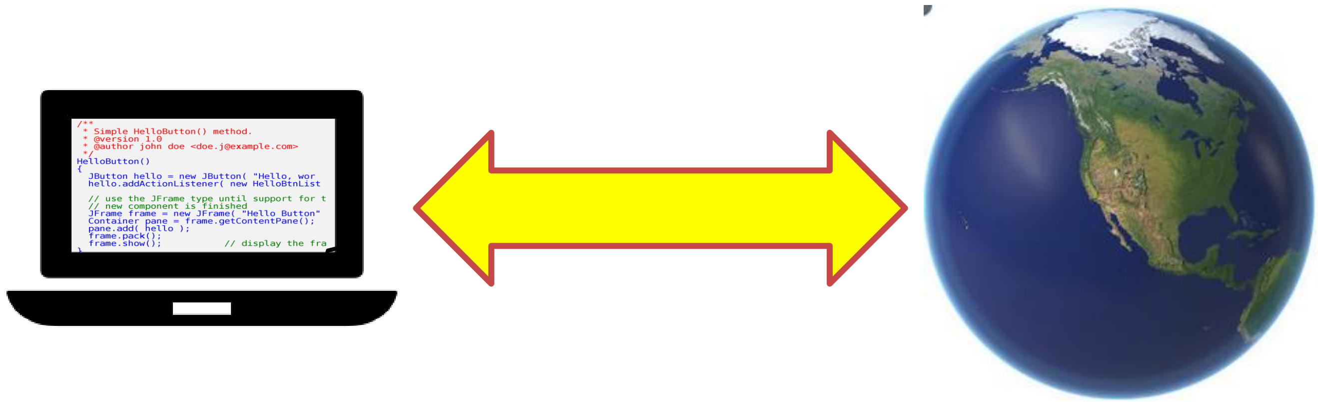


Matching Software with Reality

-- *"Software Meets the Real World"* –

A Reflective Back to Front View

Bran Selić

Malina Software Corp. (Canada)
Monash University (Australia)

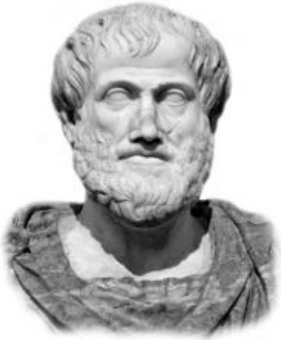
A Teaser…

- Do you happen to know the interesting story behind this number?

42

We shall return to this question later…

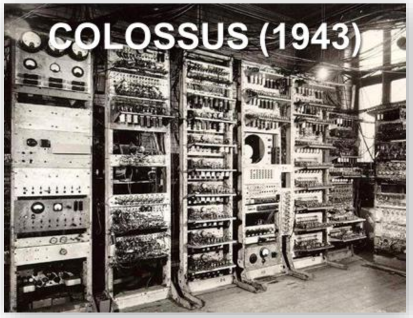
Part 1:
Where We Came From



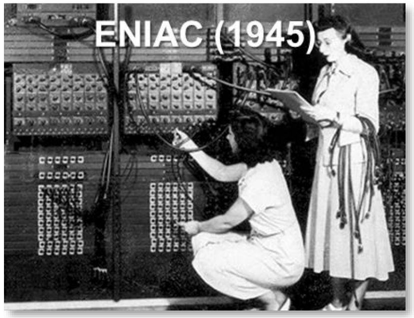
"If we desire to understand something, we need to know how it came to be."

-Aristotle

Where It All Started…



Codebreaking



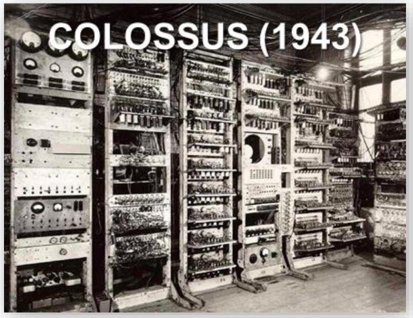
Ballistics Table Computations

- **The initial applications were for numerical analyses**
  - ⇒ Computing technology design was heavily influenced by the end users: mathematicians
- **Mathematicians preferred to view computers as technological embodiments of abstract mathematical concepts**
  - This approach has had (and still has) a fundamental influence on software technology

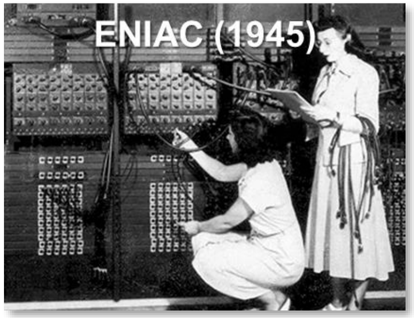
Edsgar W. Dijkstra:

"Too few people recognize that the high technology so celebrated today is essentially a mathematical technology."

Where It All Started…



Codebreaking



Ballistics Table Computations

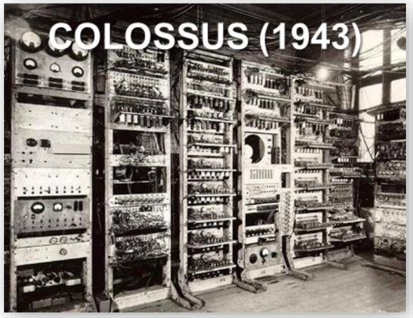
- The initial applications were for numerical analyses
  - ⇒ Computing technology design was heavily influenced by the end users: mathematicians
- Mathematicians preferred to view computers as technological embodiments of abstract mathematical concepts
  - This approach has had (and still has) a fundamental influence on software technology

Edsgar W. Dijkstra:

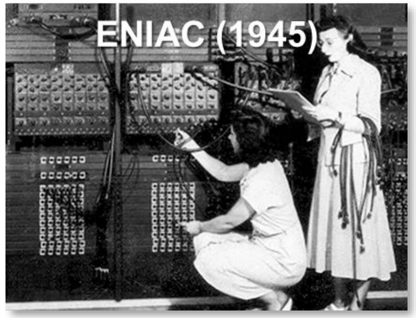
> *"I see no meaningful difference between programming methodology and mathematical methodology."* (EWD 1209)

~~engineering~~

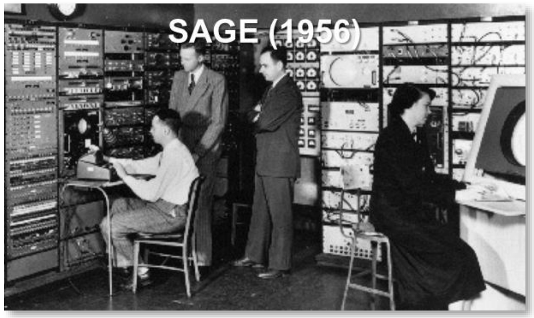
The First Engineering Applications



Codebreaking



Ballistics Table Computations



Air Defense
(Real time)

- **Application to monitoring and control of *real-world phenomena***
  - *A qualitative shift away from the mathematical view of computers and computing towards the engineering domain*

Actually, the real justification for interrupts was to provide a means for computers to detect and respond in a timely fashion to changes in their physical environment

Edsgar W. Dijkstra:

"[The interrupt] was a great invention, but also a Pandora's Box… essentially, for the sake of efficiency, concurrency [became] visible… and then all Hell broke loose." *(EWD 1303)*

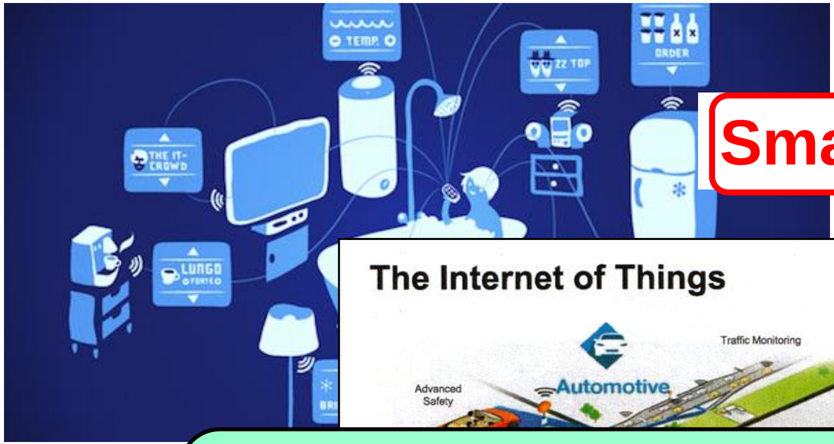
The Consequences

- **This mathematical bias has led to a situation where many of our core software technologies are not well suited to engineering type applications**
  - Most major computer languages only support abstract numerical types (integer, real, Boolean…) but not physical value types (pounds, liters, seconds, etc.)
    - E.g., Mars Climate Orbiter disaster: due to metric vs. Imperial data type incompatibility
  - Most major theories of computation assume that computations are instantaneous
  - Even real-time OS schedulers use abstract concepts such as priority for scheduling real-time tasks
  - …and, so on
- **However, today's trend is a greater turning towards engineering-type applications**

Part 2:

Current Trends in Software-based Systems

The Trend of 21st Century Software…



Self-driving Smart Cars

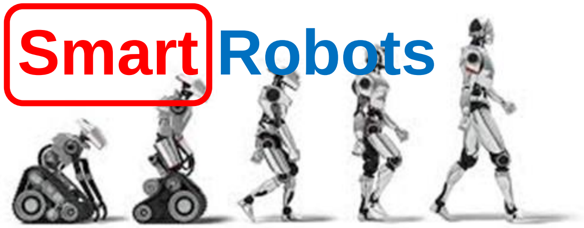
Smart Homes

"Smart Anything Everywhere"

-- Official slogan of EU's H2020 research initiative (https://smartanythingeverywhere.eu/)

Smart Cities

Smart Robots

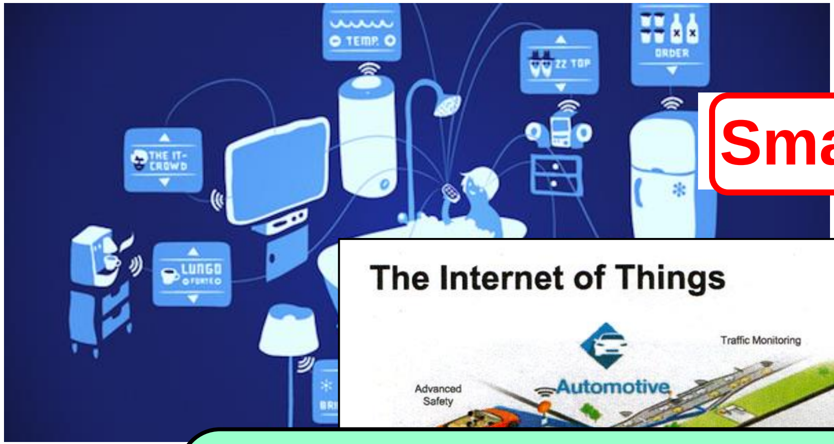


The Trend of 21st Century Software…

MONASH University

Self-driving Smart Cars

Smart Homes



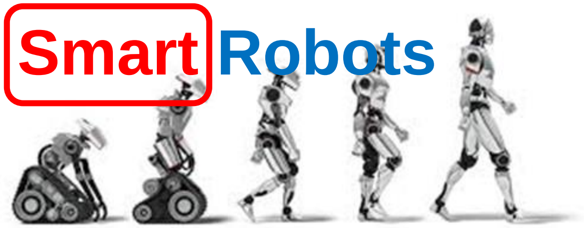
"Smart Anything Everywhere"

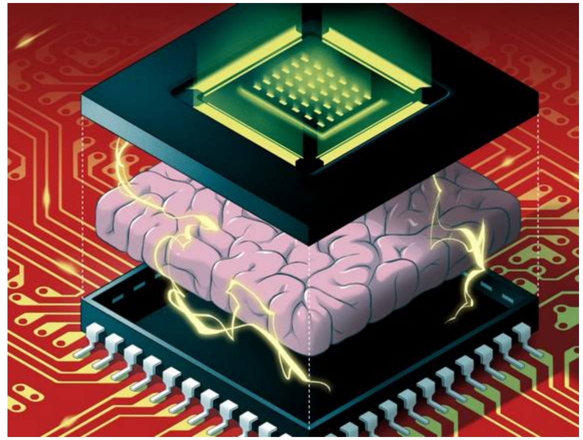
-- Official slogan of EU's H2020 research initiative (https://smartanythingeverywhere.eu/)

What exactly do we mean by "smart"?

