as the **Third Law of Thermodynamics**
- It’s usually expressed as *“**entropy tends to zero at absolute zero**”* or as *“it’s impossible to reach absolute zero”*
- In other words, you cannot **isolate** a system

Complex Systems

Non-Equilibrium Systems

- Most interesting real systems (e.g. plants, animals, machines, and planets) operate in a range of states which are **far from equilibrium**
- They take **inputs** of energy/matter from the environment in order to operate
- They produce **outputs** of waste energy/matter to the environment
- They have **complex** internal structure and may occur in **large scale** communities

Large Scale Behaviour

- Some **behaviour** of large scale systems can be predicted from low level properties, or described by simple laws e.g.
    - **Mass** of system = sum (masses of components) acting at centre of gravity
    - **Motion of rigid bodies** with a centre of gravity is described by Newton's Law $s = vt + 1/2\ at^2$
    - **Behaviour of a gas** is given by Boyle's Law $PV = RT$
- These predictions are only **valid** over **certain ranges**

Complex Systems

- Systems with **many levels of nested components** can exhibit complex behaviours, e.g. actuating mechanisms, moving around, sending messages, making things, etc.
- They can operate as **functional processes**, transforming specific inputs into specific outputs
- They can also **protect** themselves against changes in the environment and exhibit **resilience**
- They can **repair** themselves and **reconfigure** in response to external conditions

Emergence

- Sometimes, large scale complex systems exhibit **surprising** and **unexpected** behaviour, e.g.
  - **Phase change** from gas to liquid or solid, **crystallisation** into geometric shapes, **complex motion** of water waves
  - **Chemical processes**, e.g. catalysis
  - **Biological processes**, e.g. life, consciousness
- We use multiple **levels of description** to document these **emergent properties**

The Problem of Emergence

- Higher level behaviour appears to show **order**. The basic problem is to explain how this arises from apparent **disorder** in large scale systems

- Some, e.g phase change, is due to **micro-level interactions** between atoms and molecules, as a substance is cooled

- But some natural systems show **self organisation**, e.g. growth of living things. How does this happen?

Philosophical Explanations

- Philosophers traditionally use a number of approaches to explain complex behaviour
  - **Holism** => i.e. systems can **only be understood as a whole**
  - **Reductionism** => i.e. systems can be **fully understood from their basic parts**
  - **Vitalism** => living systems are made of **different stuff** from inorganic systems
- None of these explanations is satisfactory:
  - We **gain insight** by taking systems apart and understanding subsystems
  - We have **no evidence** for vitalism; living systems are made of the **same basic stuff** as simple systems
  - But can **all complex behaviours,** e.g. life, brains, consciousness, etc. be **understood** using just basic physics?

Emergence from Scale

- Experiments with **cellular automata**, e.g. Game of Life, have found self sustaining patterns, showing:
  - Simple **local rules** applied over a large scale can lead to **complex patterns** of behaviour
  - Some patterns include **gates** and **memory**, making them equivalent to a computer
- Some natural processes appear to be exhibit **symmetry** and **computation**, e.g. crystal growth

Mechanism for Emergence

- What is the **difference** between a working system and a pile of components? or between a living creature and a dead one?

- Some patterns of behaviour are **independent of the actual system components** or even their type, e.g. motion of fluids. How can we explain that?

- The answer to these question appears to be **information** and **computation**, which enables the parts to be organised and specifies interactions

Models of Computation

Systems Principle # 4

- **Complex systems use some form of computation**
- Three **classes** of system provide models of computation:
  - **Finite state machines** (finite fixed memory)
  - **Pushdown automata** (finite stack memory)
  - **Turing Machines** (infinite two stack memory) and equivalents, e.g. CAs
- Turing showed that:
  - Finding the output of a TM is an **undecidable problem**
  - All TMs have **equivalent capability**, **no matter how designed.** We call this the **Universality of Computation:** any computer can do the same as any other

