Complex Systems Thinking

David Slater - Beyond Bows and Arrows

“If you can’t model it, you can’t manage it”

Functional Systems Modelling

SCSCS working Group on Complex Systems

The Challenge

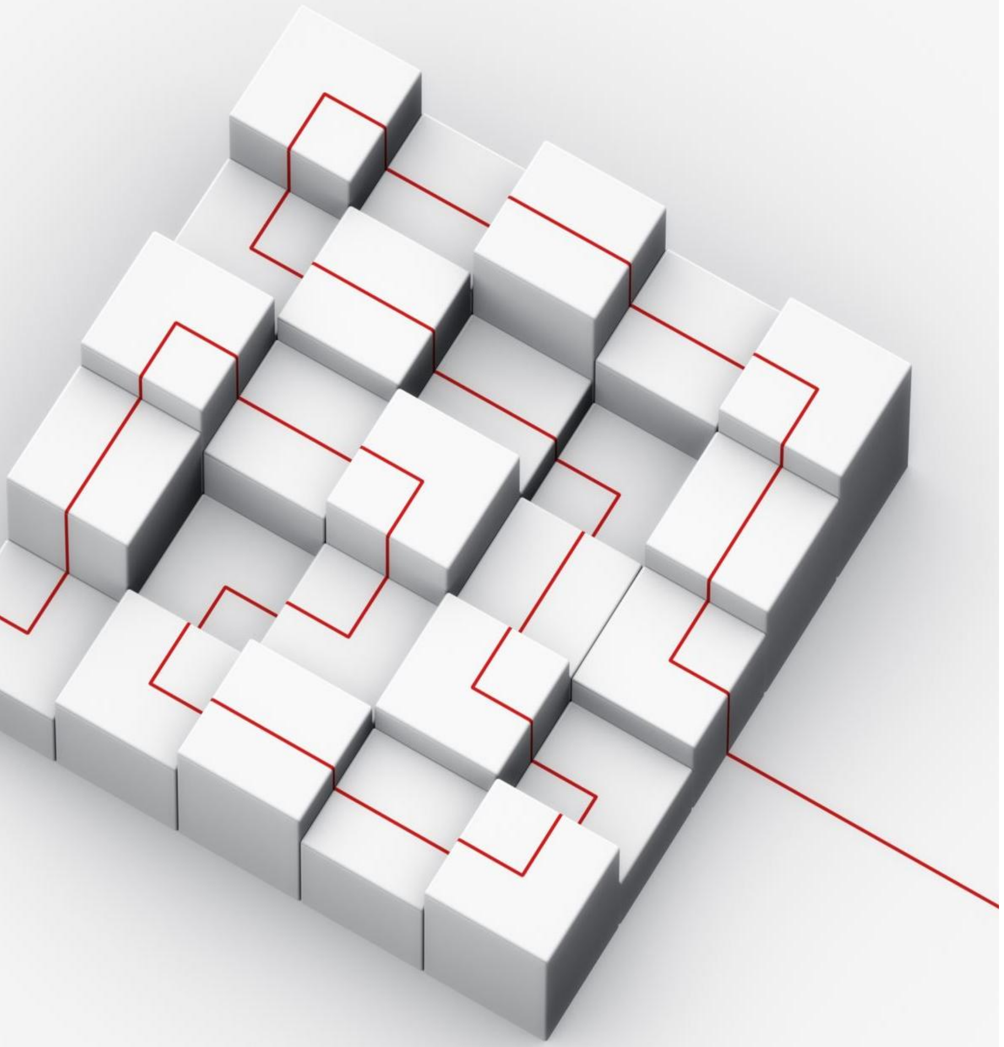
The Hidden Opportunity in Systems Engineering: How UK Industry Can Unlock Its Next Leap Forward

Adam Lancaster Head of Engineering Capability for Systems Engineering (BAE) June 25, 2026

The systems we're building today are **not just more complex, they are more interdependent.** Software, hardware, safety, security, human factors, electromagnetic effects, and operational context all interact in ways that demand stronger integration.

Whether we're designing a next-generation defence platform, a safety-critical nuclear system, a semiconductor manufacturing process, or a high-assurance financial service, the opportunity is the same: to create engineering environments **where clarity, traceability, and integration are built in** from the start.

Objective of this Presentation Explore a Solution!



Three messages

COMPONENTS ARE NOT ENOUGH

- Systems thinking is not component inventory

Knowing the parts gives knowledge — but not necessarily understanding.

FUNCTIONS ARE THE UNIT OF UNDERSTANDING

- FRAM gave functions interactive couplings

It showed how performance emerges from interacting functions, not simple chains.

THE CRITICAL STEP IS EXECUTION

- **FSM (Functional Systems Modelling)** makes functions executable and traceable

(The aim is a generic silo-agnostic approach)

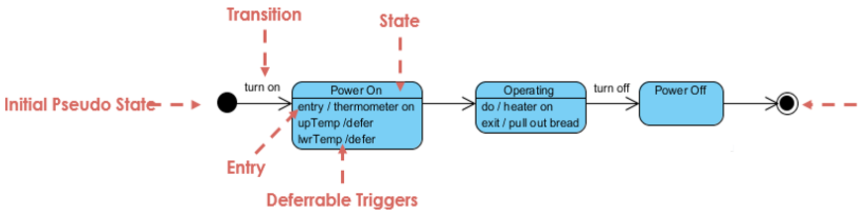
In short - From drawing Bows and Arrows to running functional models.

How does this work then?

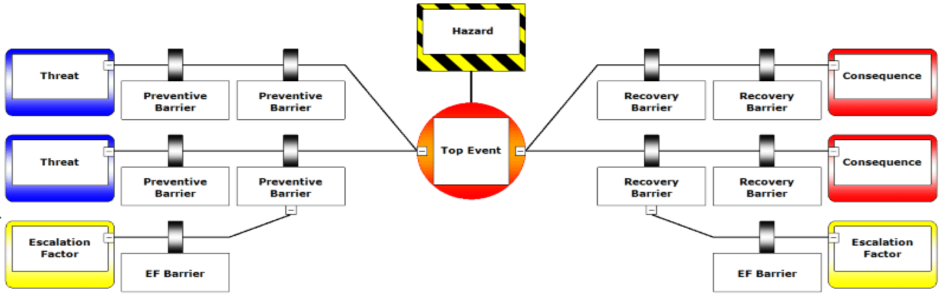
Draw a diagram?

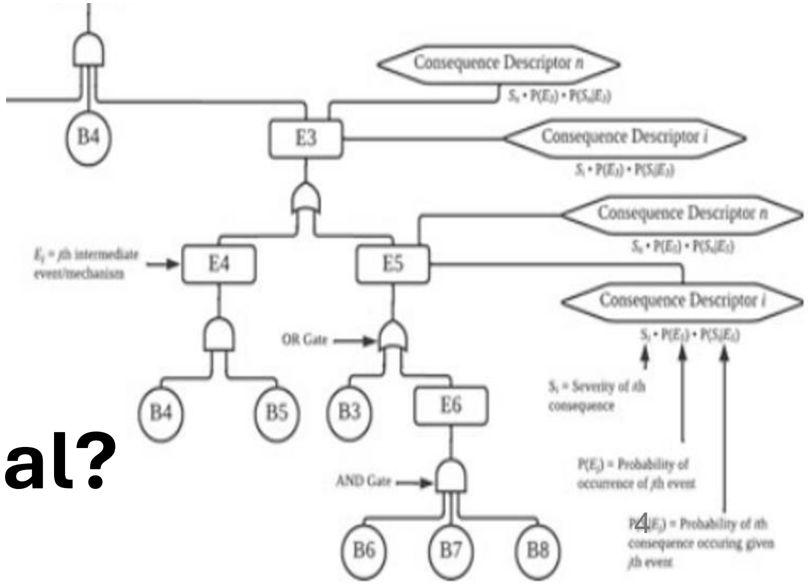
Model it?

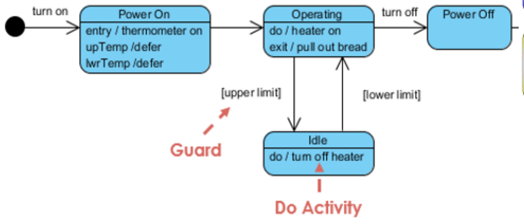
How?