Smart Cities

Smart Robots

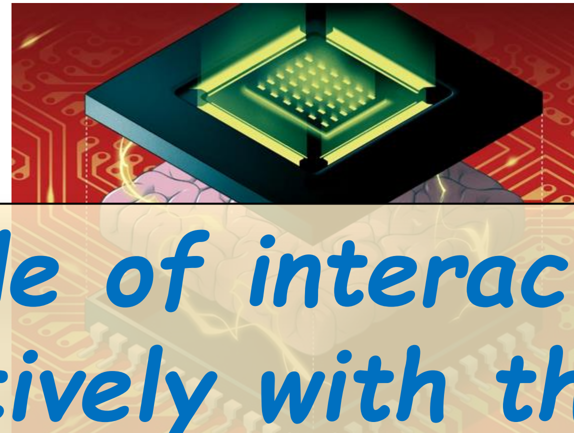




- **A system that can:**
  a. <u>Sense</u> the state of its context and detect relevant changes in that context or in its own internal state, *as they occur* (i.e., *in real time*)
  b. <u>Respond</u> to such changes in a timely manner in a way that is consistent with or conducive to its intended purpose
  c. <u>Adjust</u> its behavior to deal with previously unknown or unexpected situations based on available data and/or its history

***NB: my definition**

MONASH University



Capable of interacting effectively with the "real" (i.e., physical) world

- A system that can:
  a. Sense the state of its context and detect relevant changes

Key Question:

Do we currently have the know-how to design and build such "Smart" systems in a reliable and systematic manner?

*NB: my definition

Part 3:
An Illustrative Example

Case Study: Hi-Tech Parking Brake (1)

Traditional parking brake ($$$)

replaced by

Pushbutton (¢)

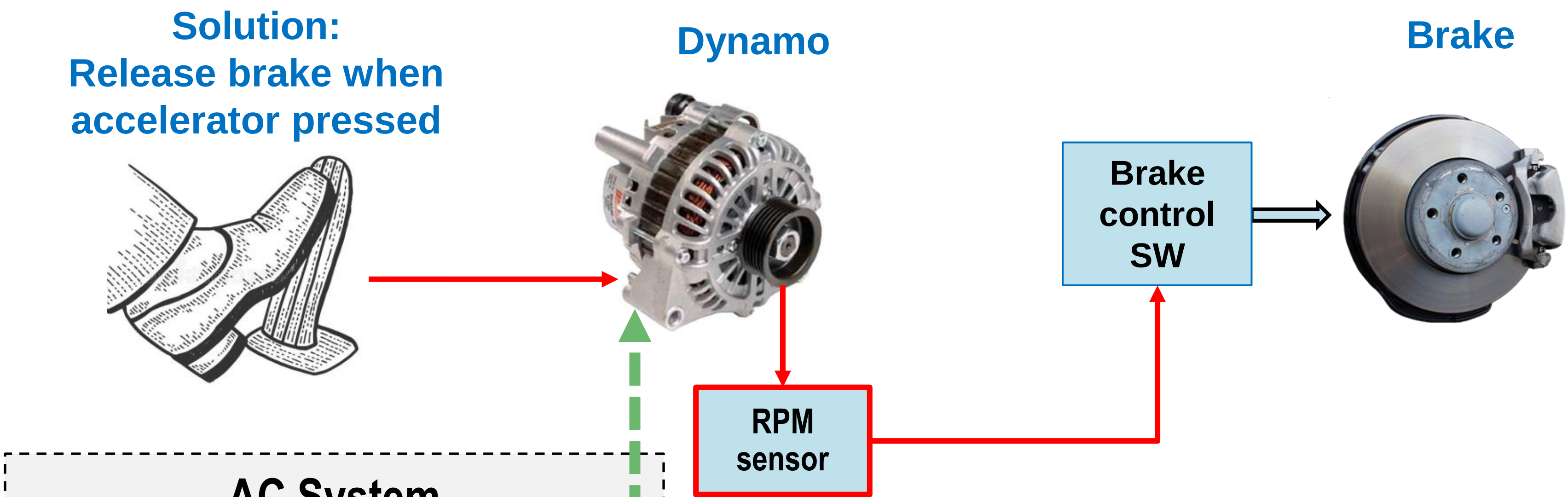
Brake control SW

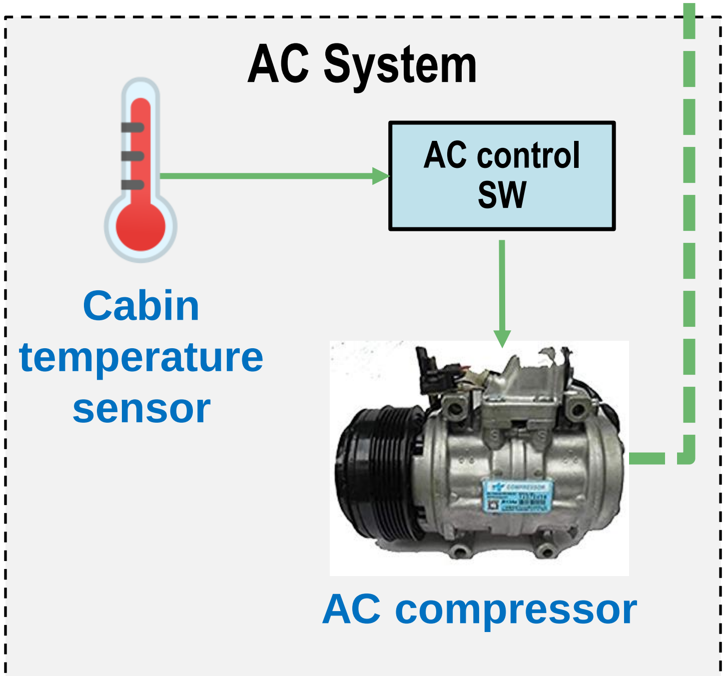
Brake

But, how should we release the brake?

Case Study: Hi-Tech Parking Brake (2)

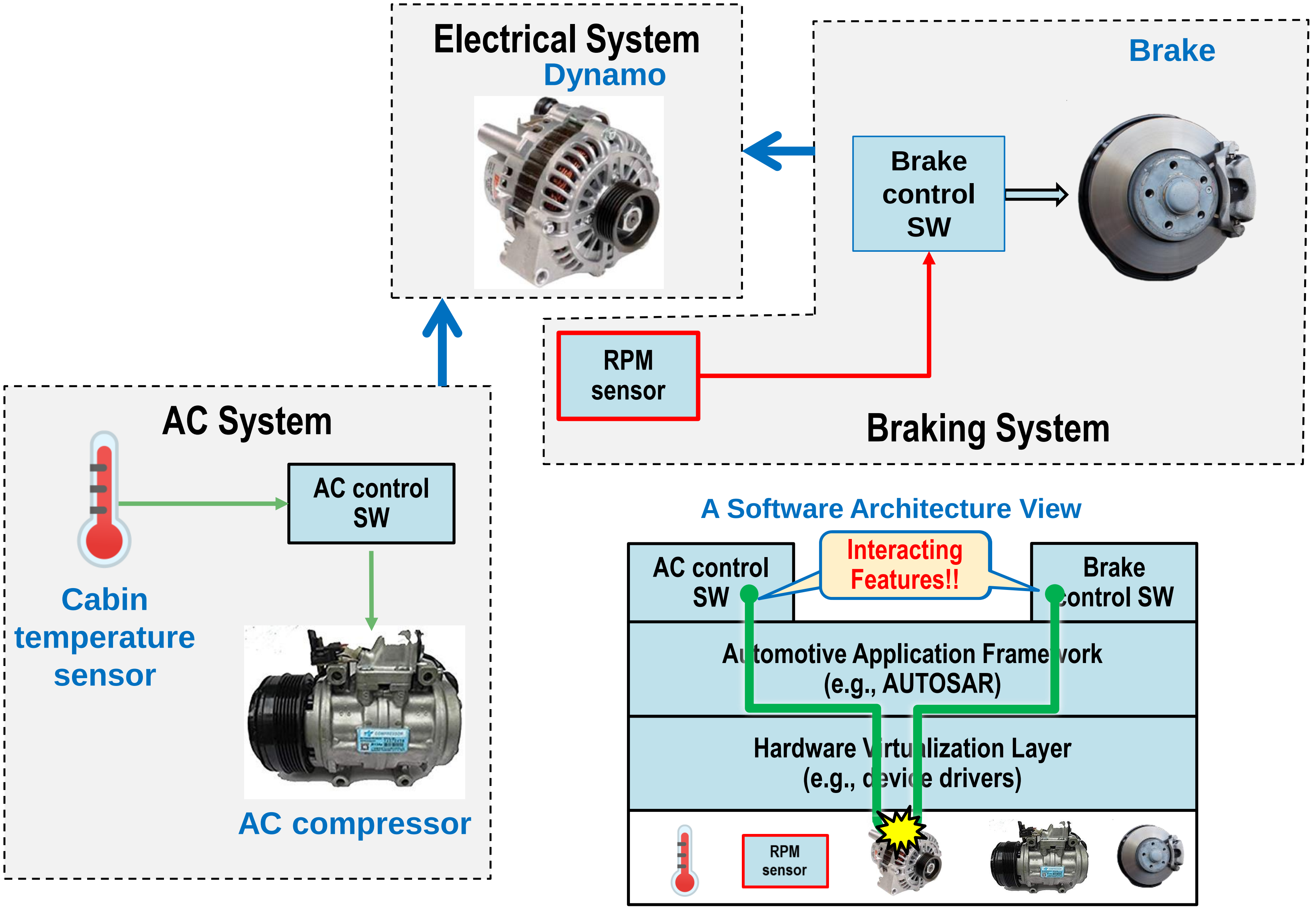
MONASH University



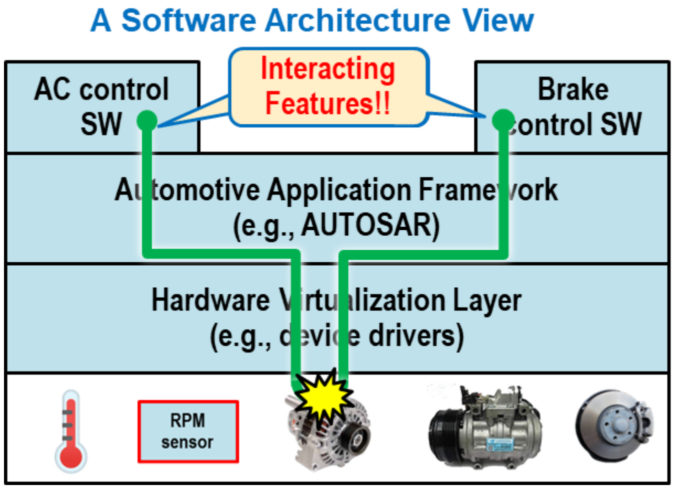


1. On a hot day, the driver *enabled the parking brake* and exited the vehicle temporarily – *leaving the door open*
2. High external temperature quickly raised cabin temperature
3. AC control software activated AC compressor
4. This activation created additional demand for electrical power
5. Dynamo suddenly increased its RPM rate
6. *Brake released; car started moving with no driver present!*

"Divide and Conquer"?



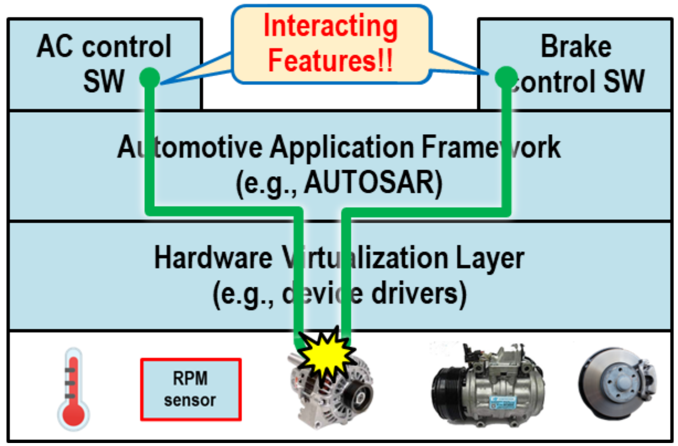
The Feature Interactions Problem



- Feature interactions may occur when two or more feature executions inadvertently interfere with each other, resulting in an undesirable outcome
- Necessary conditions:
  - A hazardous precondition (initial state): combination of system and environment states that has the potential to cause feature interactions
  - Shared resources: One or more system or environment resources that are shared by interacting/concurrent feature executions
  - Temporal overlap: A particular interleaving of action steps belonging to different feature executions leading to at least one of them producing an undesirable outcome
    - *But, only some interleavings can cause feature interaction (time dependent, state dependent)*

The Feature Interactions Problem (cont.)

