

Formality and Non-formality

Michael Jackson
jacksonma@acm.org

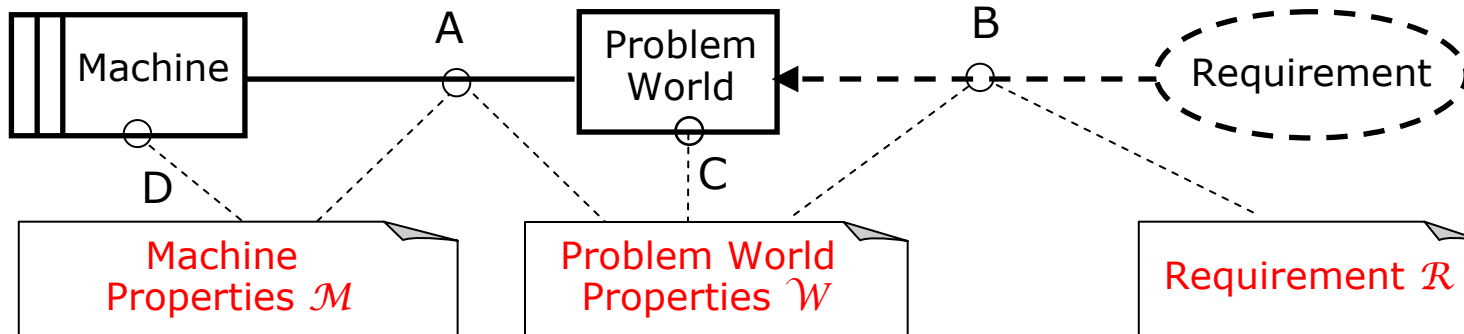
BCS RESG Formal-Lite Event
University of York
September 19, 2007

Software and software engineering

- Software may be about pure mathematics
 - Factorise a large number
 - Find convex hull of set of points in 3D
 - Prove the four-colour theorem
 - ...
- Software engineering is about the world
 - Open and close the Rotterdam barrier
 - Support the administration of a library
 - Control a radiation therapy machine
 - Control application of a car's brakes
 - ...