Initial State Machine Diagram







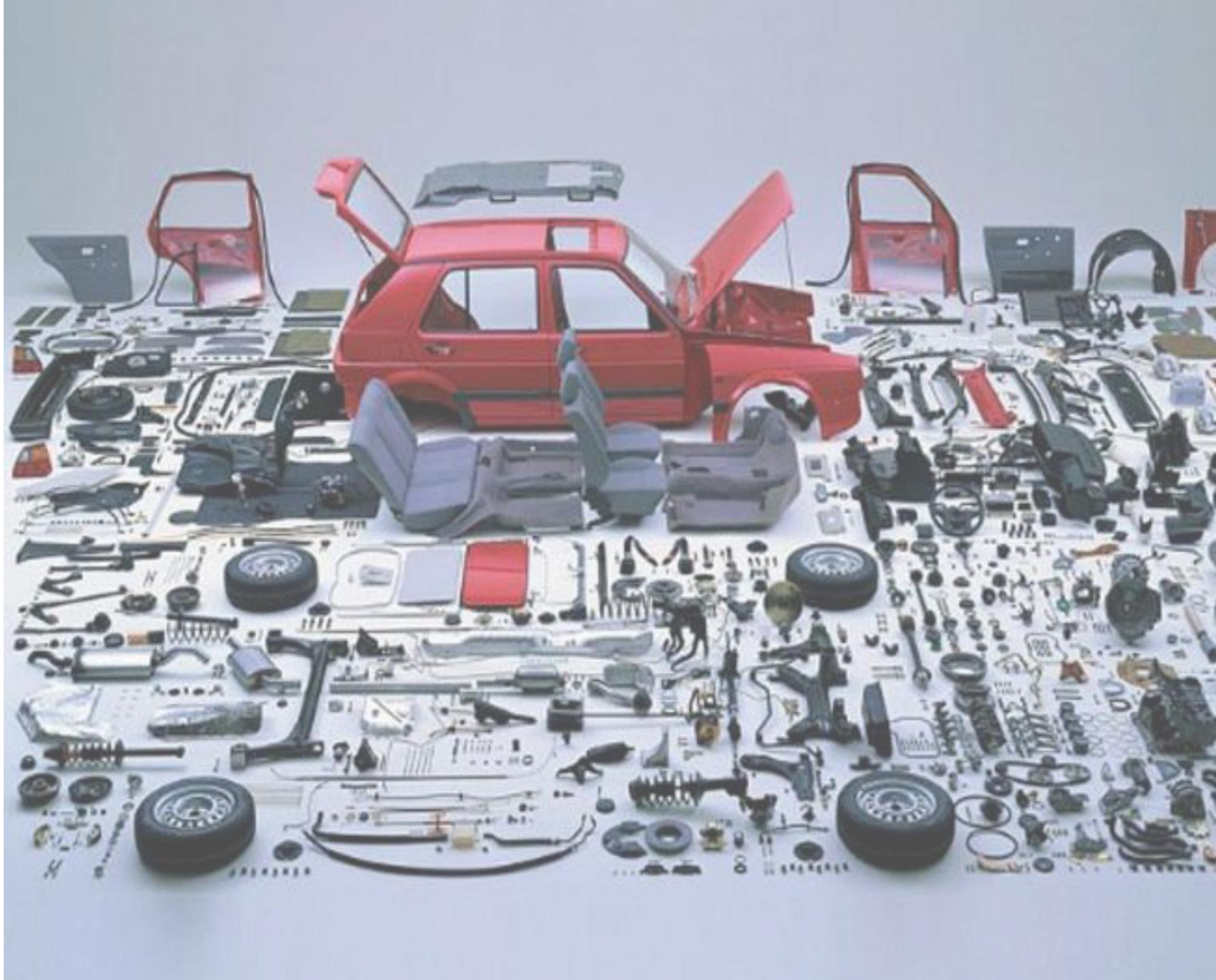
Read the Manual?

Extended State Machine Diagram

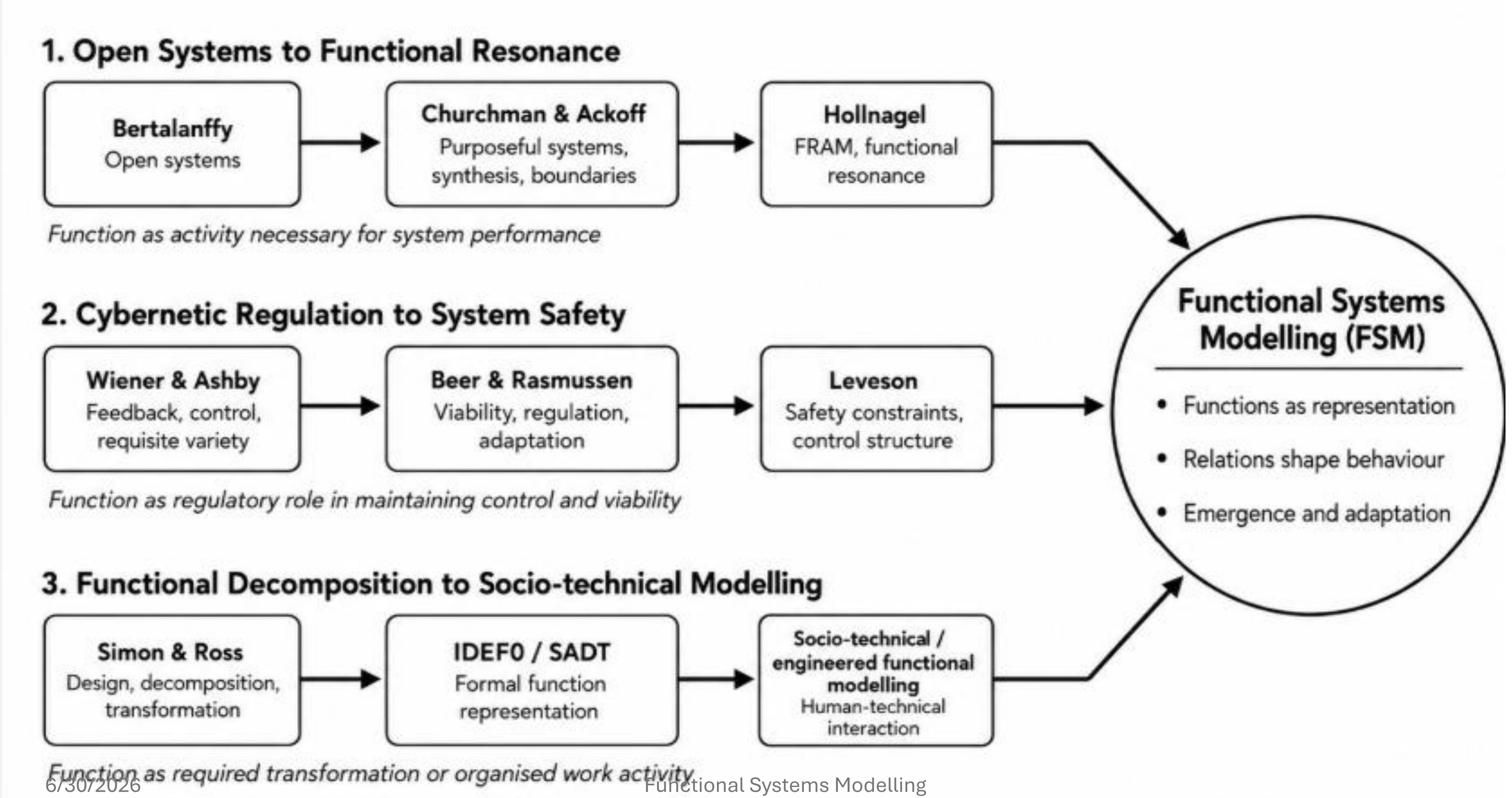
Systems Thinking is not about breaking it down into separate boxes and component parts

Individual Component Data gives you KNOWLEDGE –

But not UNDERSTANDING (Ackoff)