A Software Architecture View

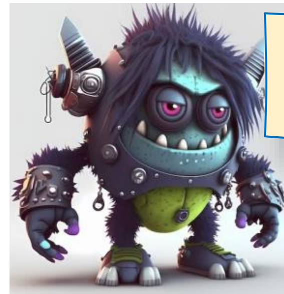


- **What makes feature interactions highly problematic:**
  - The source and cause of the conflict are not always obvious (i.e., difficult to anticipate)
  - The interacting features are often specified independently of each other
  - In feature rich systems, this can result in an unmanageable combinatorial explosion of possible feature interaction scenarios
    - E.g.: In classical telephony, these were in the order of 10^4
    - *How many can we expect to find in something as complex as a "Smart City"?*

⇨ ***It is safe to conclude that in these kinds of complex systems it will never be practically feasible to identify, in advance, all possible feature interactions that can (and, invariably, will) occur***

The Real World: HERE BE DRAGONS

MONASH University



Complexity

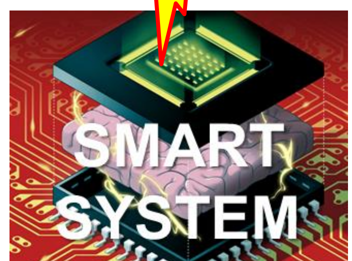
Large size + intricate structure and behavior

Concurrency

A well-known design challenge for software

Unpredictability

Uncertainty about what can happen and when



Idiosyncrasy

Instances of the same class can behave very differently

Mutability

Dynamically changing characteristics

Physical Constraints

Limits dictated by laws of physics

Part 4:

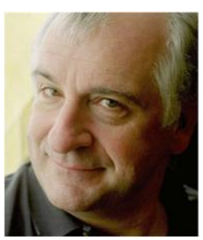
The Reality-Software Relationship

- **Do you happen to know the interesting story behind this number?**

42

  - **It is the answer to: "Life, the Universe, and Everything"**

(Sadly, the exact wording of the question has been lost)



Douglas Adams: *"The Hitchhiker's Guide to the Galaxy"*

Back to Our Teaser…

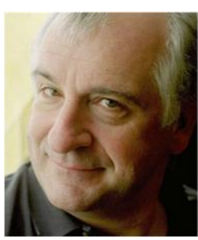
- Do you happen to know the interesting story behind this number?

42

So? What does this have to do with software?

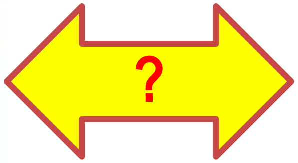
The only entities that computers manipulate directly are <u>numbers</u>!

(Sadly, the exact wording of the question has been lost)

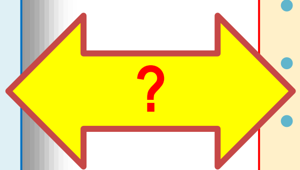


The meaning (semantics) of those numbers are captured (partly) in the software code

MONASH University



- Continuous (i.e., non-digital)
- Informal (Gödel's theorem?)
- Complex (heterogeneous)
- Dynamic/mutable
- Unpredictable/chaotic



- Discrete (digital)
- Mathematically formal (logic)
- Comprehensible
- Static (hardware base)
- Deterministic

MONASH University

?

How much information is lost when we model reality using a computer?

- Continuous (i.e., non-digital)
- Informal (Gödel's theorem?)
- Complex (heterogeneous)
- Dynamic/mutable
- Unpredictable/chaotic

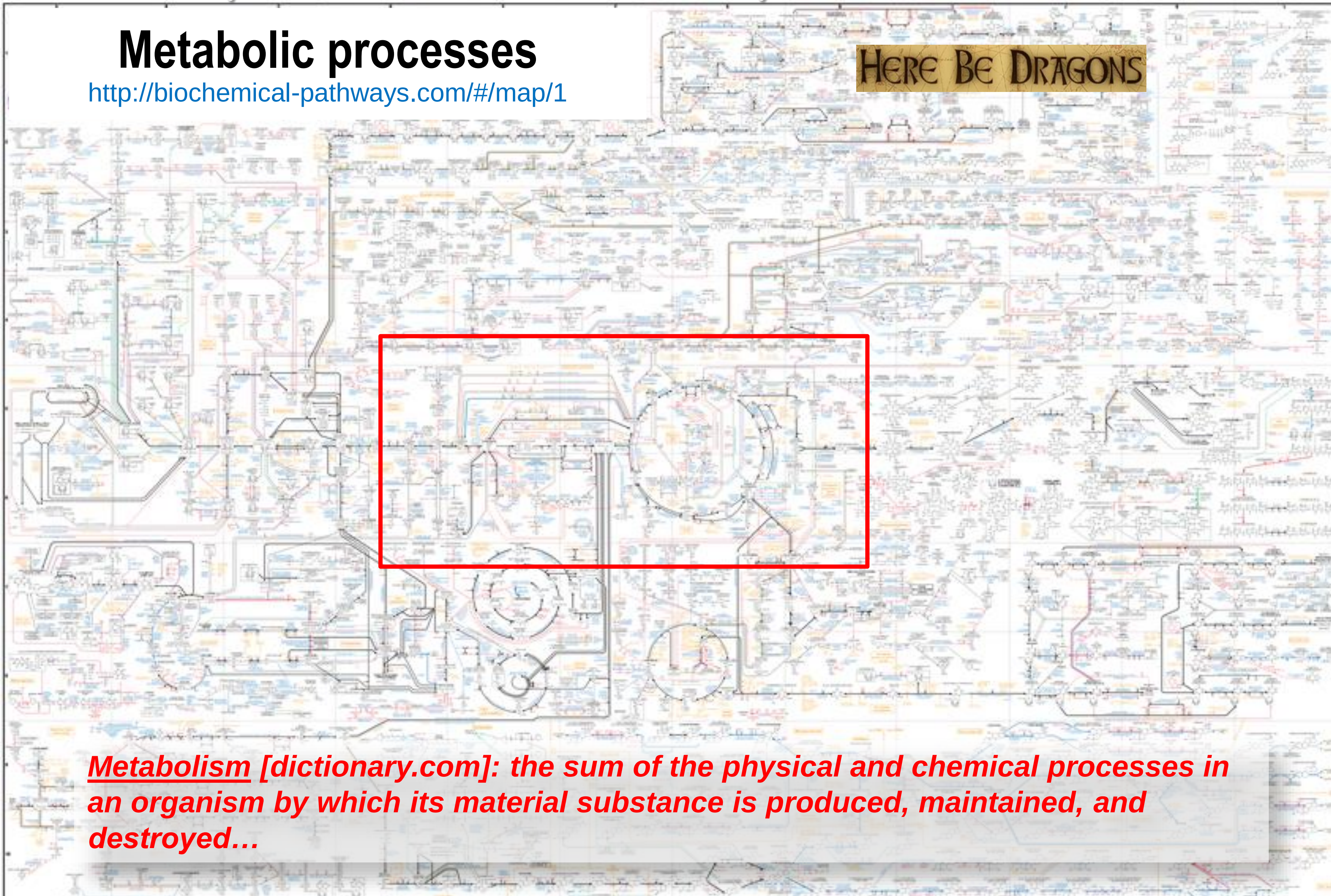
?

- Discrete (digital)
- Mathematically formal (logic)
- Comprehensible
- Static (hardware base)
- Deterministic

MONASH University

Metabolic processes

http://biochemical-pathways.com/#/map/1



<u>Metabolism</u> [dictionary.com]: the sum of the physical and chemical processes in an organism by which its material substance is produced, maintained, and destroyed…