Finite State Machines

- A finite state machine (automaton) is an **abstract system** which can can exist in a limited number of states, e.g. vending machine, lift
- The states provide a form of **memory**
- It can accept a **defined set of input signals** (often thought of as a string of symbols in some defined grammar) which cause **transition** from one defined state to another
- It provides a **restricted model of computation**, with less capability than a Turing machine - but then most real computers have finite memory

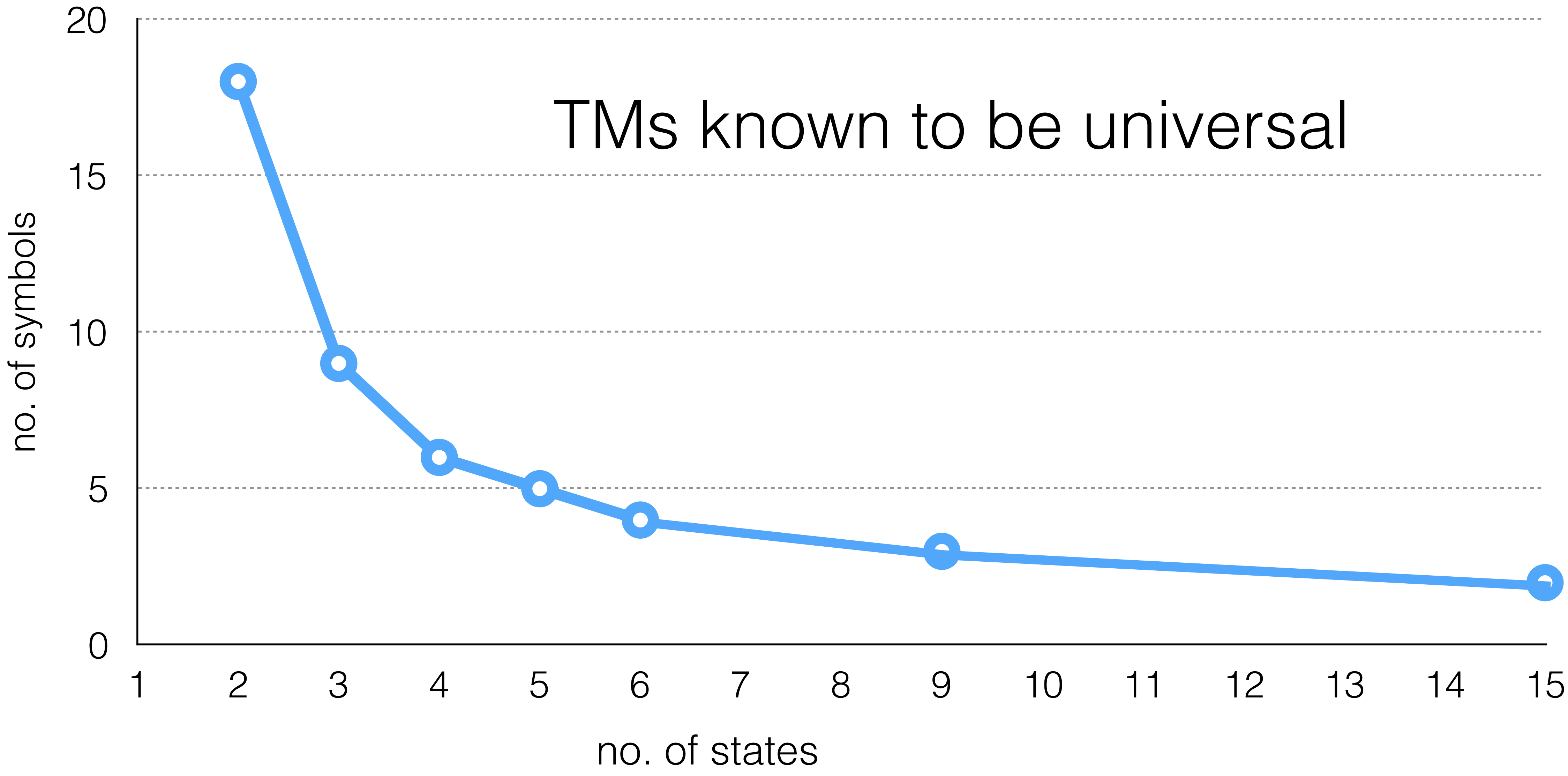
Computation in Nature

- Studies of one dimensional cellular automata suggest that some **natural processes** are equivalent to automata, e.g. turbulent fluids, weather systems
- We know that **biological cells** use a computing mechanism based on DNA
- Therefore, natural systems may perform **computation** and may exhibit **universality**
- Although we **cannot predict their behaviour** (through lack of information), we may be able to simulate it

Small Automata

- Margenstern studied Turing Machines with small numbers of **symbols** and **states**
- He asked what is the **smallest** automaton that exhibits universality?
- The answer is **surprisingly small**: a TM with 5 symbols and 5 states shows universality
- Studies of **small Cellular Automata** also show universality

Universality of TMs



Complex Adaptive Systems

Complex Adaptive Systems

- **Real world** complex systems have a number of common features:
  - **protective envelope**, with input/output ports
  - static **hierarchical structure** of components with **dynamic operational relationships**
  - the capability to **control** their function in response to external conditions
  - the capability to **exchange messages** between components and with the external environment
  - the capability to **model** themselves and the external environment
  - and sometimes, the capability to **replicate** themselves

System Structure

- Static **hierarchical** nesting of components follows from the definition of a system. Can be represented by a **tree**

- Dynamic **networked** structure depends on how components “invoke” each other by passing messages. Can be represented by a **graph**

- Graphs can include **cycles** (loops) and **recursion**

Control Systems

- Control systems (e.g. thermostat, speed controller) can maintain some parameter at a constant value (steady state) via a **negative feedback loop**. Requires two data values plus an algorithm:
    - **Desired** value of parameter
    - **Actual** measured value of parameter
    - **Algorithm** for returning parameter to desired value