Silos of Systems Thinking



The evolution of systems modelling

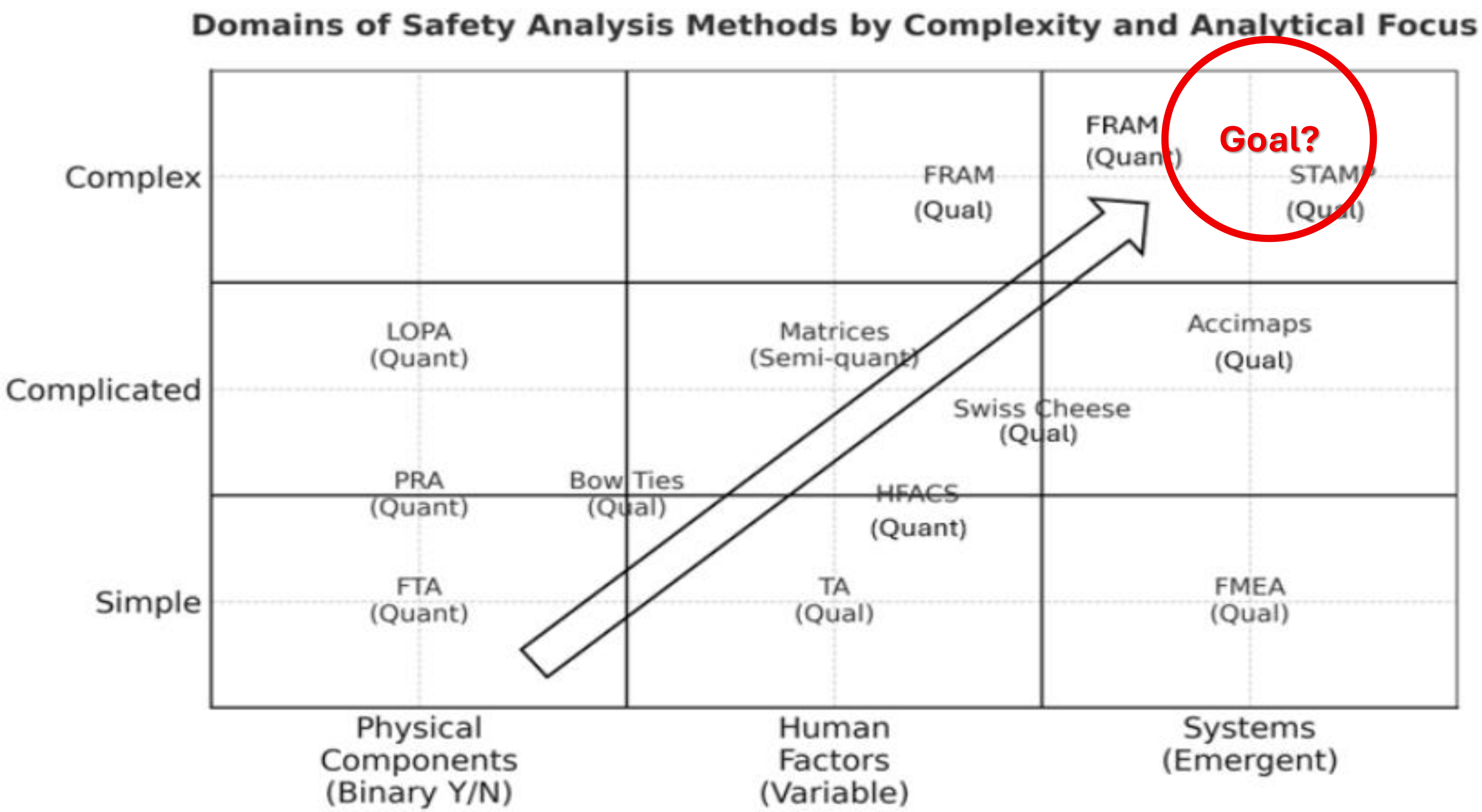


Figure 1. Domains of safety analysis methods by complexity and analytical focus. Methods are positioned according to their epistemic domain—from simple, component-based quantitative analyses (FTA, PRA) to complex, emergent system approaches (STAMP, FRAM)—showing the historical and conceptual evolution of safety methodology.

First, we modelled things.

Then we modelled failures.

Then we modelled functions.

Now we need to model adaptive functional systems in motion.

Things go wrong through the same mechanisms by which they normally go right.

But Qualitative models are just not enough to answer detailed needs

Safety-critical systems increasingly require models that can be queried, run, compared, updated and audited.

Not only "what is connected to what?" but "what happens when this function is late, weak, overloaded, missing, uncertain, degraded or contradicted by experience?"

Useful / useable generic functional modelling is surely not just drawing the model. It is running the model.

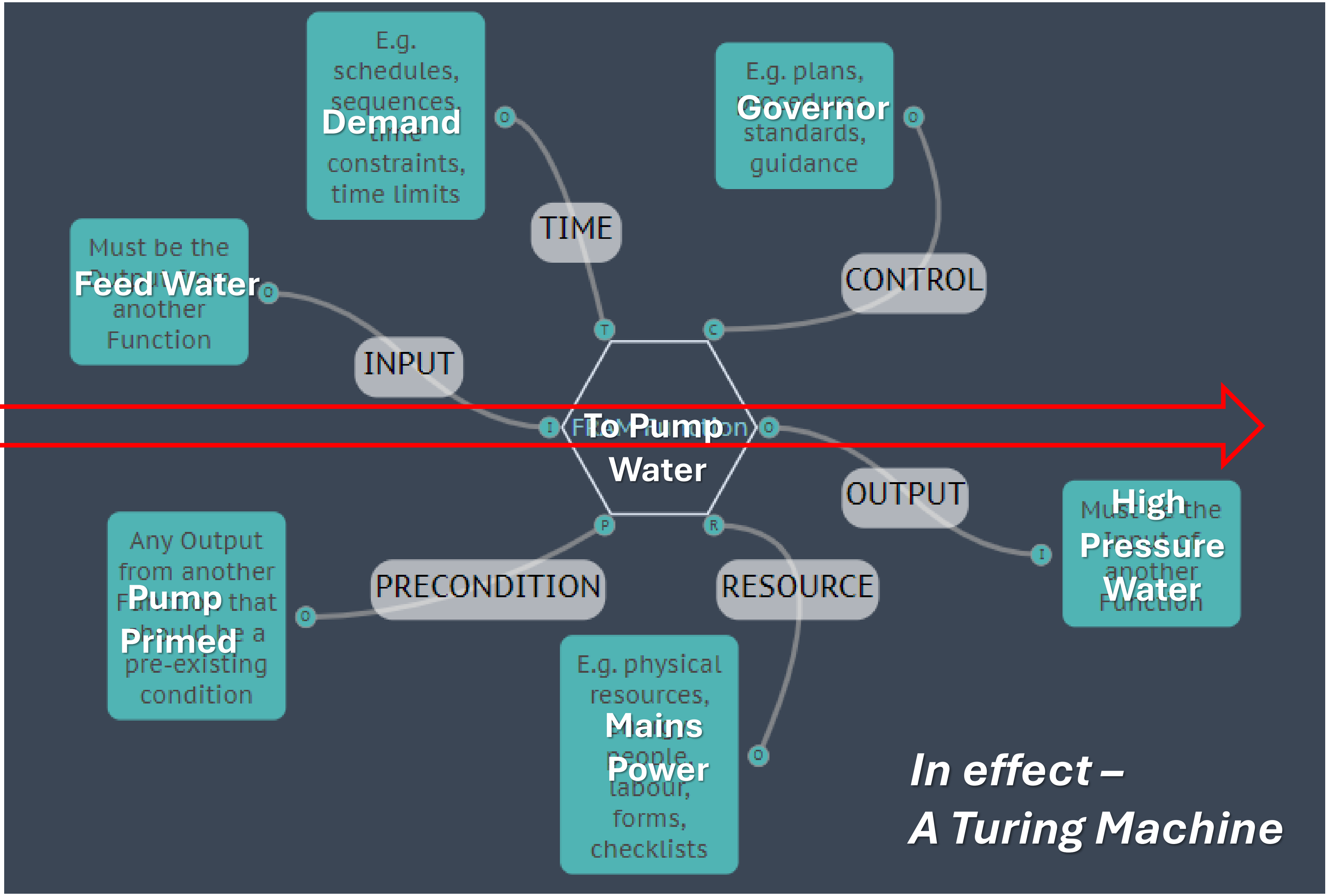
- **UK Government COVID Response**
- **Drug Dispensing**
- **Emergency response of Trauma centres**
- **Hospital Bed availability**
- **Gulf of Mexico Oil Spill**
- **Formula 1 Pit stop**
- **Dollar / Yen Currency exchange rate**
- **Digital Twins**
- **Drones**

IF:

Hollnagel's FRAM function can be made quantitative – THE BRIDGE

All Aspects present and correct
- Input,
- Precondition,
- Resources,
- Control and
- Timing

THEN:
- Pump water



This is a line of computer code, and we can execute it as a program (The FRAM Model Interpreter (FMI) software Tool)

If we know how that function operates, ($P_{h(kW)} = q\ \rho\ g\ h\ /\ (3.6\ 10^6)$), the program can calculate the OUTPUT as Metadata

From FRAM Function to Executable Function

FRAM gave us - functions - couplings – interaction - emergence

We can now generalise and simplify e.g., -

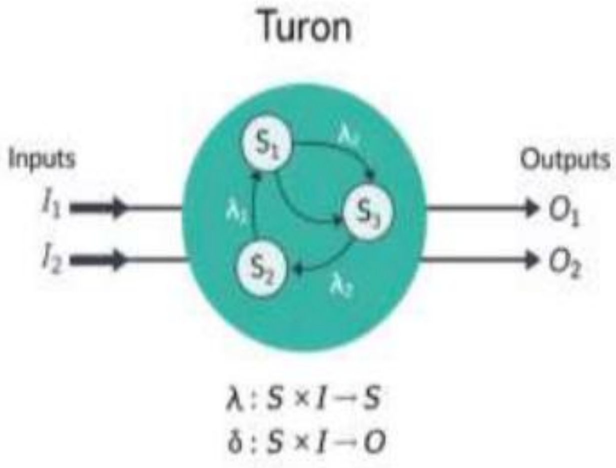
“Resonance” = propagation of performance variability

Resilience? Graceful extensibility???