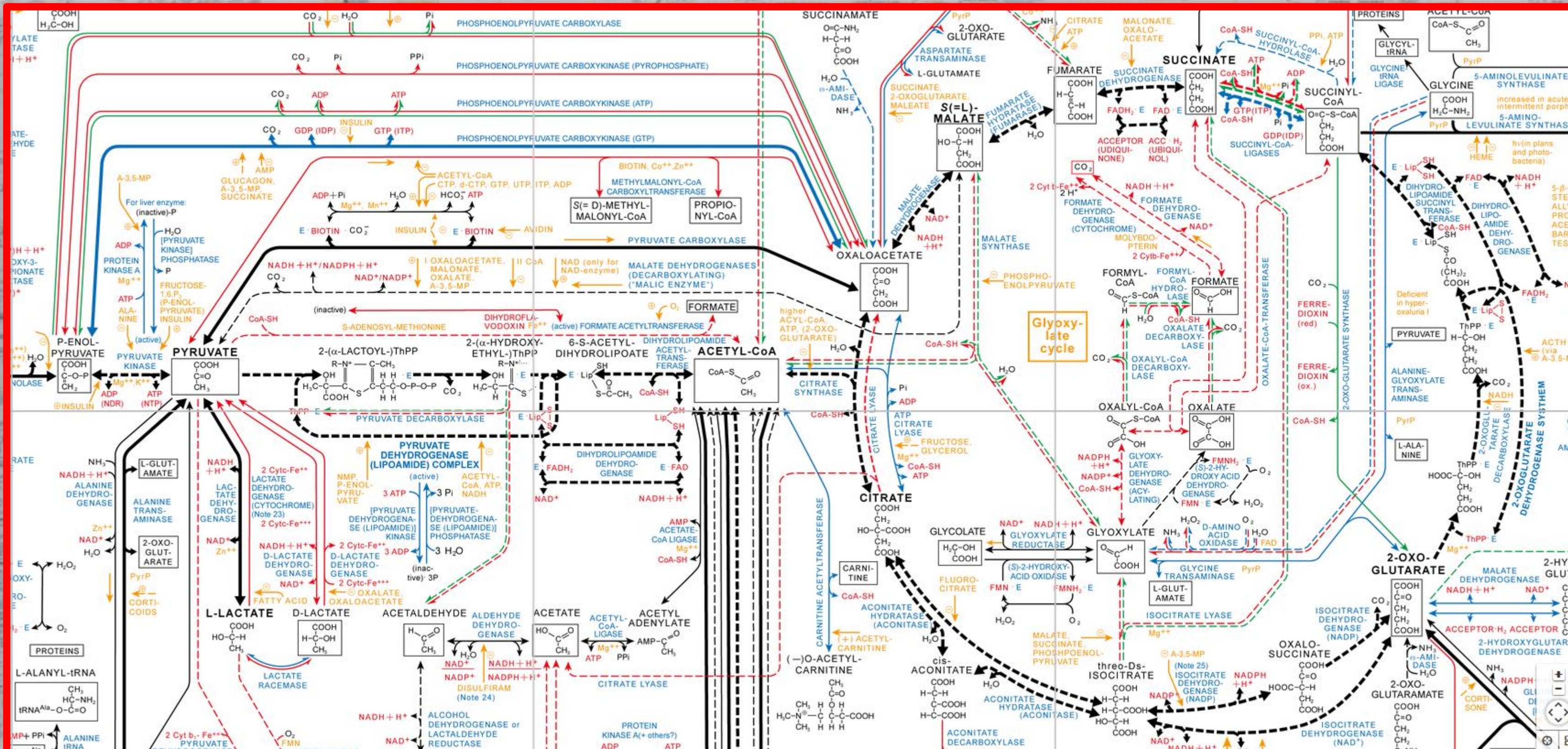
The Complex Nature of Reality

MONASH University

Metabolic processes

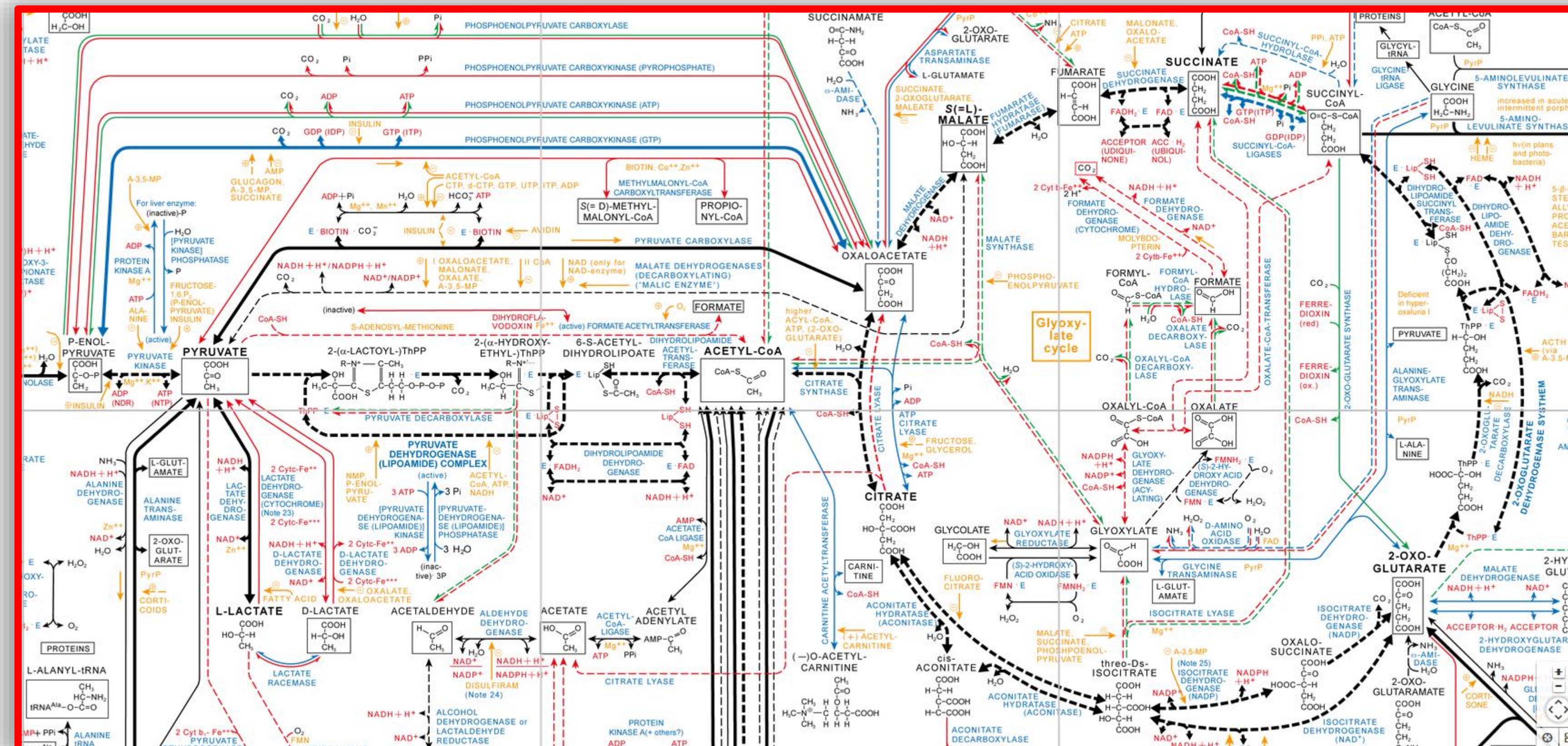
http://biochemical-pathways.com/#/map/1

HERE BE DRAGONS



Some Key Observations

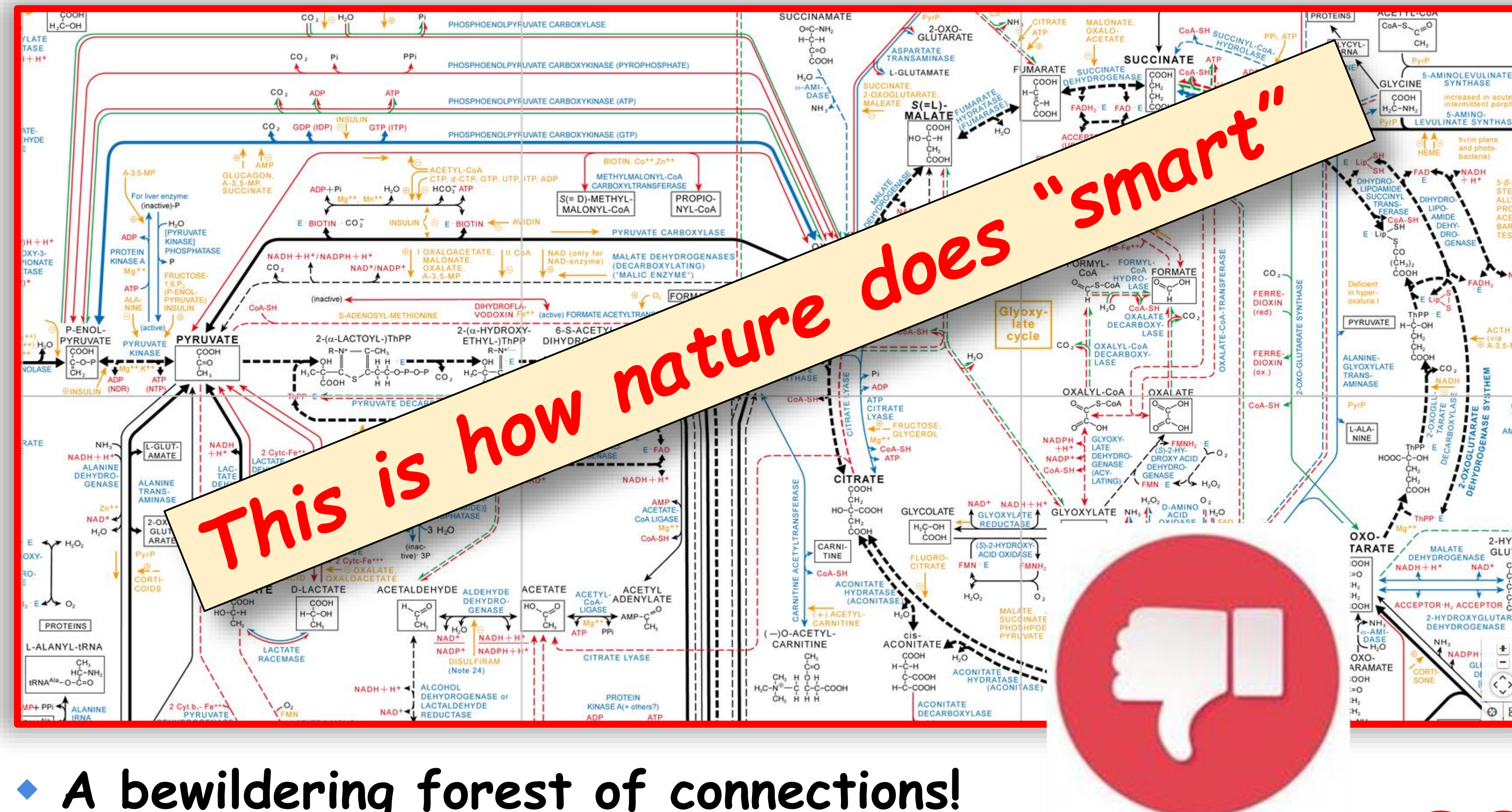
MONASH University



- A bewildering forest of connections!
- No clear or crisp modularity

Some Key Observations

MONASH University



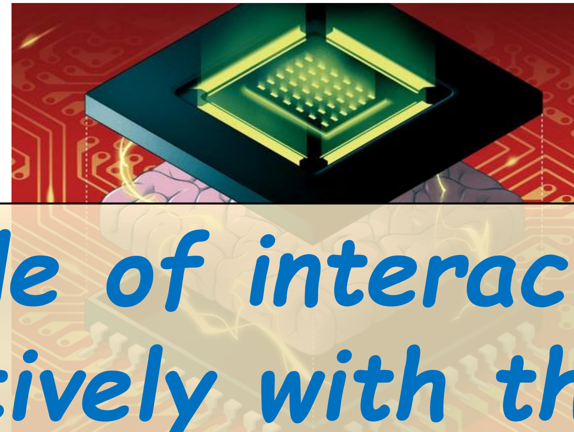
- A bewildering forest of connections!
- No clear or crisp modularity

Bad design!??

Part 5:

Our Current Arsenal for Complex ("Smart"?) System Design

MONASH University



Capable of interacting effectively with the "real" (i.e., physical) world

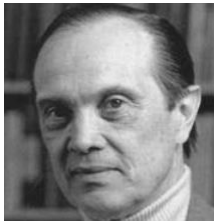
- A system that can:
  a. Sense the state of its context and detect relevant changes in that context or in its own internal state, as they occur

Key Question:

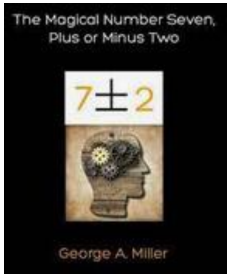
Do we currently have the know-how to design and build such "Smart" systems in a reliable and systematic manner?

*NB: my definition

Miller's Magic Number 7 (plus or minus 2)



G. A. Miller



The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information[1]

George A. Miller (1956)
Harvard University

- **A thesis dealing with the limits of human cognition**
  - An average human can keep track of a maximum of 7 ± 2 items in short-term memory
- **Inspired further psychological research into other cognition limits:**
  - Rapid enumeration of number of objects ("subitizing"): limit of 4
  - etc.

Edsgar W. Dijkstra:

"... as a slow-witted human being I have a very small head and I had better learn to live with it and to respect my limitations and give them full credit."

The Technique of "Chunking"

- **A method of overcoming the "7 ± 2" limit**
  - E.g.: 12128254767 versus 1-212-825-4767
  - An application of the old "divide and conquer" method, applied recursively
  - In essence, it is a form of ***abstraction***

    "the act of considering something as a general quality or characteristic, apart from concrete realities, specific objects, or actual instances" **[Dictionary.com]**

- **Abstraction has been recognized as *an essential skill in system design* (software or otherwise):**

Abstraction – the key to Computing?

Jeff Kramer

Department of Computing, Imperial College London
j.kramer@imperial.ac.uk

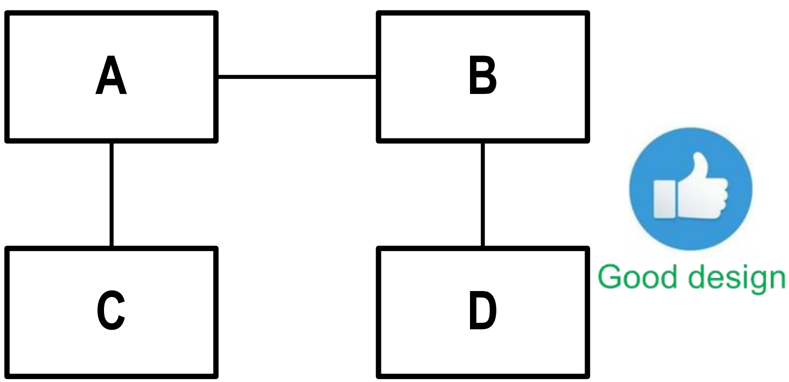
Comm. of the ACM 50(4): 36-42

The KISS* Principle of Design

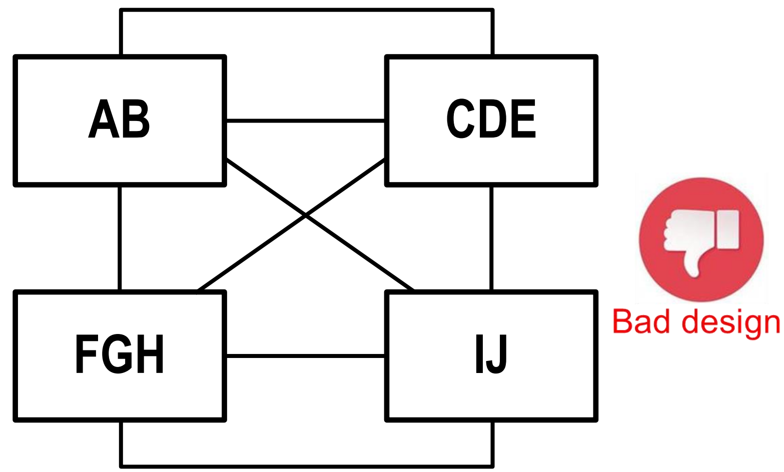
Edsgar W. Dijkstra:

"Simplicity is a prerequisite for reliability"

"The art of programming is about organizing complexity"