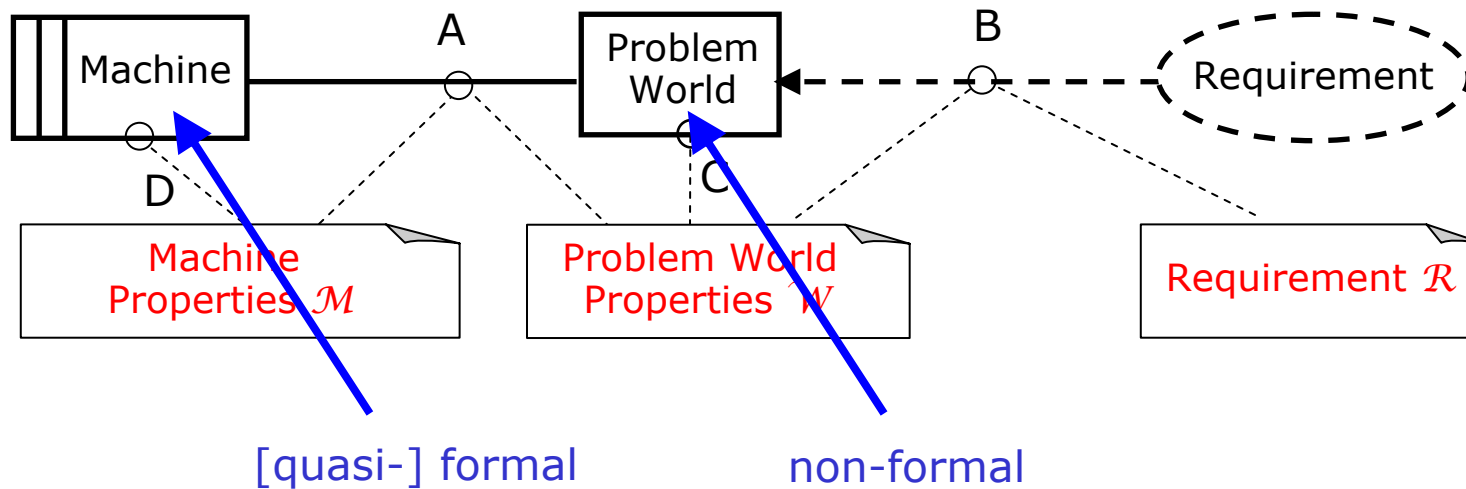


Machine, problem world, requirement



- **Phenomena**
 - A: phenomena shared between Machine and World
 - MotorOn, SensorOn[f], DoorMotorOn, ButtonPress[b], ...
 - B: phenomena mentioned in the Requirement
 - CarArrives[f], DoorOpens, PassengerRequest[f], ...
 - C: hidden phenomena of the problem world
 - UserWaiting[u,f], CableBroken, ...
 - D: hidden phenomena of the machine
 - malloc(size), ...
- The Machine and the Problem World interact ... $\mathcal{M}, \mathcal{W} \neq \mathcal{R}$

Formality and non-formality



“Thus, the essence of useful software consists in its being a constructively interpretable description of properties of two ... structures: [formal] hardware and [non-formal] application domain, respectively.”

W M Turski; *And No Philosopher's Stone Either*
(response to Brooks *No Silver Bullet*, IFIPS 1986)

Formality: always necessary, never sufficient

- Is formality necessary? Of course it is!
 - The hardware/software is [quasi-]formal
 - How else can we reason accurately?
- Is formality sufficient? Of course it isn't!
 - Formal reasoning about a non-formal world yields **unreliable conclusions**
 - Formal reasoning alone cannot address **failure of a formalisation**
- Interplay of formal and non-formal
 - A crucial topic in software engineering
 - A major challenge

Character of non-formal worlds

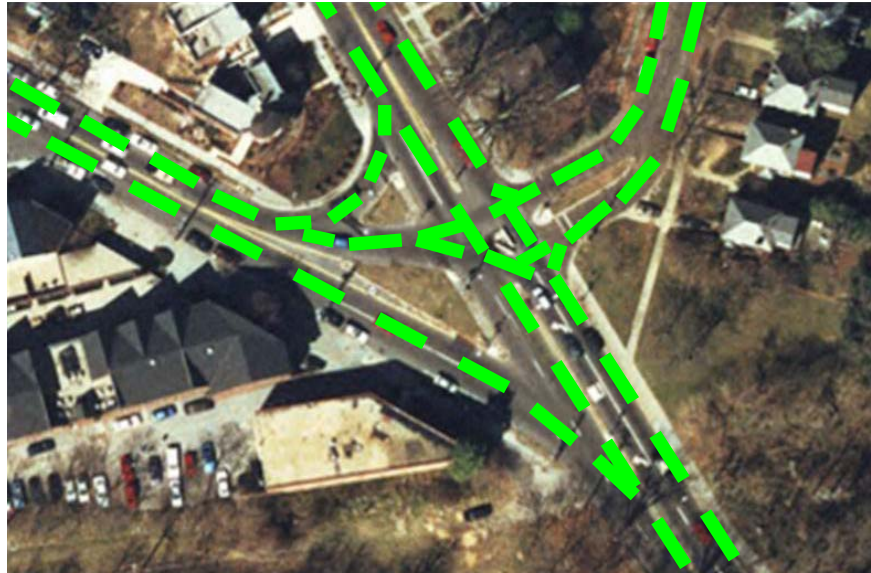
- Continuous phenomena
 - eg: What counts as 'LiftRising'?
- Fewer regularities, more individual cases
 - eg: Flow in complex traffic intersection
- No finite alphabet of phenomena is enough
 - eg: Comet 1 metal fatigue
- Universe changing over time
 - eg: operator skills in Therac-25
- No exact, reliable axioms or theorems
 - eg: in Car Park Control: $\#carsOut(t) \leq \#carsIn(t)$?
- Control often matters
 - eg: Who pressed the accelerator pedal?
- ...