Next step

- formalise the function (**a “Turon”)**

as an executable unit with state:



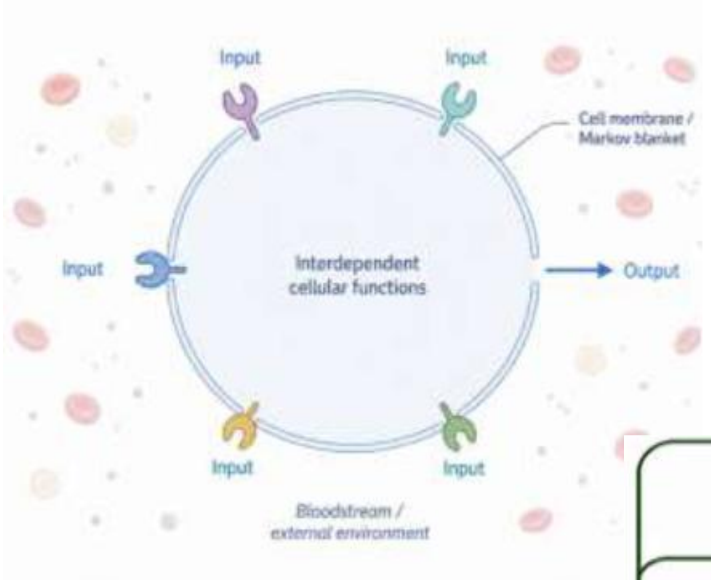
Formally, a Turon may be represented (Figure 1) as:

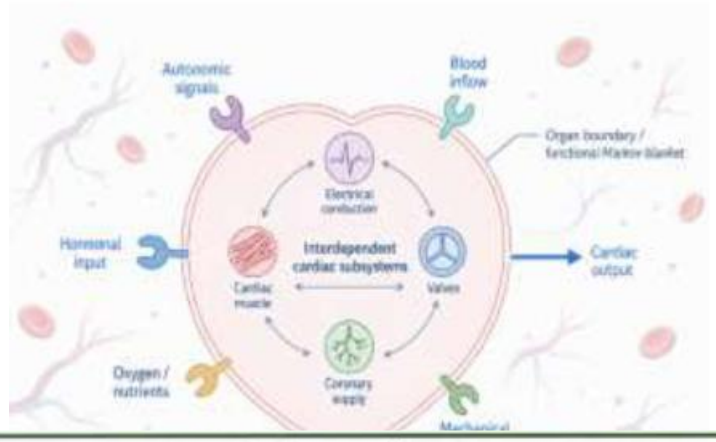
$$T = (\Sigma_{in}, \Sigma_{out}, S, \gamma, \delta, \lambda, \Theta)$$

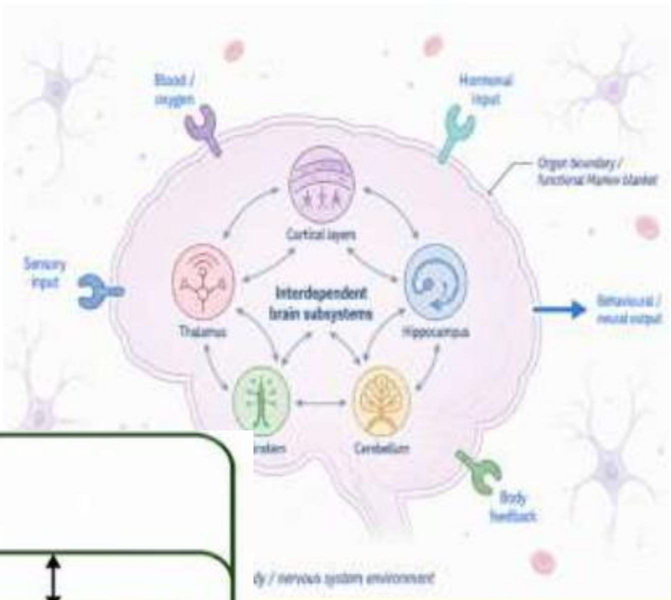
where Σ_{in} is the set of incoming typed signal packets, Σ_{out} is the set of emitted signal packets, S is the internal state space, γ is the guard or activation function, δ is the state-transition function, λ is the output function, and Θ is the parameter set, including thresholds, gains, delays, priorities, decay rates and merge rules.

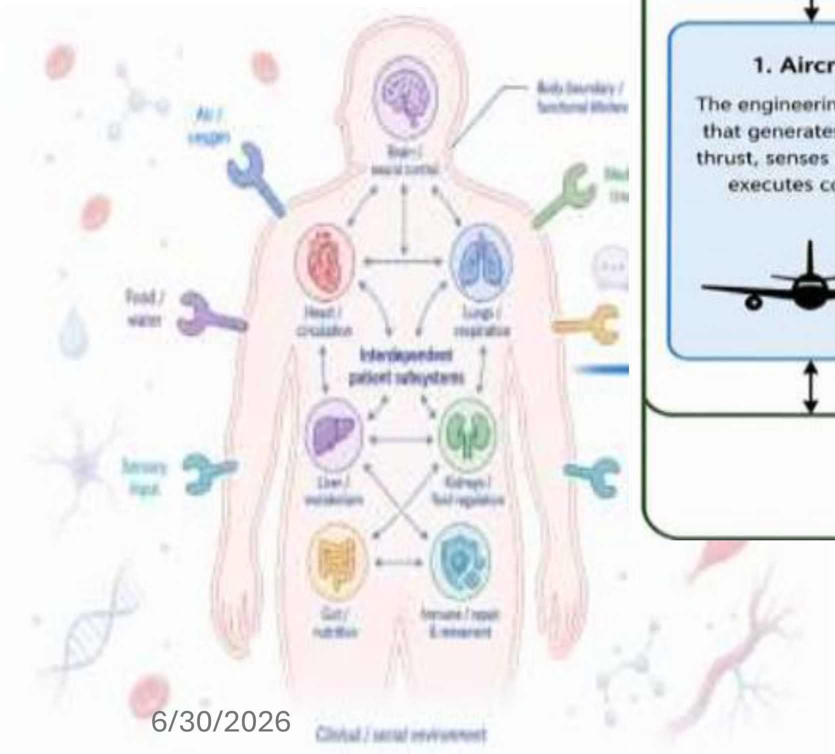
System view

- a system is an interacting set of Turons
- the system boundary is defined by a **Markov blanket**

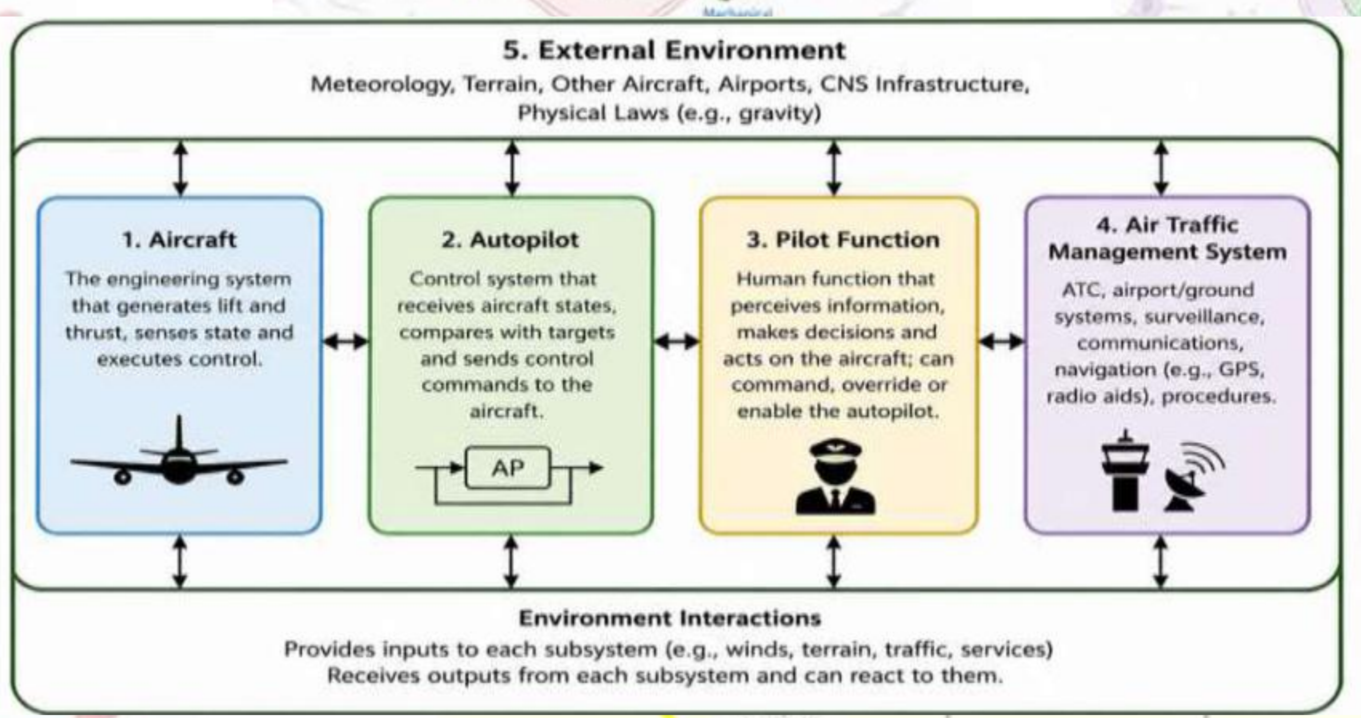

Input
Input
Cell membrane / Markov blanket
Input
Interdependent cellular functions
Output
Input
Input
Bloodstream / external environment