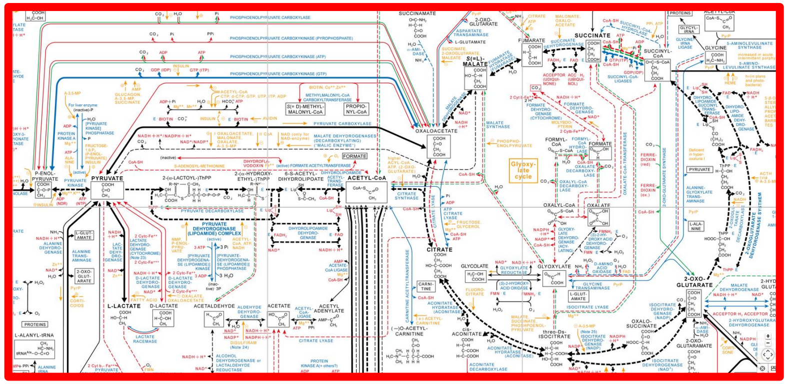
- *Each module performs a single well-defined function/feature ("Divide-and-Conquer" approach)*
- *Inter-module couplings are minimized*



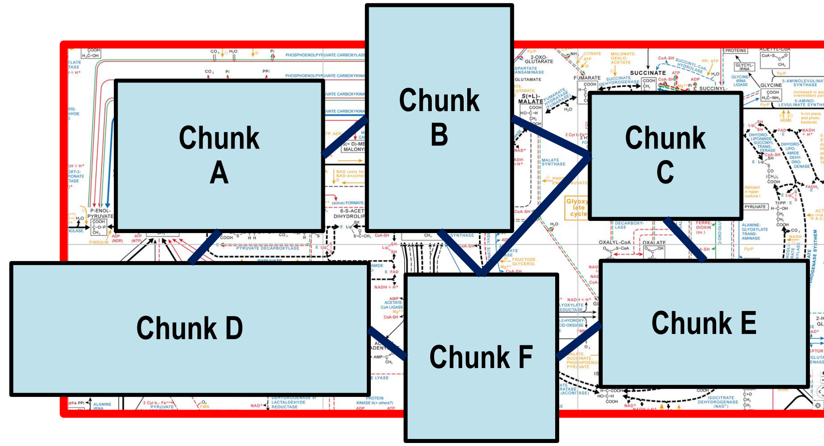
- *Modules combine multiple functions*
- *Large number of inter-module couplings*

(*) Keep It Simple Stu**d

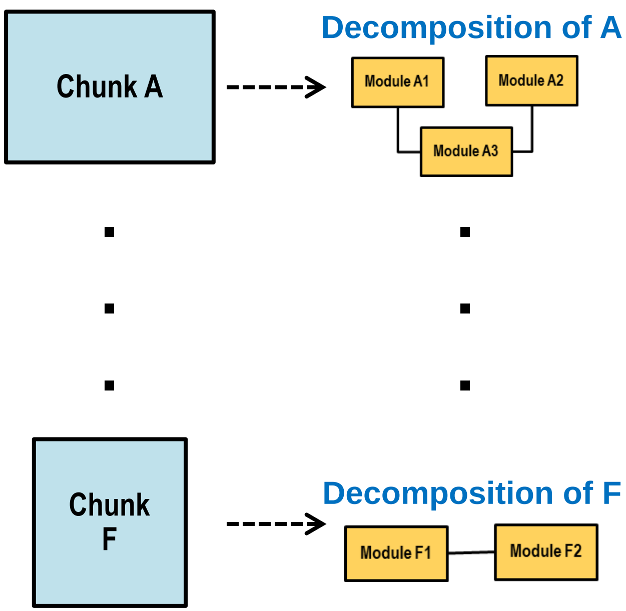
Traditional “Divide and Conquer” Modeling



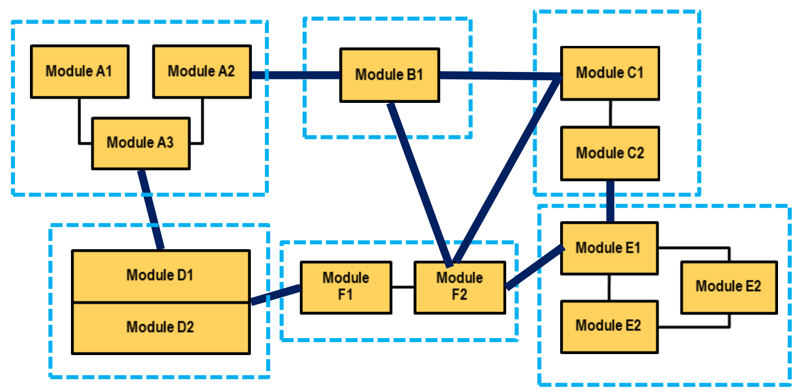
Step 0: A Complex System



Step 1: Partition and reduce (abstraction)



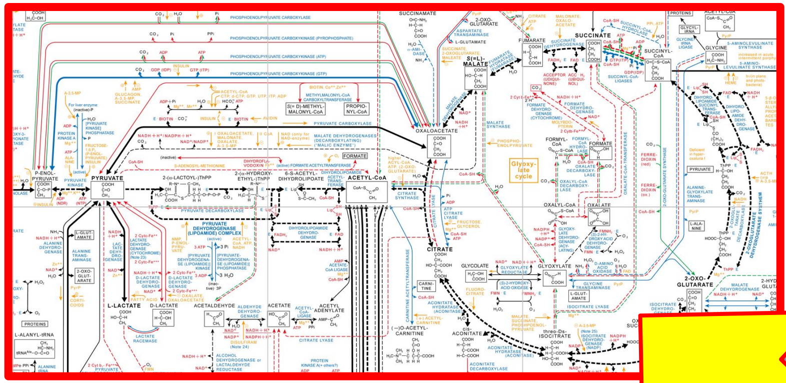
Step 2: Recurse



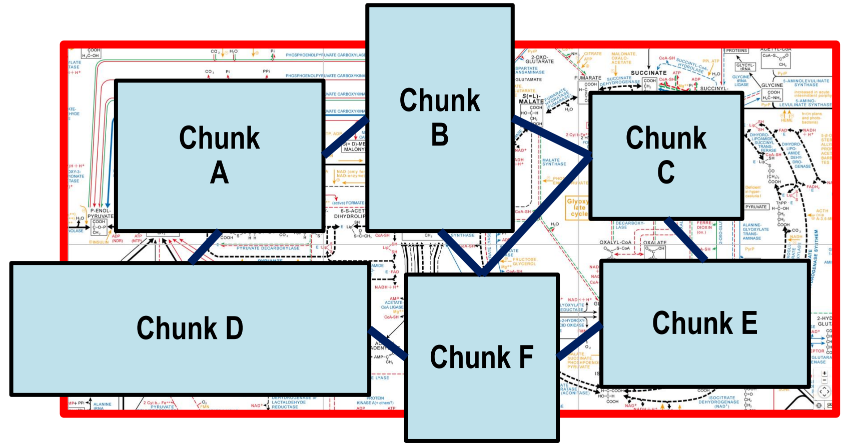
Step 3: Reassemble

Traditional "Divide and Conquer" Modeling

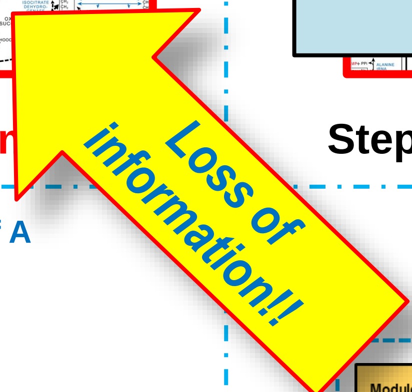
MONASH University

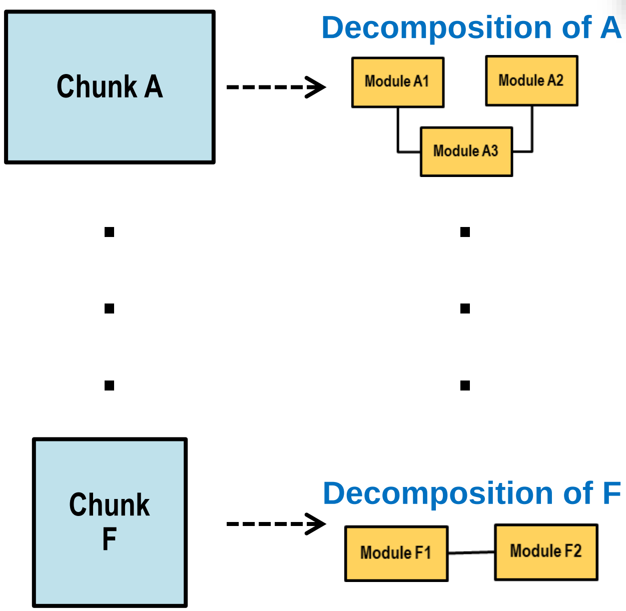


Step 0: A Complex System

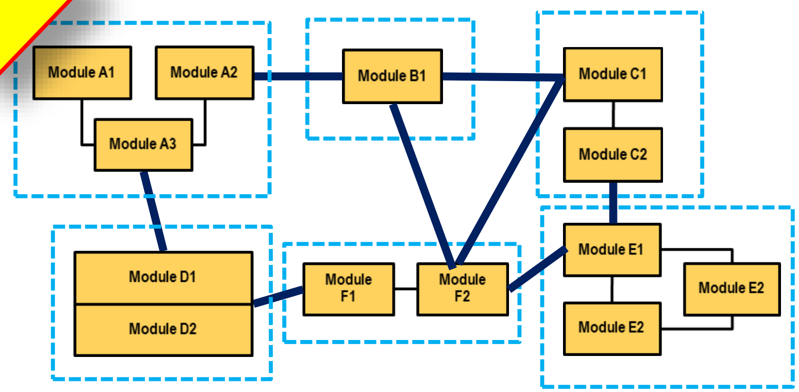


Step 1: Partition and reduce (abstraction)





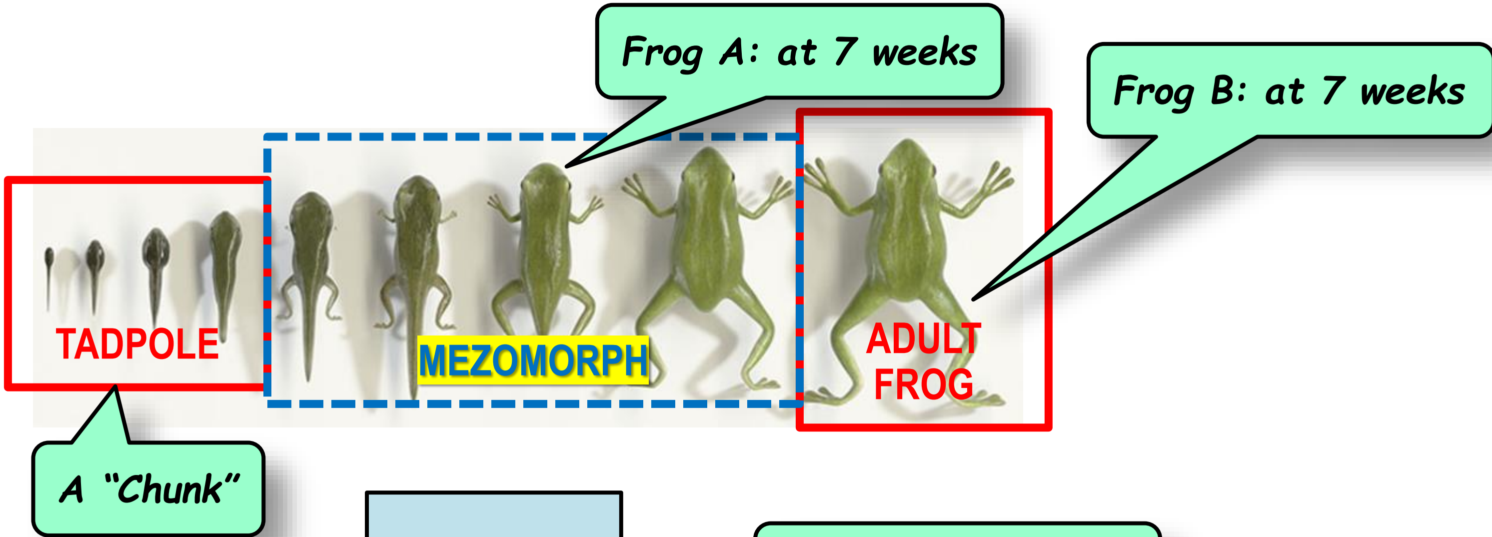
Step 2: Recurse

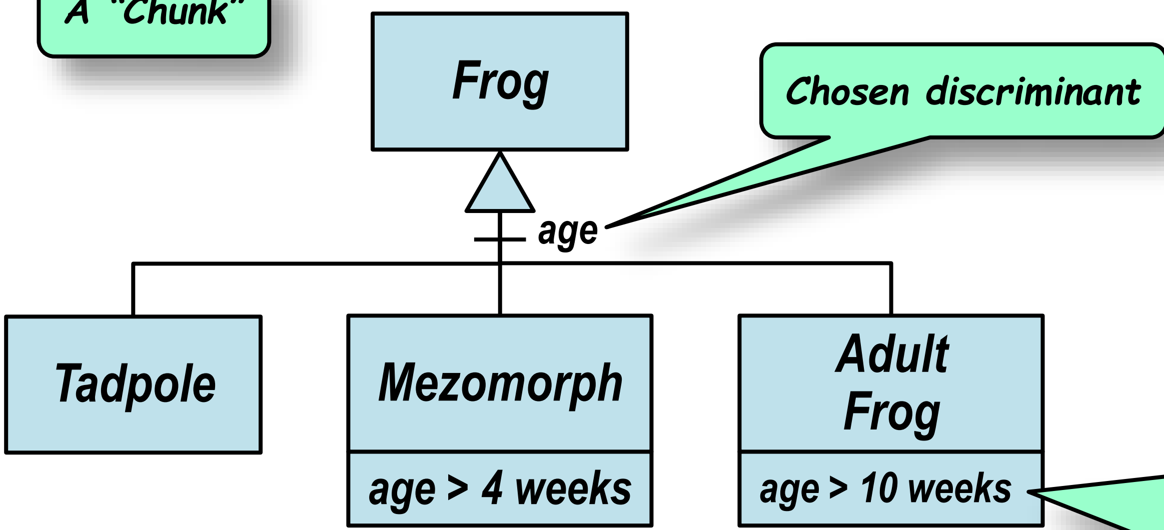


Step 3: Reassemble

MONASH University

- A *continuous* and *idiosyncratic* dynamic process





Information lost!