- Most complex systems and all living things include control systems

Modelling

- Most complex systems (e.g. computers, living things) exist within **communities** of similar systems
- Their environment is **hazardous** and **competitive**, but may offer **rewards** (e.g. food)
- To function successfully, they need to know about the environment by **modelling** it and **monitoring** it
- Information is **encoded** within the system and associated with algorithms for **updating** the model, **responding** to changes, and **reacting** externally

Self Modelling

- Complex systems also need to **monitor and control themselves**, to:
  - **manage** inputs, workload, etc.
  - **maintain** outputs with varying inputs
  - ensure **correct operation** of internal components
  - **adapt** to failures and external changes
- For this, they need a **model of their own structure**

Replication

- Because systems are fragile, the capability to adapt and respond will **degrade with time.** All systems become **defunct** eventually
- Longer term, only systems with the capability to **replicate** themselves will survive
- This means **copying all the internal and external model information** and creating a physical replica, as in the reproductive cycle of life
- Artificial hardware systems are rarely capable of self replication but **software often is**, e.g. viruses

Ecosystems

- Darwin recognised that animals and plants exist within competitive **ecosystems**, where there are many complex **predator/prey relationships**: “everything is connected to everything else”
- He also recognised that individual plants and animals are subject to **natural variation**
- And an ecosystem functions as a system which exerts **selective pressure** on individuals within it

Systems Principle # 5

- In a large scale ecosystem of variable systems, **selective pressure will favour replication of the best adapted systems**

- This is the **Theory of Evolution.** It describes a mechanism for **learning** about the ecosystem

- It also applies to economic markets, software releases, and other systems where each iteration incorporates **new adaptations**

- The selective pressure is **customer requirements!**

Conclusions

What have we Learnt?

- **All** systems are subject to known **scientific principles**
- Many natural systems show **aspects of computation**
- All **complex adaptive systems** are **computers**
- **Evolution** occurs in **systems with many similar components**, not in individual component systems
- **Systems development** is the iterative process of learning customer requirements