Autonomic signals
Blood inflow
Hormonal input
Interdependent cardiac subsystems
Cardiac output
Oxygen / nutrients

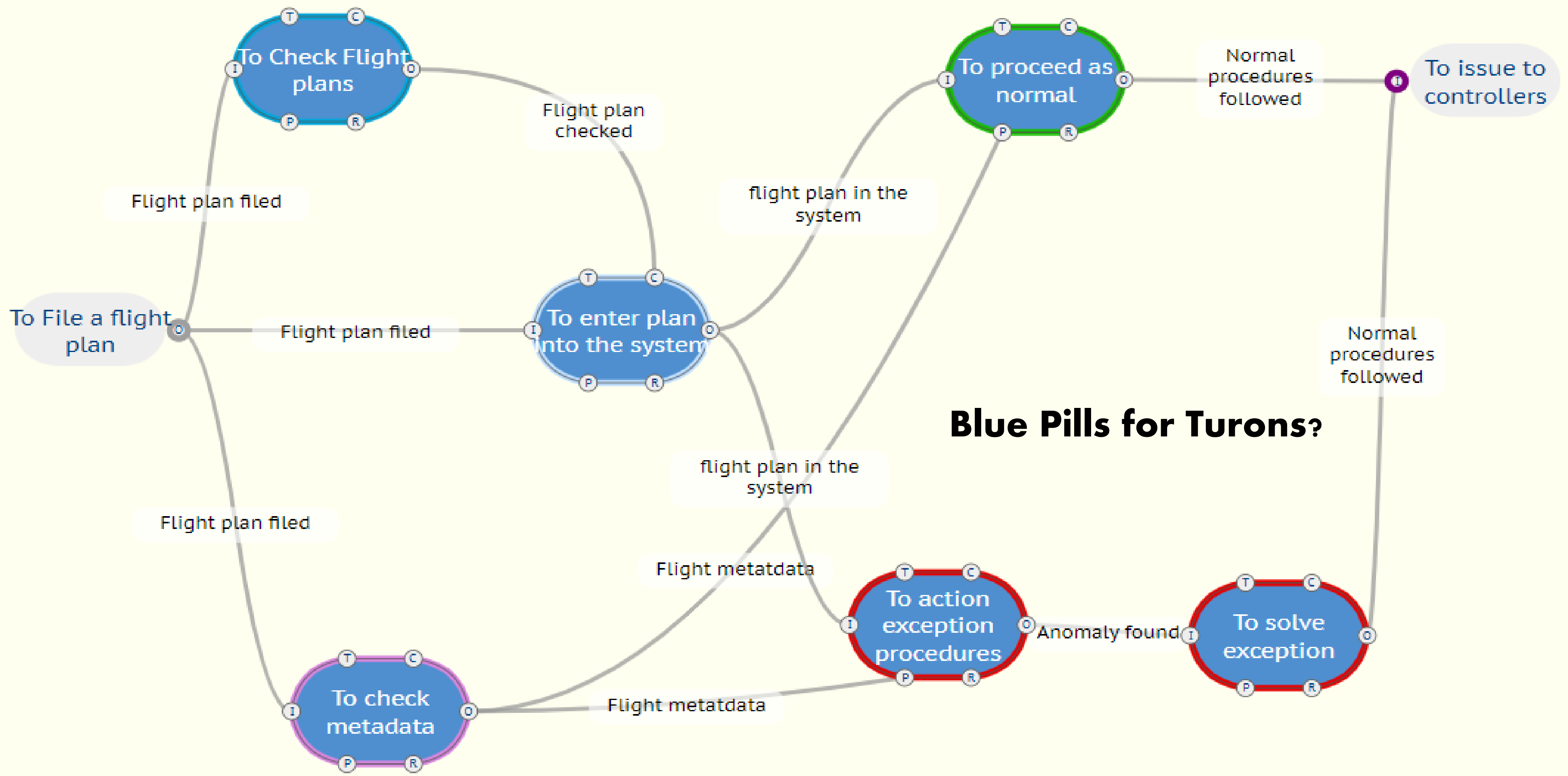

Interdependent brain subsystems


Interdependent patient subsystems


To drink tea


5. External Environment
Meteorology, Terrain, Other Aircraft, Airports, CNS Infrastructure, Physical Laws (e.g., gravity)
1. Aircraft
The engineering system that generates lift and thrust, senses state and executes control.
2. Autopilot
Control system that receives aircraft states, compares with targets and sends control commands to the aircraft.
AP
3. Pilot Function
Human function that perceives information, makes decisions and acts on the aircraft; can command, override or enable the autopilot.
4. Air Traffic Management System
ATC, airport/ground systems, surveillance, communications, navigation (e.g., GPS, radio aids), procedures.
Environment Interactions
Provides inputs to each subsystem (e.g., winds, terrain, traffic, services)
Receives outputs from each subsystem and can react to them.

A model builder for visualising system behaviours



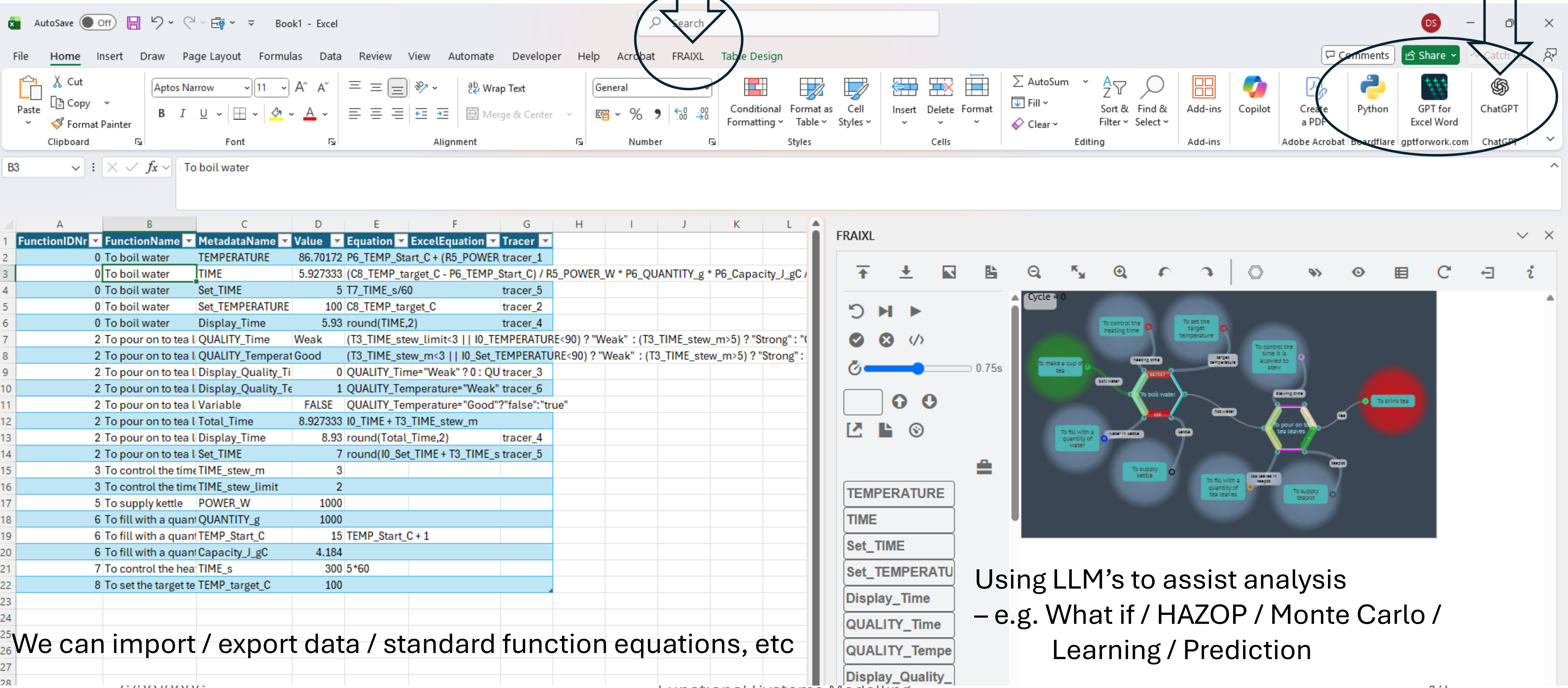
Blue Pills for Turons?