⇒ Classification is an imperfect and subjective approximation of reality, whose primary objective is to help us cope with complexity

Example: Modeling the Lifecycle of a Frog

MONASH University

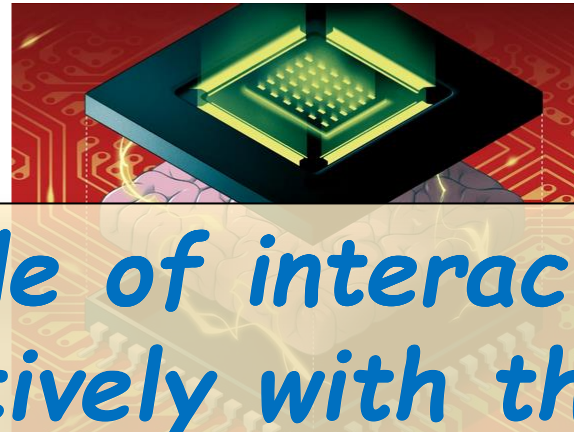
- A more realistic (overlapping) classification

- Not directly supported by any conventional OO language
- Requires a different approach to classification

B. Selić and A. Pierantonio: “Fixing Classification: A Viewpoint-based Approach, ISOLA 2021

What These Two Examples Suggest

- The parking brake problem illustrates a category of design issues due to the inherent complexity of the real world, which do not lend themselves readily to exhaustive analysis
  - ⇒ We are unlikely to have the ability to design systems that will incorporate ready-made solutions to all possible eventualities that might arise in reality
- The dynamic and "fuzzy" classification problem points out that our current technologies are based on an idealized (mathematical?) view of reality and, hence, inadequate
  - … and that, perhaps, given the semantic gap between these technologies and reality, we may never be able to fully bridge the gap



Capable of interacting effectively with the "real" (i.e., physical) world

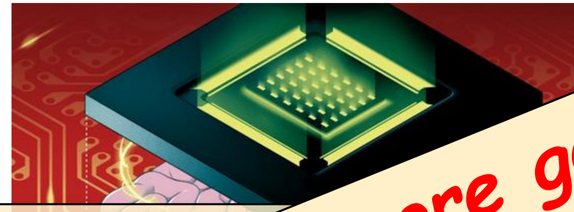
- **A system that can:**
  a. Sense the state of its context and detect relevant changes in that context or in its own internal state, as they occur

Key Question:

Do we currently have the know-how to design and build such "Smart" systems in a reliable and systematic manner?

*NB: my definition

MONASH University



Capable ... eff...

- A...

...levant changes

... as they occur

...tion:

...y have the know-how to d... build such "Smart" systems i... a reliable and systematic manner?

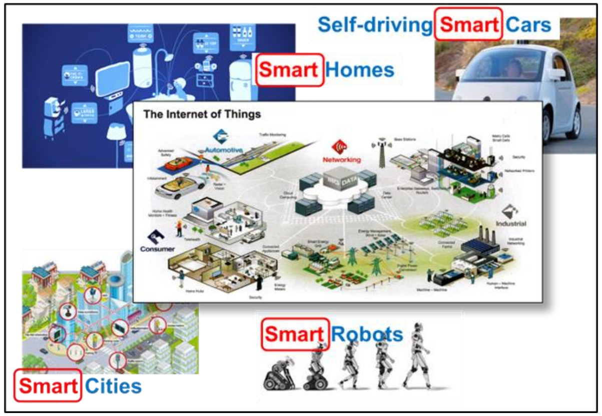
The cited examples and, more generally, the overall experience suggest that the answer to the Key Question is that <u>we do not yet have a good handle on how to design reliably and consistently the "smart" systems that we desire</u>.

*NB: my definition

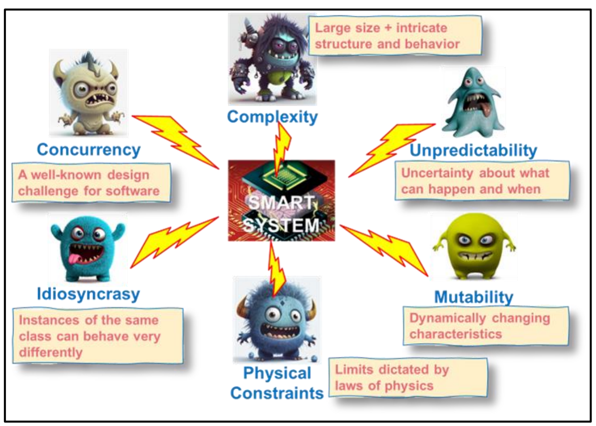
Part 6:

What Can We Do?

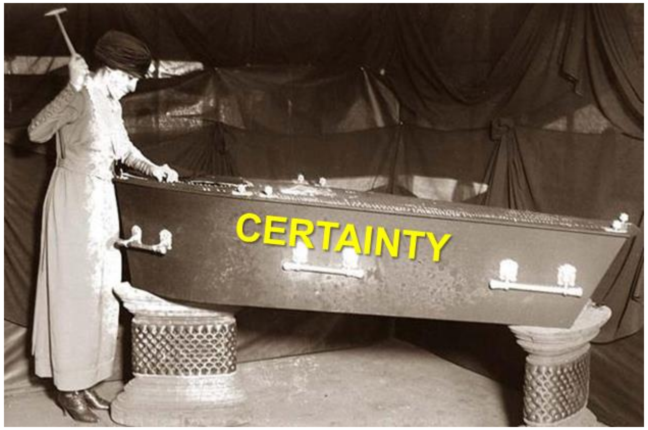
The Loss of Certainty!!



+



implies



D.Garlan's Research Program for Dealing with Uncertainty

David Garlan

D. Garlan, "Software Engineering in an Uncertain World", *Proc. FSE/SDP Workshop on Future Software Engineering Research, (125-128) 2010.*

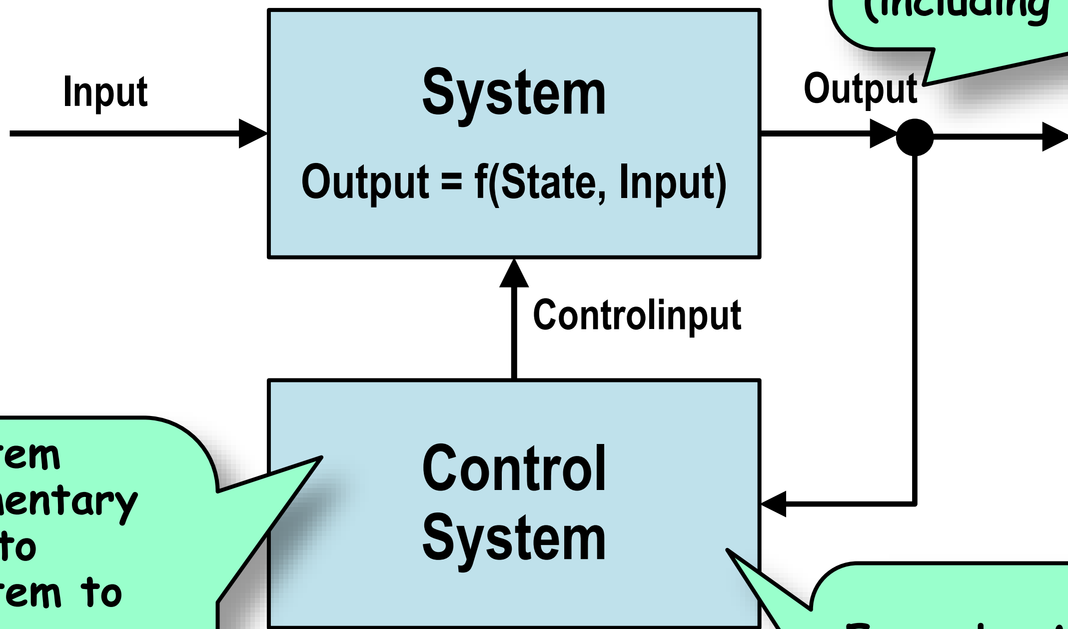
- Utility instead of correctness
- Bounded approximation instead of precision [certainty?]
- Closed-loop systems (⇨)
- Incorporate uncertainty as a first-class design concern
- Resiliency/adaptation in the presence of unpredictable/unexpected events
- New formal methods and tools for reasoning in the presence of uncertainty
- New methods of machine learning that ensure "reasonable" behavior (⇨)

Closed Loop Systems

MONASH University

- **Based on classical feedback control theory**

The system may depart from its desired behavior for any of a variety of known *or unknown reasons* (including bugs)



The control system provides supplementary inputs designed to restore the system to the desired state/behavior *regardless of what caused the erroneous behavior*

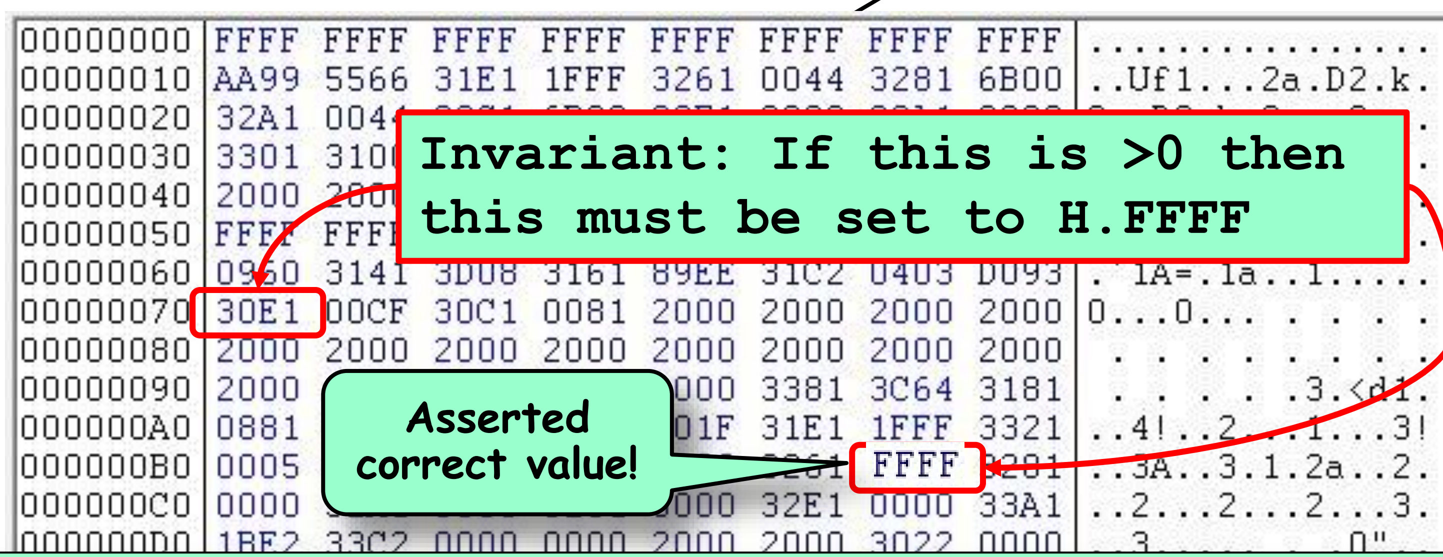
Example: MAPE(K)

