Universal laws and architecture 3: Foundations for Sustainable Infrastructure

John Doyle
John G Braun Professor
Control and Dynamical Systems, EE, BE
Caltech
The 'skin of an onion' analogy is also helpful. In considering the functions of the mind or the brain we find certain operations which we can explain in purely mechanical terms. This we say does not correspond to the real mind: it is a sort of skin which we must strip off if we are to find the real mind. But then in what remains we find a further skin to be stripped off, and so on. Proceeding in this way do we ever come to the 'real' mind, or do we eventually come to the skin which has nothing in it? In the latter case the whole mind is mechanical.

1950, Computing Machinery and Intelligence, *Mind*
“Universal laws and architectures?”

• Universal “conservation laws” (constraints)
• Universal architectures (constraints that deconstrain)
• Mention recent papers*
• Focus on broader context not in papers
• Lots of case studies for motivation

*try to get you to read them?
• Turing 100\textsuperscript{th} birthday in 2012
• Turing
  – machine (math, CS)
  – test (AI, neuroscience)
  – pattern (biology)
• Arguably greatest*
  – all time math/engineering combination
  – WW2 hero
  – “invented” software

\textbf{Turing (1912-1954)}

Compute

*Also world-class runner.
Key papers/results

• Theory (1936): Turing machine (TM), computability, (un)decidability, universal machine (UTM)
• Practical design (early 1940s): code-breaking, including the design of code-breaking machines
• Practical design (late 1940s): general purpose digital computers and software, layered architecture
• Theory (1950): Turing test for machine intelligence
• Theory (1952): Reaction diffusion model of morphogenesis, plus practical use of digital computers to simulate biochemical reactions
Each theory ≈ one dimension
Tradeoffs *across* dimensions
Assume architectures a priori
Progress is encouraging, but…
Stovepipes are an obstacle…
Turing’s 3 step research:
0. Virtual (TM) machines
1. hard limits, (un)decidability using standard model (TM)
2. Universal architecture achieving hard limits (UTM)
3. Practical implementation in digital electronics (biology?)

Essentials:
0. Model
1. Universal laws
2. Universal architecture
3. Practical implementation

Turing as “new” starting point?
Turing’s 3 step research:
0. **Virtual (TM) machines**
   1. hard limits, (un)decidability using standard model (TM)
   2. Universal architecture achieving hard limits (UTM)
   3. Practical implementation in digital electronics (biology?)

Essentials:
0. **Model**
   1. Universal laws
   2. Universal architecture
   3. Practical implementation
• ...being digital should be of greater interest than that of being electronic. That it is electronic is certainly important because these machines owe their high speed to this... But this is virtually all that there is to be said on that subject.
• That the machine is digital however has more subtle significance. ... One can therefore work to any desired degree of accuracy.

1947 Lecture to LMS
• … digital … of greater interest than that of being electronic …
• …any desired degree of accuracy…
• This accuracy is not obtained by more careful machining of parts, control of temperature variations, and such means, but by a slight increase in the amount of equipment in the machine.

1947 Lecture to LMS
• Digital more important than electronic…
• Robustness: accuracy and repeatability.
• Achieved more by internal hidden complexity than precise components or environments.

Turing Machine (TM)
• Digital
• Symbolic
• Logical
• Repeatable
• … quite small errors in the initial conditions can have an overwhelming effect at a later time. The displacement of a single electron by a billionth of a centimetre at one moment might make the difference between a man being killed by an avalanche a year later, or escaping.

1950, Computing Machinery and Intelligence, Mind
• ... quite small errors in the initial conditions can have an overwhelming effect at a later time....

• It is an essential property of the mechanical systems which we have called 'discrete state machines' that this phenomenon does not occur.
• Even when we consider the actual physical machines instead of the idealised machines, reasonably accurate knowledge of the state at one moment yields reasonably accurate knowledge any number of steps later.

1950, Computing Machinery and Intelligence, *Mind*
Turing’s 3 step research:
0. Virtual (TM) machines
1. **hard limits, (un)decidability using standard model (TM)**
2. Universal architecture achieving hard limits (UTM)
3. Practical implementation in digital electronics (biology?)
Logic

\[ \infty \text{ memory} \]

- Time
  - Fast
  - Slow

- Logic

- TM has \( \infty \) memory

- Space
  - Large
Logic

- Time: slow → fast
- Space: large space → space is free

TM has $\infty$ memory

$\infty$ memory
time?

Decidable problem = \exists \text{ algorithm that solves it}

Most naively posed problems are undecidable.
Turing’s 3 step research:
0. Virtual (TM) machines
1. hard limits, (un)decidability using standard model (TM)
2. **Universal architecture achieving hard limits (UTM)**
3. Practical implementation in digital electronics (biology?)
2. Universal architecture achieving hard limits (UTM)

- Software: A Turing machine (TM) can be data for another Turing machine
- A Universal Turing Machine can run any TM
- A UTM is a virtual machine.
The halting problem

• Given a TM (i.e. a computer program)
• Does it halt (or run forever)?
• Or do more or less anything in particular.
• Undecidable! There does not exist a special TM that can tell if any other TM halts.
• i.e. the program HALT does not exist. 😞
**Thm:** TM $H=$HALT does not exist.