Open Source software

6/30/2026

Now available as an Excel add in –

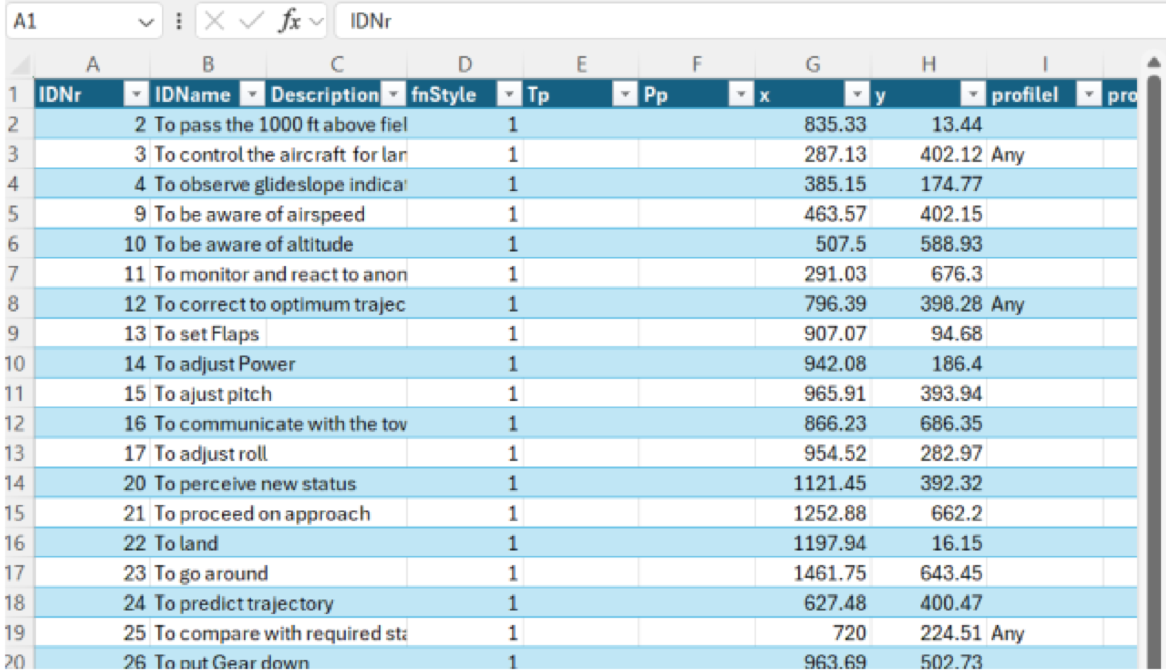
which allows us to build starter System models with ChatGPT



FunctionIDNr	FunctionName	MetadataName	Value	Equation	ExcelEquation	Tracer
0	To boil water	TEMPERATURE	86.70172	P6_TEMP_Start_C + (R5_POWER		tracer_1
0	To boil water	TIME	5.927333	(C8_TEMP_target_C - P6_TEMP_Start_C) / R5_POWER_W * P6_QUANTITY_g * P6_Capacity_J_gC /		
0	To boil water	Set_TIME	5	T7_TIME_s/60		tracer_5
0	To boil water	Set_TEMPERATURE	100	C8_TEMP_target_C		tracer_2
0	To boil water	Display_Time	5.93	round(TIME,2)		tracer_4
2	To pour on to tea l	QUALITY_Time	Weak	(T3_TIME_stew_limit<3 || I0_TEMPERATURE<90) ? "Weak" : (T3_TIME_stew_m>5) ? "Strong" : "		
2	To pour on to tea l	QUALITY_Temperat	Good	(T3_TIME_stew_m<3 || I0_Set_TEMPERATURE<90) ? "Weak" : (T3_TIME_stew_m>5) ? "Strong" :		
2	To pour on to tea l	Display_Quality_Ti	0	QUALITY_Time="Weak" ? 0 : QU		tracer_3
2	To pour on to tea l	Display_Quality_Te	1	QUALITY_Temperature="Weak"		tracer_6
2	To pour on to tea l	Variable	FALSE	QUALITY_Temperature="Good"?"false":"true"		
2	To pour on to tea l	Total_Time	8.927333	I0_TIME + T3_TIME_stew_m		
2	To pour on to tea l	Display_Time	8.93	round(Total_Time,2)		tracer_4
2	To pour on to tea l	Set_TIME	7	round(I0_Set_TIME + T3_TIME_s		tracer_5
3	To control the time	TIME_stew_m	3			
3	To control the time	TIME_stew_limit	2			
5	To supply kettle	POWER_W	1000			
6	To fill with a quant	QUANTITY_g	1000			
6	To fill with a quant	TEMP_Start_C	15	TEMP_Start_C + 1		
6	To fill with a quant	Capacity_J_gC	4.184			
7	To control the hea	TIME_s	300	5*60		
8	To set the target te	TEMP_target_C	100			

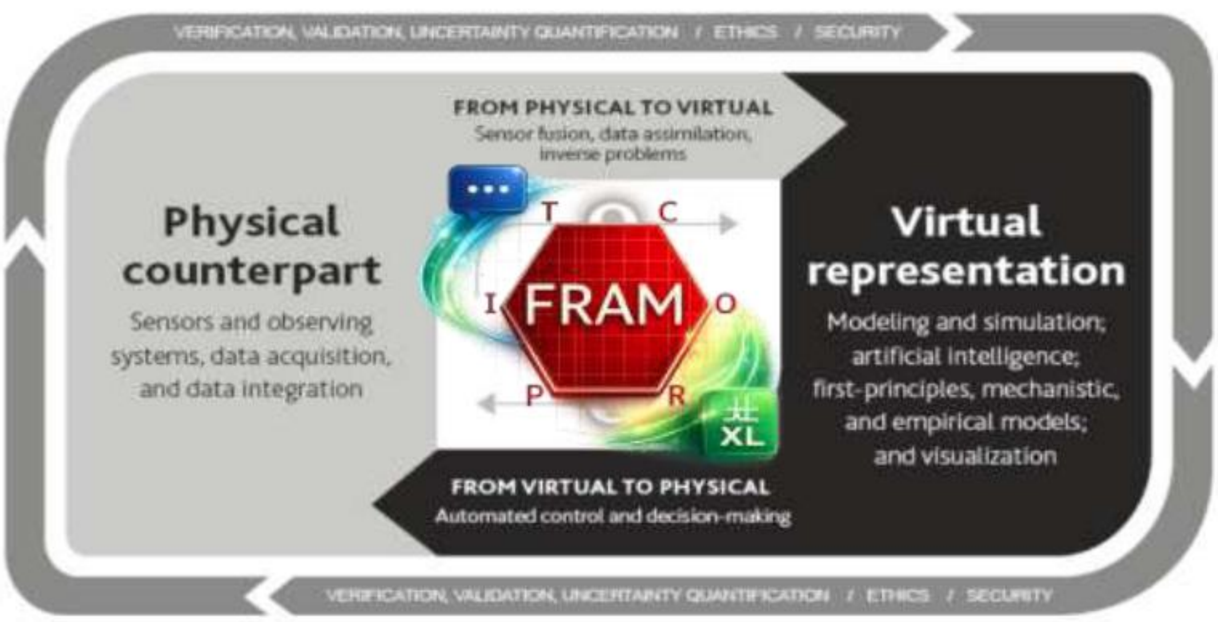
We can import / export data / standard function equations, etc