(Monitor, Analyze, Plan, Execute) (K = Knowledge)

Feedback Control in Software: Example

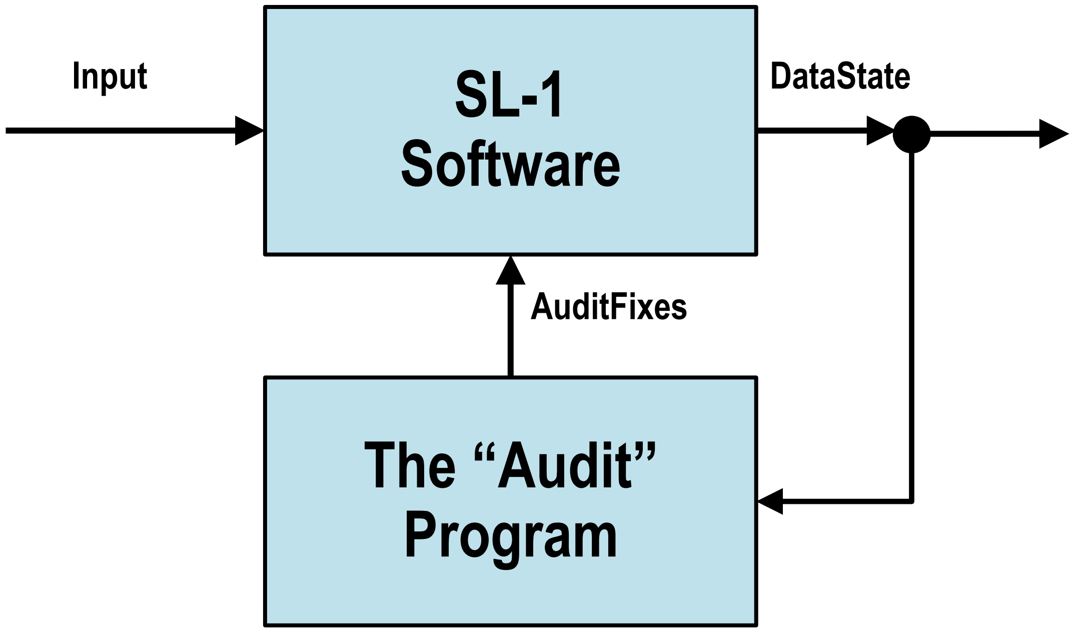
MONASH University

- **[1976] The SL-1 PBX:**
  - Run-time faults had to be detected and fixed in real-time *without service disruption*
- **The "Audit" program**
  - An *independent* memory crawler and data consistency enforcer



This early piece of software manifests a key idea that may be crucial to how we construct software in the future!

- The SL-1 Audit program

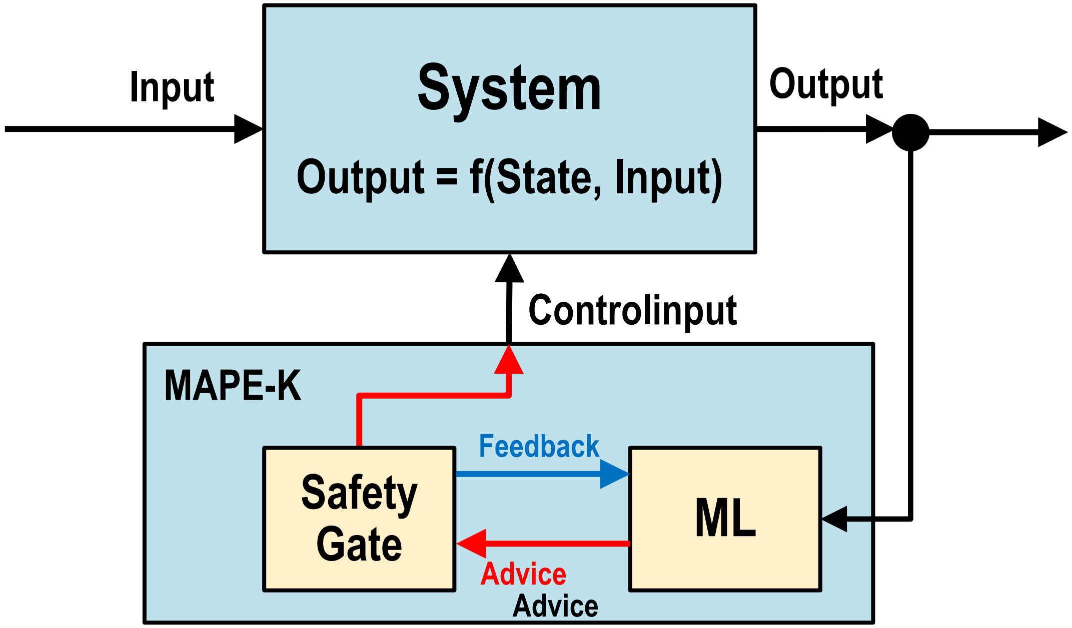


back

The Role of AI Components

MONASH University

- The "K" in MAPE-K provides the adaptation capability through Machine Learning (ML)
- But AI/ML is itself characterized by uncertainty!
  - Its outputs may or may not be appropriate to a given situation
- A possible architecture for dealing with this:



Biomimicry as Inspiration

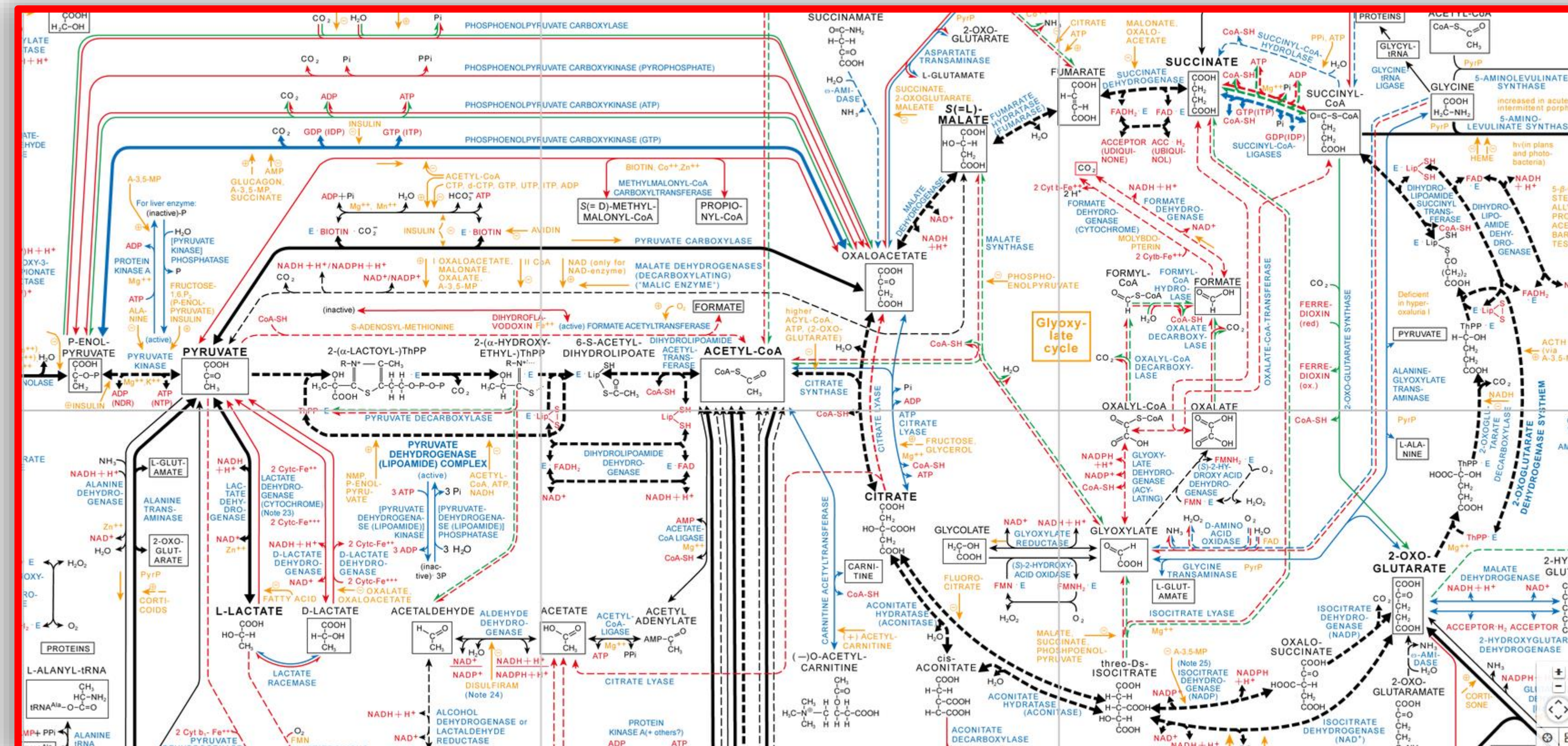
- **Neural networks are inspired by mimicking key features of how "smart" biological systems work**
- **Biomimicry [https://computingforsustainability.com]:**

  Biomimicry is a design discipline that studies nature's best ideas and then imitates these designs and processes to solve human problems. Studying a leaf to invent a better solar cell is an example of this "innovation inspired by nature."

- **Can biomimicry help drive software engineering research?**

  The core idea is that nature, imaginative by necessity, has already solved many of the problems we are grappling with. Animals, plants, and microbes are the consummate engineers. They have found what works, what is appropriate, and most important, what lasts here on Earth. This is the real news of biomimicry: <u>After 3.8 billion years of research and development, failures are fossils, and what surrounds us is the secret to survival.</u>

MONASH University

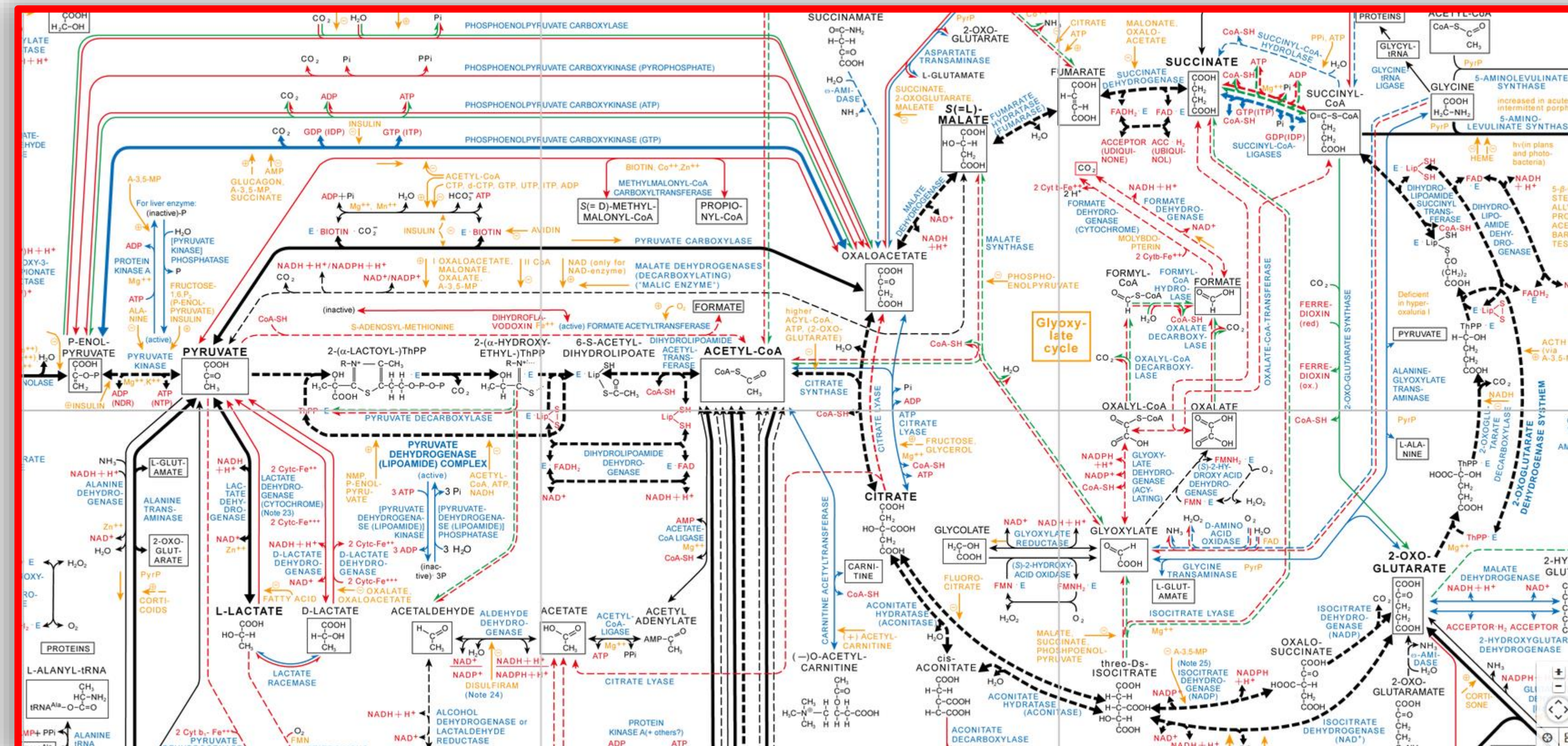


- A bewildering forest of connections!
- No clear or crisp modularity

Example: What Can We Learn From This?