Fewer regularities, more individual cases



- What concurrent flows are permitted?
- Where should light units be positioned?
- What control regime for the lights?

Fewer regularities, more individual cases



- What concurrent flows are permitted?
- Where should light units be positioned?
- What control regime for the lights?

No finite alphabet of phenomena is enough



de Havilland Comet 1

- Stresses on Comet 1 fuselage
 - Cabin pressurisation/depressurisation
 - Torsion from forces on wings, tail and fin
- Metal fatigue cracks caused by ...
 - Both stresses simultaneously (not tested)
 - Window corner sites for crack growth
- Formal design models excluded fatigue phenomena

Operator skills increasing with time

PATIENT NAME : TEST			A	1
TREATMENT MODE: FIX	BEAM TYPE: X ENERGY (KeV):		25	
	ACTUAL	PRESCRIBED		
UNIT RATE/MINUTE	0	200		
MONITOR UNITS	50 50	200		
TIME (MIN)	0.27	1.00		
GANTRY ROTATION (DEG)	0.0	0	VERIFIED	
COLLIMATOR ROTATION (DEG)	359.2	359	VERIFIED	
COLLIMATOR X (CM)	14.2	14.3	VERIFIED	
COLLIMATOR Y (CM)	27.2	27.3	VERIFIED	
WEDGE NUMBER	1	1	VERIFIED	
ACCESSORY NUMBER	0	0	VERIFIED	
DATE : 84-OCT-26	SYSTEM: BEAM READY	OP.MODE: TREAT	AUTO	
TIME : 12:55. 8	TREAT : TREAT PAUSE	X-RAY	173777	
OPR ID: T25VO2-RO3	REASON: OPERATOR	COMMAND:		

“If the prescription data was edited at a fast pace (as is natural for someone who has repeated the procedure a large number of times) the overdose occurred.

...

“The software error is just a nuisance on the Therac-20 because this machine has independent hardware protective circuits for monitoring the electron beam scanning.”

Leveson and Turner; An investigation of the Therac-25 accidents

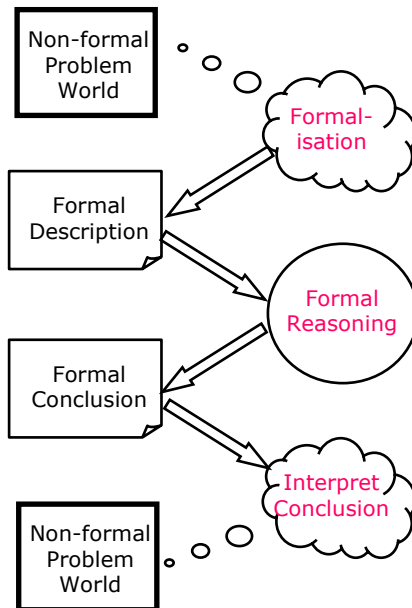
Control often matters

- Improved parking brake system *
- Button on fascia for brake-on
- Automatic brake-off when driver presses accelerator pedal
- Test driver returns to factory
 - Stops car in front of gates
 - Presses brake-on button
 - Leaves car, walks to gates
 - Starts to open factory gates
 - Brake goes off, car moves
- What had happened?
 - Was there a design error?



* Manfred Broy recounted this story

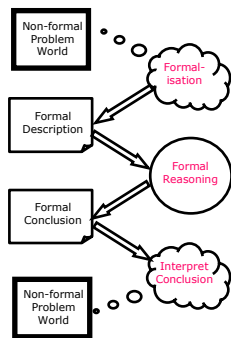
Reasoning about non-formal worlds



- To reason in the problem world we need:
 - **Formalisation**: necessarily imperfect but judged adequate to the particular problem world and requirement
 - **Formal reasoning** (eg logic): about our formalised descriptions of the world
 - **Explicit interpretation** of conclusions: including experimental truth checking