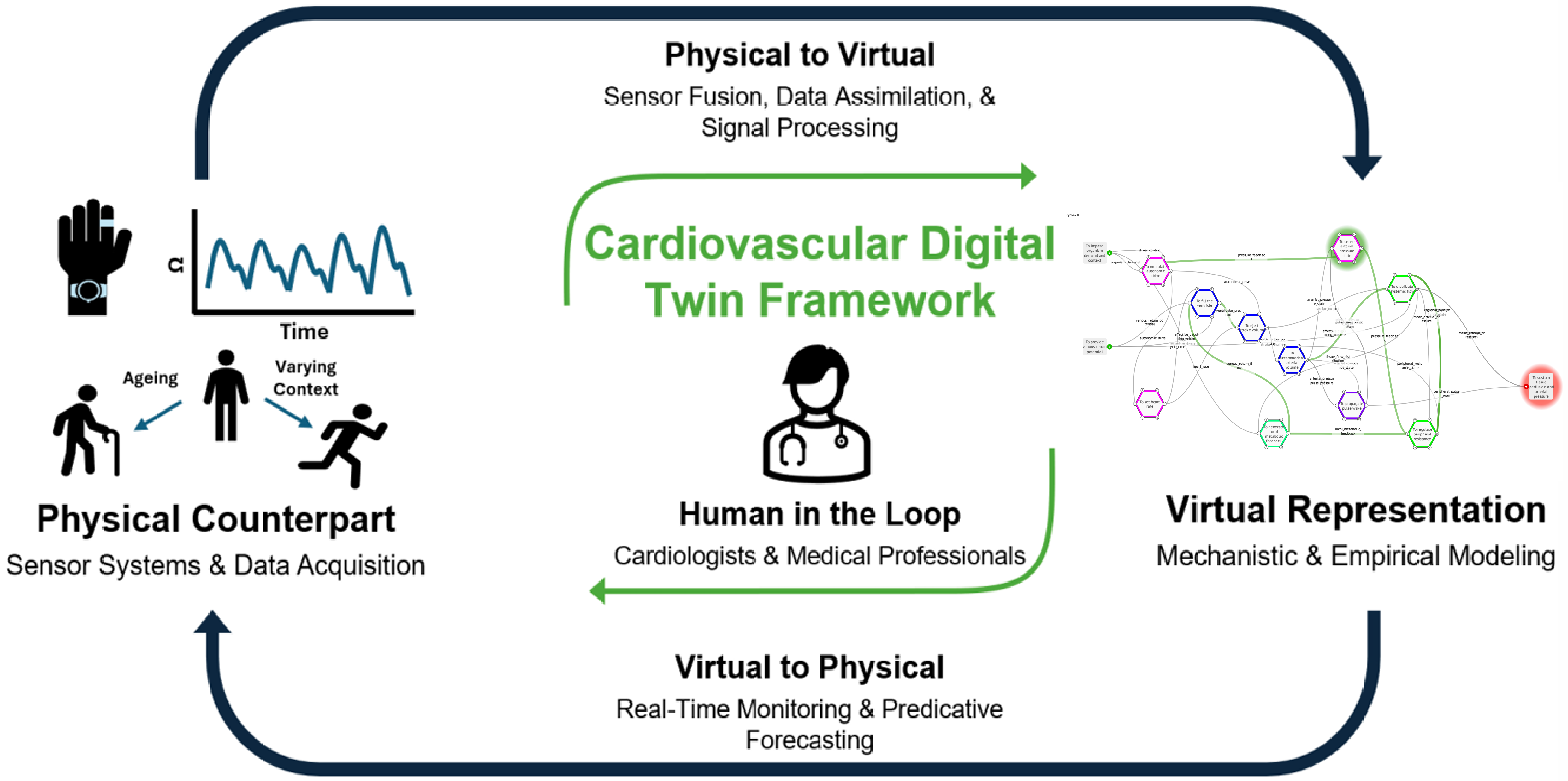
Using LLM's to assist analysis – e.g. What if / HAZOP / Monte Carlo / Learning / Prediction



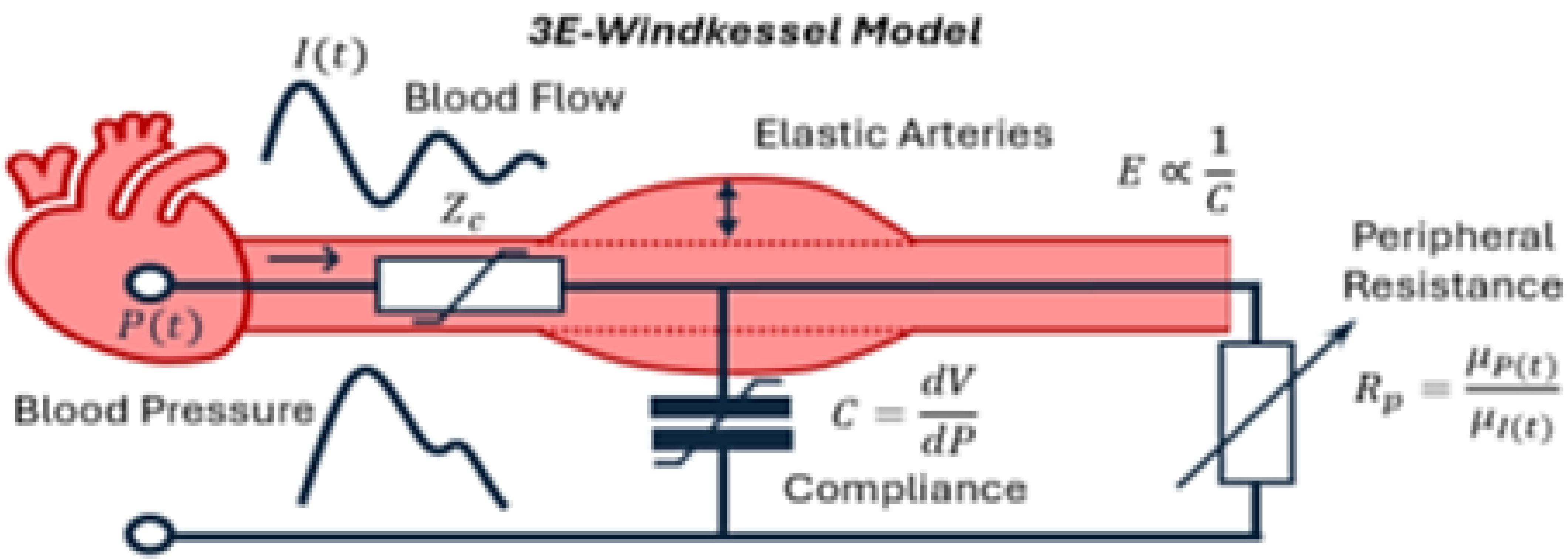
A1 | IDNr

IDNr	IDName	Description	fnStyle	Tp	Pp	x	y	profileI	pro
2	To pass the 1000 ft above fiel		1			835.33	13.44		
3	To control the aircraft for lar		1			287.13	402.12	Any	
4	To observe glideslope indica		1			385.15	174.77		
9	To be aware of airspeed		1			463.57	402.15		
10	To be aware of altitude		1			507.5	588.93		
11	To monitor and react to anon		1			291.03	676.3		
12	To correct to optimum trajec		1			796.39	398.28	Any	
13	To set Flaps		1			907.07	94.68		
14	To adjust Power		1			942.08	186.4		
15	To ajust pitch		1			965.91	393.94		
16	To communicate with the tov		1			866.23	686.35		
17	To adjust roll		1			954.52	282.97		
20	To perceive new status		1			1121.45	392.32		
21	To proceed on approach		1			1252.88	662.2		
22	To land		1			1197.94	16.15		
23	To go around		1			1461.75	643.45		
24	To predict trajectory		1			627.48	400.47		
25	To compare with required sta		1			720	224.51	Any	
26	To put Gear down		1			963.69	502.73		





Physical to Virtual
Sensor Fusion, Data Assimilation, & Signal Processing
Cardiovascular Digital Twin Framework
Time
Ageing
Varying Context
Physical Counterpart
Sensor Systems & Data Acquisition
Human in the Loop
Cardiologists & Medical Professionals
Virtual Representation
Mechanistic & Empirical Modeling
Virtual to Physical
Real-Time Monitoring & Predicative Forecasting



$$\frac{P(t)}{R_p} + C\frac{dP(t)}{dt} - I(t) = 0$$

where $P(t)$ is blood pressure, $I(t)$ is blood flow, C is compliance, and R_p is peripheral resistance. So the model links pressure and flow through storage in the arterial walls and outflow through the peripheral circulation. 6dc236e0-a858-4579-b751-870f1b7...

Does that mean we don't need to do all this standards stuff?

No – To build a car we do need to know what the parts / components are

And a Haynes manual of how you put it all together

- KNOWLEDGE

But we need to drive it (or Simulate it),
to be confident as to how it will perform on the road.