That is, there does not exist a program like this:

$$H(TM, input) \triangleq \begin{cases} 
1 & \text{if } TM\ (input) \text{ halts} \\
0 & \text{otherwise}
\end{cases}$$

**Proof** is by contradiction. Sorry, don’t know any alternative. And Turing is a god.
Thm: No such H exists.

Proof: Suppose it does. Then define 2 more programs:

\[ H(TM, input) \triangleq \begin{cases} 
1 & \text{if } TM \ (input) \text{ halts} \\
0 & \text{otherwise} 
\end{cases} \]

\[ H'(TM, input) \triangleq \begin{cases} 
1 & \text{if } H(TM, input) = 0 \\
\text{loop forever otherwise} & \text{otherwise} 
\end{cases} \]

\[ H* (TM) \triangleq H'(TM, TM) \]

Run \[ H* (H*) = H'(H*, H*) \]

\[ = \begin{cases} 
\text{halt if } H* (H*) \text{ loops forever} \\
\text{loop forever otherwise} 
\end{cases} \]

Contradiction!
Implications
• TMs and UTMs are perfectly repeatable
• But perfectly unpredictable
• Undecidable: Will a TM halt? Is a TM a UTM? Does a TM do X (for almost any X)?
• Easy to make UTMs, but hard to recognize them.
• Is anything decidable? Yes, many questions NOT about TMs.
• Large, thin, nonconvex everywhere...
Issues for engineering

- Turing remarkably relevant for 76 years
- UTMs are \(\approx\) implementable
  - Time is most critical resource
  - Space (memory) almost free
- Read/write random access memory hierarchies
- Further gradations of decidable (P/NP/coNP)

- Must crucial: You can fix bugs but it is hard to automate finding/avoiding them
Issues for neuroscience

• Brains and UTMs?
  – Time is most critical resource?
  – Space (memory) almost free?
• Read/write random access memory hierarchies?
• Brain >> UTM?

Gallistel and King

Memory and the Computational Brain
Why Cognitive Science Will Transform Neuroscience

C.R. Gallistel and Adam Philip King
Conjecture

• Memory potential $\approx \infty$

• Examples
  – Insects
  – Scrub jays
  – Autistic Savants

• But why so rare and/or accidental?
• Large memory, computation of limited value?
• Selection favors fast robust action?
Compute
Turing

Delay is most important
Bode

Control, OR

Communicate
Shannon

Delay is least important
Carnot

Turing

Heisenberg

Boltzmann

Einstein

Physics
• Suppose we only care about space?
• And time is free
• Bad news: optimal compression is undecidable.
• Shannon: change the problem!
Shannon’s brilliant insight

- Don’t worry about time or delay!
- Don’t compress and code files, worry only about infinite random ensembles

- Information theory is most popular and accessible topic in systems engineering
- Fantastic for engineering, almost useless for biology (But see Lestas, Vinnicombe, Paulsson)
- (And largely irrelevant to Internet architecture)
- Misled and distracted generations of biologists and neuroscientists
- New generation of information theorists are putting delay back in. (Cheer!)
Compute
  Turing

Delay is most important

Control, OR

Communicate
  Shannon

Delay is least important

Physics
  Boltzmann
  Carnot
  Einstein
  Heisenberg
  Bode
  Turing

Shannon
Software

Hardware

Digital

Analog

Slow execution
Flexible reprogramming

Faster execution
Less flexible

Modern technology gives lots of intermediate alternatives.
Want to emphasize the differences between these two types of layering.
The virtual is more “real” than the implementation.
Software
Hardware

Slow
Flexible

Fast
Inflexible

Digital
Analog

Hard limits?

Good architectures?
Fast
Inflexible

Slow
Flexible

Predictions
Goals

Actions

Meta-layers

Physiology

Errors

Fast
Inflexible

THINKING, FAST AND SLOW
DANIEL KAHNEMAN

WHO'S IN CHARGE?
FREE WILL AND THE SCIENCE OF THE BRAIN
MICHAEL S. GAZZANIGA
Essentials To Do

• Reyna/Brainerd: Gist, false memory
• Ashby: Automaticity, multiple memory systems,…
• Cosmides/Tooby: Risk, uncertainty, cooperation, evolution,…
Speed and flexibility are crucial to implementing robust controllers.
Speed and flexibility are crucial to implementing robust controllers.
Beyond black boxes: Putting brain physiology back in the picture

Essentials (following Turing)
0. Model
1. Universal laws
2. Universal architecture
3. Practical implementation

Beyond black boxes: Putting brain physiology back in the picture

Essentials (following Turing)
0. Model
1. Universal laws
2. Universal architecture
3. Practical implementation
Compute

Turing

Delay is most important

Control

Bode

Bode

Software

Hardware

Digital

Analog

Fast Inflexible

Slow Flexible

Act

Sense

Plant

Compute

Turing

Delay is most important

Control

Bode

Bode

Software

Hardware

Digital

Analog

Fast Inflexible

Slow Flexible

Act

Sense

Plant
The starting point involves software and hardware, which can be digital or analog. Computers follow with digital and analog components, coupled with active and lumped/distribute elements. This leads to actuating sensors/amplifiers, which connects back to the "plant."
Maybe start from here with Turing’s 3 step research:
1. hard limits, (un)decidability using standard model (TM)
2. Universal architecture achieving hard limits (UTM)
3. Practical implementation in digital electronics (biology?)

Essentials:
0. Model
1. Universal laws
2. Universal architecture
3. Practical implementation

Turing as “new” starting point?
- Each theory $\approx$ one dimension
- Tradeoffs **across** dimensions
- Assume architectures a priori
- Progress is encouraging, but...
- Stovepipes are an obstacle...
laws and architectures?

Case studies

Sharpen hard bounds

Hard limit

efficient

wasteful

fragile

robust

efficient
Viruses’ Life History: Towards a Mechanistic Basis of a Trade-Off between Survival and Reproduction among Phages

Marianne De Paepe, François Taddei
Laboratoire de Genetique Moleculaire, Evolutive et Medicale, University of Paris 5, INSERM, Paris, France

July 2006 | Volume 4 | Issue 7 | e193