- How can this process fail?
 - Inadequate initial formalisation
 - Formally erroneous formal reasoning
 - Erroneous conclusion of correct reasoning
 - eg $\frac{e1 ; e2}{e1 \wedge e2}$ may not hold in the particular case

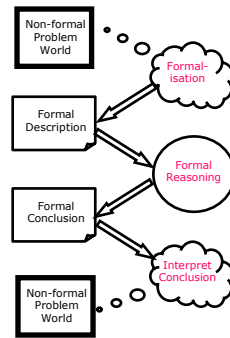
Correct reasoning, erroneous conclusions



$P \stackrel{\text{def}}{=} \text{"fuselage withstands pressure changes"}$

P holds

;



$T \stackrel{\text{def}}{=} \text{"fuselage withstands torsion stresses"}$

T holds



$P \wedge T$ **does not hold**

- Contributing parts' alphabets ignored metal fatigue
 - Metal fatigue invalidated the composed conclusion

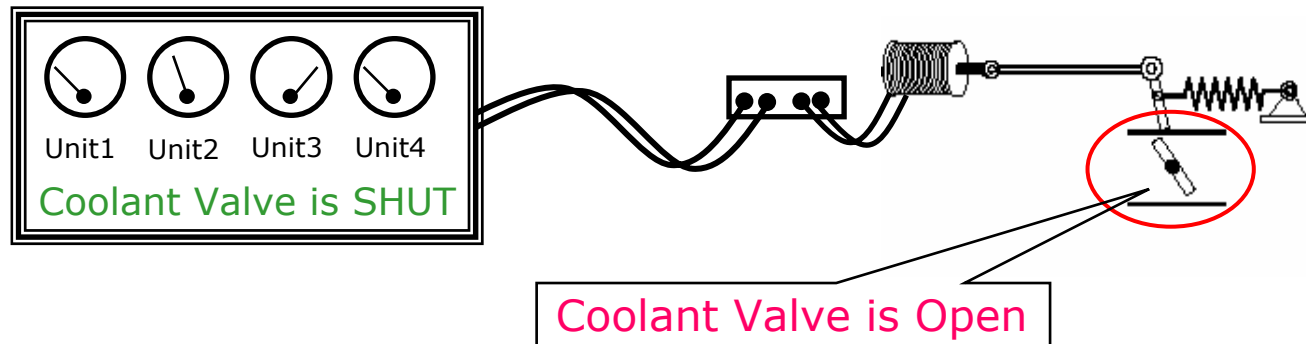
"... the behavioral output of the composite model is merely an infallible consequence of the information contained in the primary models, not of their real-world counterparts"

Bill Addis; Creativity and Innovation:
The Structural Engineer's Contribution to Design

How to avoid failures

- Demonstrate design correctness formally
 - Doesn't address formalisation failure
- Codify and avoid recognised failure types
 - Depends on specialised normal design evolved over a very long time
 - Needs careful focus on phenomena
- A theory of probable failures?
 - Needs careful focus on phenomena
 - Taxonomy of 'formalisation failures'
 - Search discipline for finding failures