MONASH University

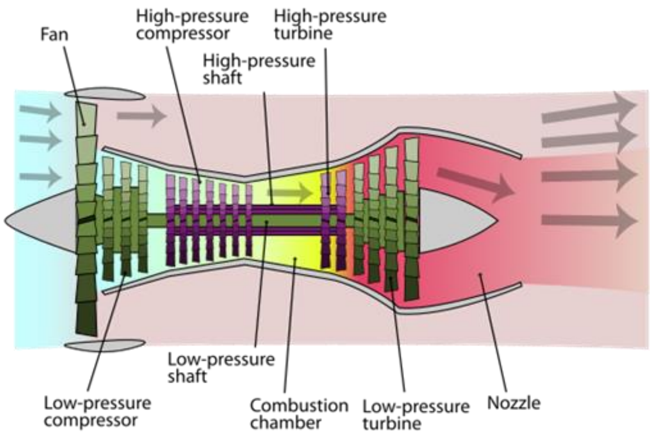
- *Should we perhaps turn our attention from the boxes to the lines?*



- **A bewildering forest of connections!**
- **No clear or crisp modularity**

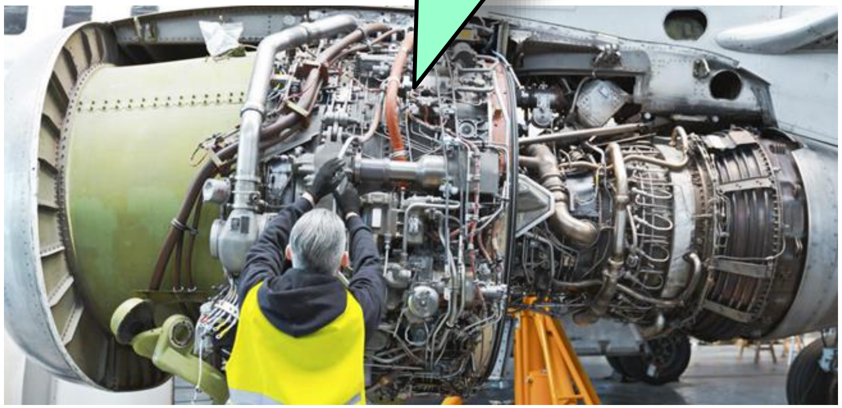
MONASH University

- Jet engine evolution



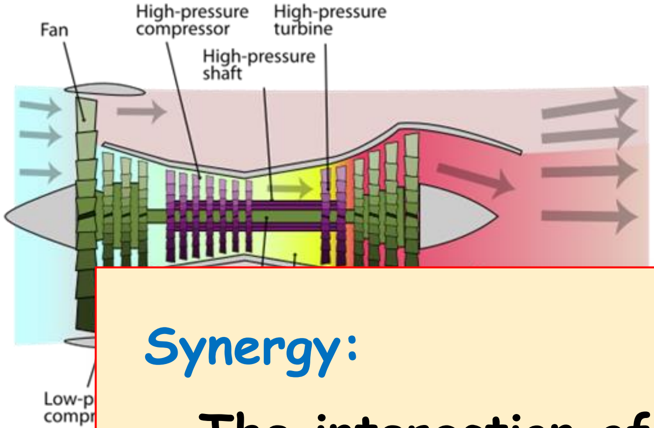
Basic Jet Engine (simple)

Large number of *synergistic* feedback and feedforward iconnections



Modern Jet Engine (complex but efficient)

- Jet engine evolution

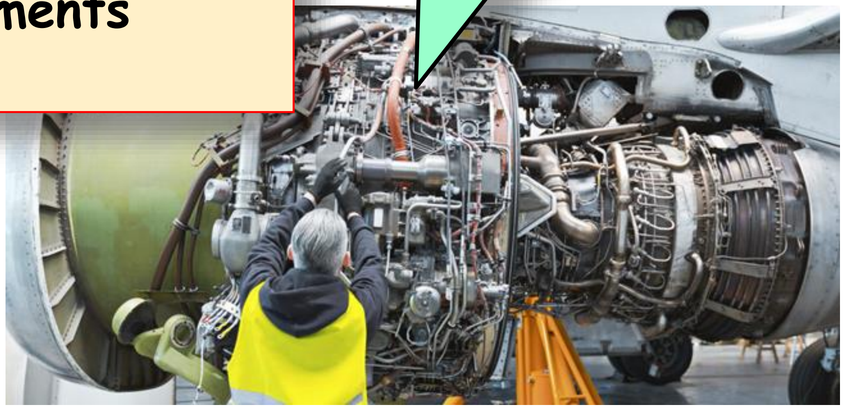


Basic Jet Engine (simple)

Synergy:

The interaction of elements that, *when combined*, produce a total effect that is greater than the sum of the individual elements

Large number of *synergistic* feedback and feedforward iconnections

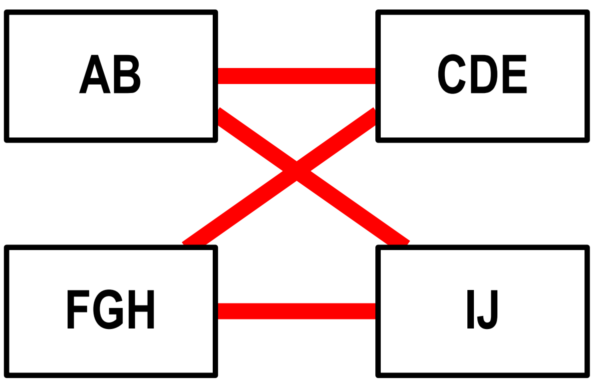


Modern Jet Engine (complex and efficient)

Time for a New Approach to Design?

- *We cannot expect to match the complexity of natural systems*
- But…

Good design?



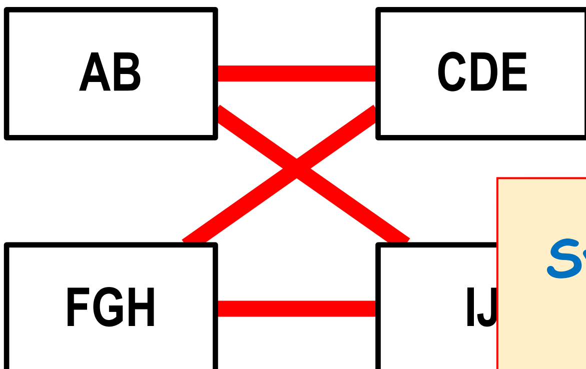
RQ: *Are there meaningful and useful design patterns based on synergistic relationships that can be discovered and exploited?*

- *Modules combine multiple **synergistic** functions/features*
- ***Synergistic** inter-module couplings*

Time for a New Approach to Design?

- ***We cannot expect to match the complexity of natural systems***
- **But…**

Good design?



- *Modules combine multiple* ***synergistic*** *functions/feat*
- ***Synergistic*** *inter-module couplings*

> ***RQ: Are there meaningful and useful design patterns based on synergistic relationships that can be discovered and exploited?***

System thinking:

After we have divided and successfully conquered, perhaps we should put *effort into putting things back together*!

Summary

1. There is a significant qualitative gap between reality and the essential nature of computer-based systems
2. This presents a major hurdle to our stated desire to construct "smart" systems
3. Our current technologies and established methods are insufficiently powerful to adequately overcome this hurdle
4. While we cannot hope to match the complexity and capabilities of biological systems, they could inspire the necessary technological and methodological advances needed to help us achieve our objectives

Appendix
A Personal Concern

If you can indulge me for a moment, what follows is a *heartfelt suggestion* to my younger colleagues, from an industry veteran…

My Appeal: Think!

MONASH University

- ☒ ChatGPT will *not* do it for you
- ☒ You will *not* find a ready-made solution on GitHub
- ☒ You will not find it by looking at your "smart"phone or PC
  - although those might be useful in the process

√ Instead, you will have to invent something completely new by *thinking deeply* about both the problem and the solution

You are Responsible for the Future

MONASH University

"If you think about any problem long enough, you will almost always find a better solution to it."

-- Ernst Munter, A true engineering master

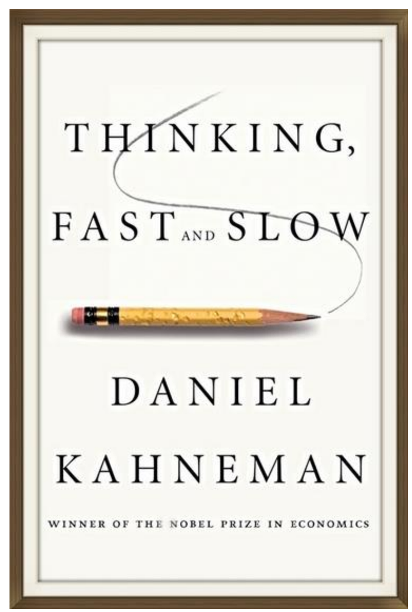
- **The kinds of highly-complex systems you are being asked to design and build are *unparalleled in history***

⇒ *They will require much originality and innovation*

⇒ *…which will require time and large amounts of reflective thinking…*

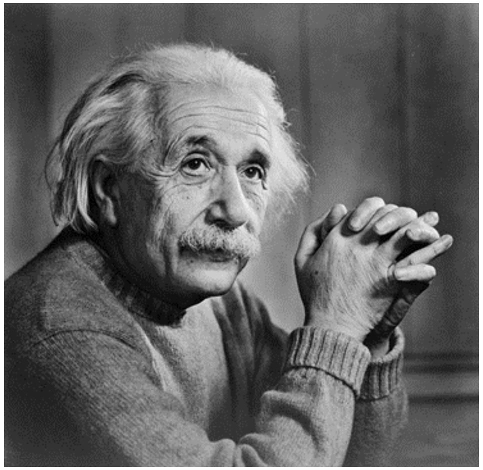
⇒ *…i.e.: "thinking slow"*

⇒ *Recommended reading:*



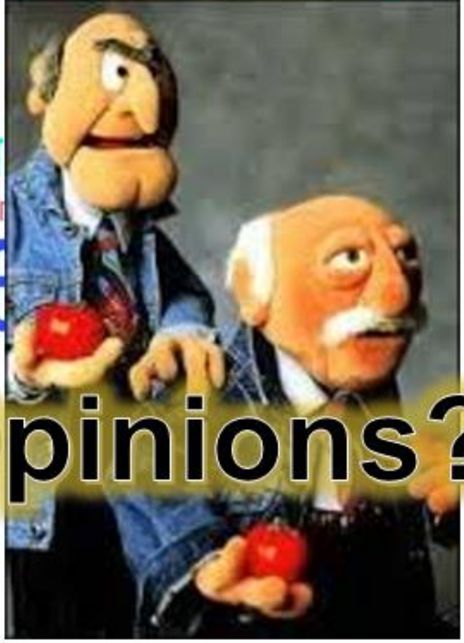
Last but definitely not least, keep in mind:

MONASH University



"Concern for man himself and his fate must always constitute the chief objective of all technological endeavours … in order that the creations of our minds shall be a blessing and not a curse to mankind."

-- Albert Einstein, 1931



Questions?

Opinions?