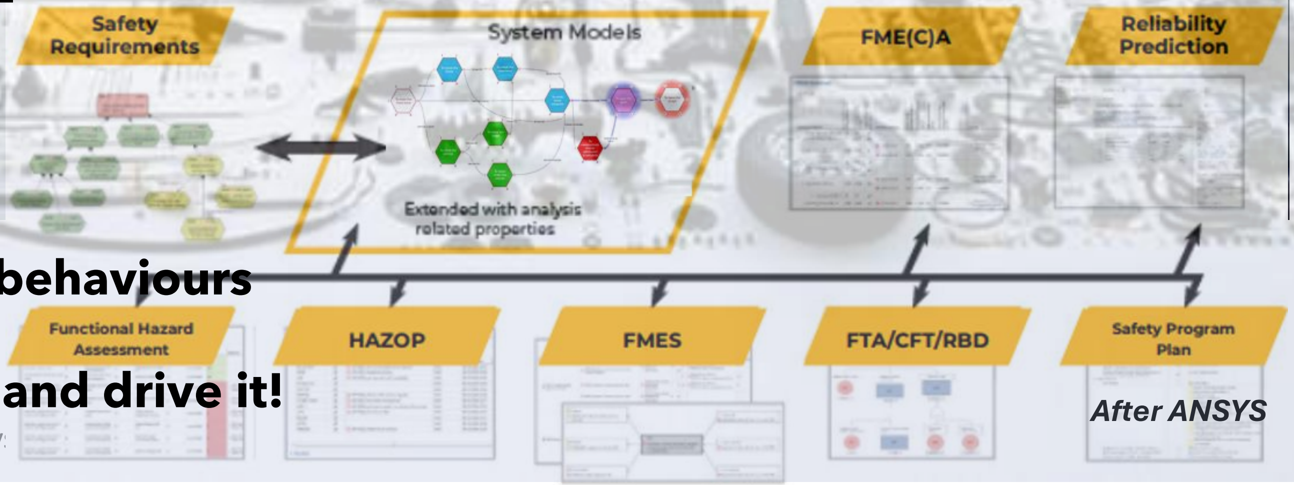
- UNDERSTANDING - (Ackoff)

So, a detailed Database on design intent, architecture and documentation is an important precursor –

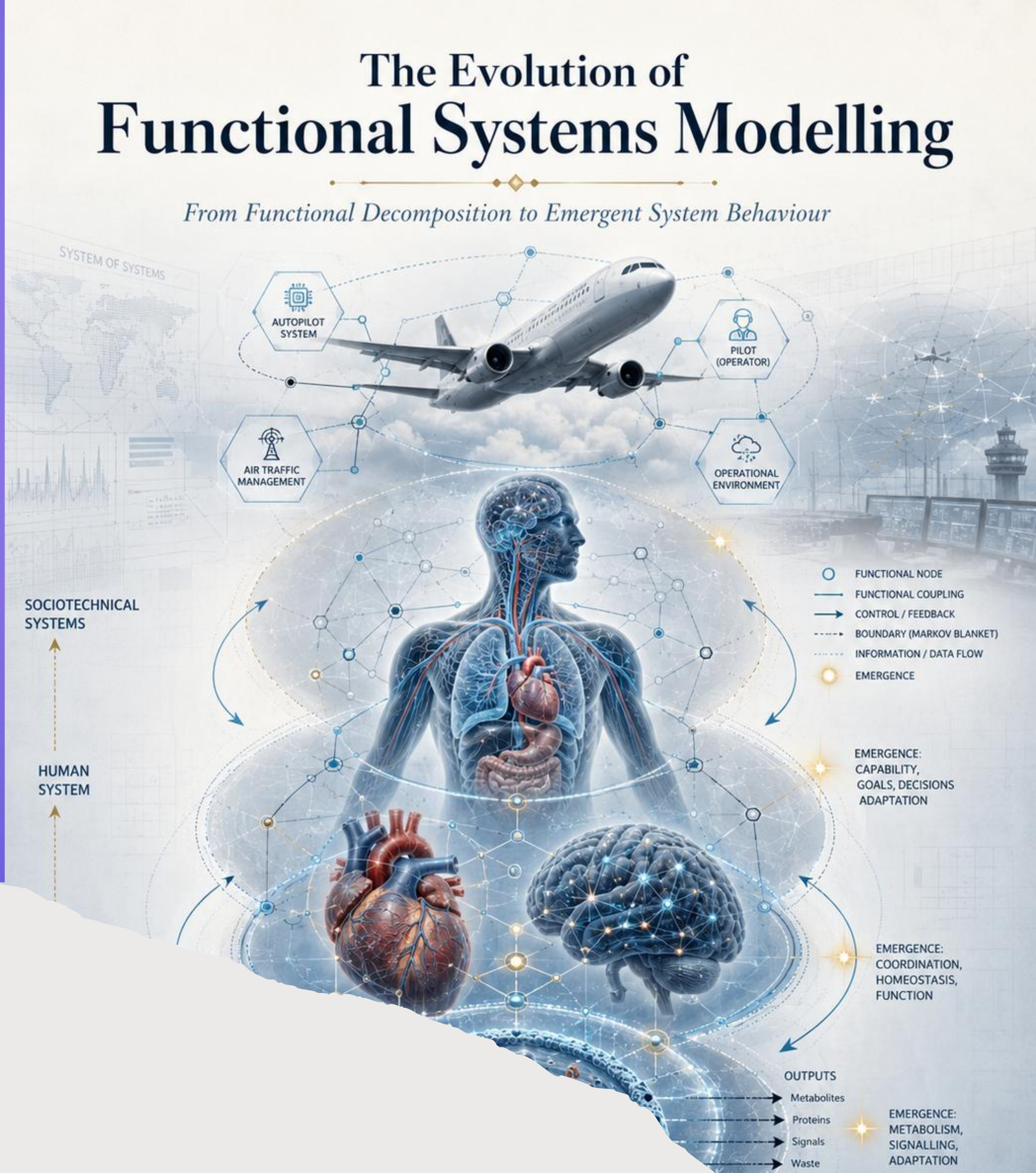
- COMPONENT Level

But we still need to build the SYSTEM
to pick up unexpected, emergent, non-linear, behaviours

So, we can now use FSM to put it all together and drive it!



After ANSYS



Thankyou - Questions?

Previous comments /questions as starters?

- Why do you want these models -purpose. (safety? Operability? Understanding?) These are not the same purposes and may require different approaches.

- What about variable behaviour and (random) deviations?

- What about humans. (I just saw one of the aircraft crash analysis issues again where the pilots just forgot (!) to extend the flaps before take-off.

- Is this really better than HAZOP combined with probabilistic fault and event trees?

- Do we really expect this modelling to produce more than we put in?

--- (why can red bull suddenly not produce a winning car anymore)

- Maybe we should not aim at a universal model, but use the set of models as a toolkit to get to grips with system behaviour.

Constraints and Assumptions

Table 1 - Classification of Contemporary Complex-Systems Modelling Methods

Method family	Semantic substrate	How structure enters	What is represented	Structural status
UML / SysML	None, intrinsic (descriptive)	Explicit, predefined topology	Components, states, interfaces, flows	Strongly structurally specified
System Dynamics (SD)	Continuous stock–flow equations	Explicit, predefined topology	Abstract accumulations & feedback	Structurally specified
Dynamic Bayesian Networks (DBN)	Probabilistic temporal inference	Explicit, predefined DAG	Random variables & dependencies	Structurally specified
Petri Nets	Token flow & firing rules	Explicit, predefined bipartite graph	Places, transitions, tokens	Structurally specified
DEVS	Discrete-event semantics	Explicit, predefined modular couplings	Components & event ports	Structurally specified
Discrete-Event Simulation (DES)	Event scheduling	Implicit in process logic	Processes, queues, resources	Process-specified
Agent-Based Modelling (ABM)	Agent rule execution	Emergent or optional	Agents & local interactions	Process-specified
Hybrid dynamical systems	State-space flows + mode switches	Implicit in equations/guards	State variables	Dynamics-specified
SDE / data-driven dynamics	Stochastic state evolution	Implicit or inferred	State variables	Dynamics-specified
FRAM (model)	Functional availability & variability	None predefined (compatibility only)	Functions & aspects	Non-structural metamodel
FRAM (instantiation)	Conditional activation	Transiently realised	Event traces	Ephemeral structure

Postscript from a colleague

(David Porter)

- I find it interesting that we've gone from functions (60s/70s/80s) to objects (80s/90s/00s) and back to functions (10s/20s) courtesy of FRAM.

- I think this is good news. I'm a big fan of object-orientation, it's a great way to write a computer program. Functions, however, are a bit more anthropomorphic and feel a closer fit to the natural/human/real world, especially when you're modelling at a more abstract level.

I'm trying to understand why we have had SO MANY different modelling methods and each one has never quite cut it. Perhaps it's an impossible dream!

- Perhaps we keep hoping to come up with a universal modelling language that can model absolutely anything and will always fail. It seems that most modelling methods lend themselves to specific niche domains. In the end, just like statisticians, we tend to use the methods with which we are most familiar.