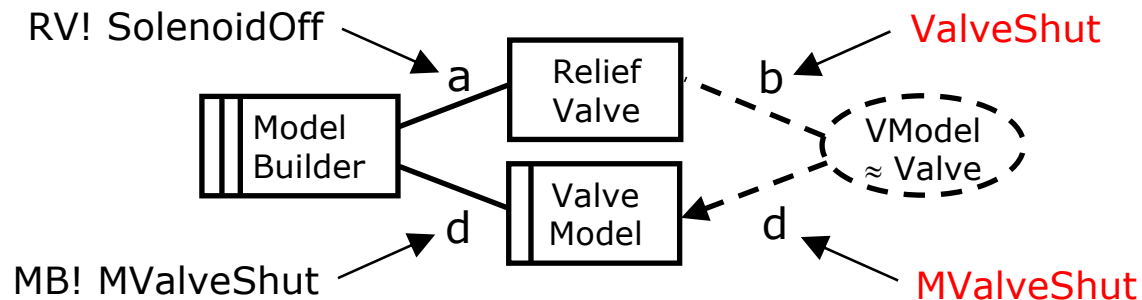
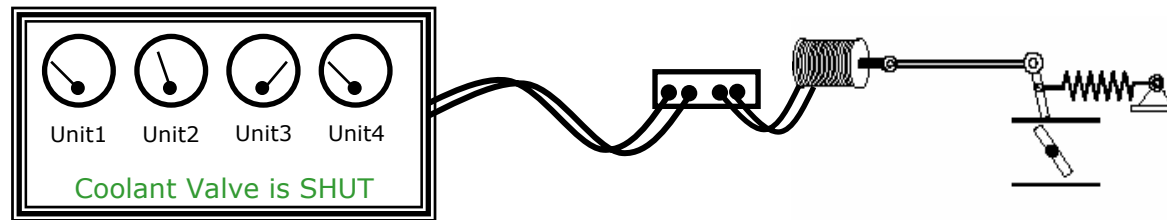
A recognisable failure type



- Three Mile Island incident, 1979
 - Partial reactor meltdown
 - Low reactor coolant level was a contributory factor
 - Relief valve was open, but panel indicated shut

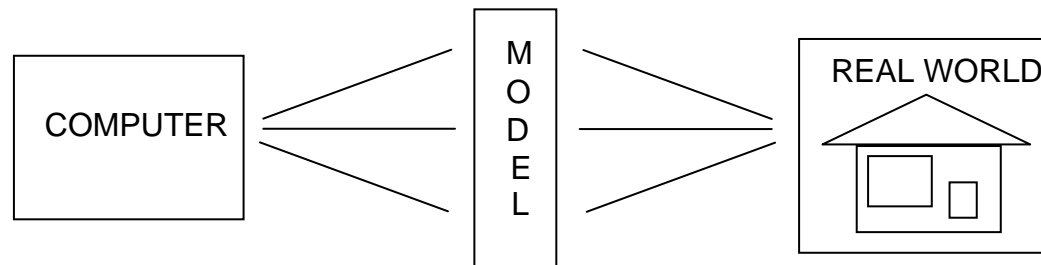
Eugene S Ferguson; Engineering and the Mind's Eye

A recognisable failure type



- Required: "MValveShut \equiv ValveShut"
- Assumed: "SolenoidOff \equiv ValveShut"
 - Is this an assertion in \mathcal{W} or a definition of 'ValveShut' or a definition of 'SolenoidOff'?
- A failure of model-world correspondence concern

A theory of formalism and non-formality?



- “The technical subject of model theory ... is a study of the relationship on the left. ... at this point in intellectual history, we have no theory of this right-hand side relationship.”

Brian Cantwell Smith: The Limits of Correctness;
Symposium on Unintentional Nuclear War, Budapest, Hungary, June/July 1985

- I have called Cantwell Smith’s **model** a **formalisation**
- We need a discipline to search out errors
 - Based on past errors in formal developments
 - Based on a theory of likely error loci

Searching for formalisation failures

- Reasoning about composition can introduce failures

$$\frac{e1 ; e2}{e1 \wedge e2} \text{ may not hold}$$

- Can we hope to identify potential failures?
 - $\alpha_1(e1)$ and $\alpha_2(e2)$ are the phenomena of $e1$ and $e2$
 - Form symmetric difference $\alpha = \alpha_1(e1) \Delta \alpha_2(e2)$
 - Consider:
 - Direct relationships among phenomena of α
 - Any further phenomena related to α
 - Form causality graphs for alphabet phenomena
 - Find patterns often associated with failure
 - ... ?

The challenge of the non-formal world

“We can never bridge the
formal/informal gap ...”

Jones, O’Hearn, Woodcock;
Verified Software: A Grand Challenge
IEEE Software April 2006

- Perhaps we can do better ...
 - ... but only if we try
- Ignoring this challenge is ignoring the central dependability issue in software-intensive systems

Thank you