

Cyber-Physical Programming

Challenge and Response

Michael Jackson
The Open University
jacksonma@acm.org

HaPoC Zuerich
27-29 October 2021

Cyber-Physical Programming

- 1 CP System Examples
- 2 Behaviour Structure
- 3 The Bipartite System
- 4 Programmability
- 5 Causality

- 6 Causal Failures
- 7 Axioms & Behaviours
- 8 Triplets
- 9 Model Structure
- 10 Envoi

1 CP System Examples

Radiotherapy



Flight Control



Automotive



Drug Pump



Floor Cleaner



Vending



Cyber = steer or govern
(specifically: real world **BEHAVIOUR**)

Physical = **CONCRETE** tangible world
(mechatronic, natural, built, people, etc etc)

Many dimensions of variety
(**COMPLEXITY**, phenomena, size, locality, **CRITICALITY**, ..)

2 Behaviour Structure

- * Many CP systems have complex behaviour

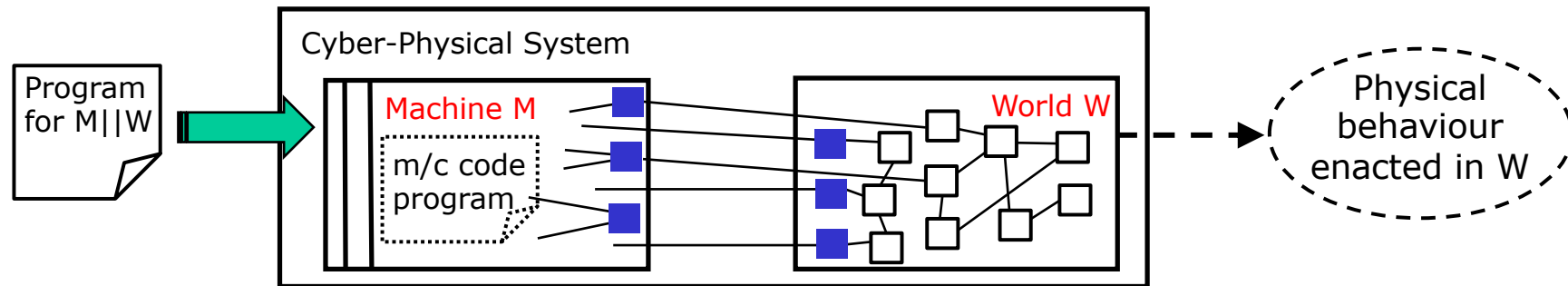


Self-park, limit speed, cruise control, anti-skid braking, air conditioning, clean charcoal filter, custom driver options, emission test, software update, fuel management, active suspension, ignition cycle, stop-start ...

- * **CONSTITUENT** behaviours: **CONCURRENT**, terminating or not
 - * **INTERACTING** both in software and in the physical world (parts of the physical world act as shared variables)
 - * Mutual incompatibilities (eg aircon and stop-start)
- * System behaviour involves much more than the car itself (driver, road, weather, visibility, other road users, ...)

3 The Bipartite System

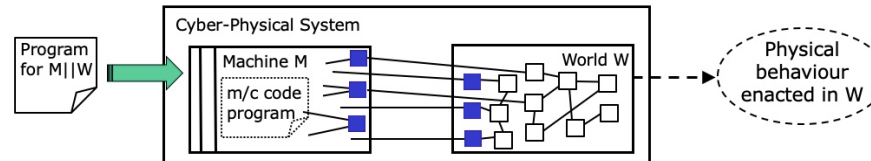
- * The system to program has 2 parts: **MACHINE M** + **WORLD W**



□, ■ = physical **DOMAIN** ■ = **INTERFACE** — = **SHARED PHENOMENA**

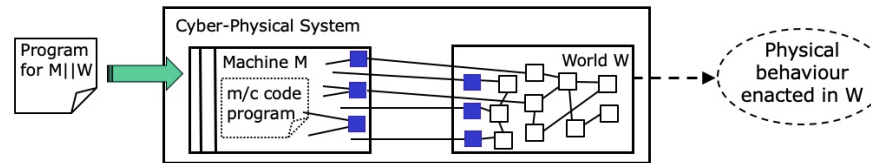
- * Domains are **ENTITIES** participating in **PHYSICAL BEHAVIOUR**
 - * M, W share phenomena at interface domains ■
(M: only ■ shown; W: □, ■ include human participants)
- * ■ → = Abstract program text → concrete **M/C CODE** FOR M
 - * An inescapable (formerly manual) refinement task

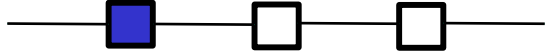
4 Programmability



- * M is **FULLY PROGRAMMABLE**
 - * Store and traverse a graph of m/c instructions
 - * Specified instruction effects ('add', 'jump', ...) are **AXIOMS**
 - * Instruction execution is **RELIABLE** (though imperfectly)
- * W is only **PARTLY PROGRAMMABLE**
 - * External programs only: W has no store and traverse
 - * "Instructions" are shared phenomena at interface ■
 - * The world is not formal: "axioms" are **CONTINGENT**
 - * W's interface and other domains are **UNRELIABLE**

5 Causality



- * How can M govern W beyond the interface domains?
 - * W "axioms" are **CAUSAL LINKS** in/between domains
 - * Causes and effects are events, states, ...
 - * Links can form chains 
- * Each causal link has an **EFFECTUATING DOMAIN** D in W_i ..
.. specifies conditions (eg current state of D)
- * Cause --> effect may be M --> W or W --> M
 - * 'Activator' and 'Sensor' are relative terms
- * Causality **SEMANTICS** may be intricate (**INHIBITION** etc)
 - * Causality is the logic of **CONTRIVANCES** [Polanyi]

6 Causal Failures

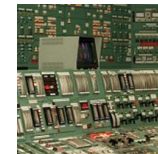
- * Some historic failures of causality modelling



Warsaw A320

- * Reverse thrust only if plane is on ground
- * Flooded runway: no wheel rotation caused
- * 'Rolling' landing: only one leg compressed

- * Causal link from relief valve to indicator ..
.. was not imputed to any identified domain
- * Relief valve stuck open but indicated closed



3 Mile Island



BMEWS radar

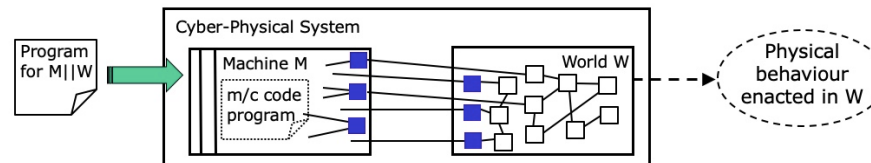
- * 1960:USSR missile strike launch indicated
- * Radar link: cause should be strike launch ..
.. but was unexpected position of rising moon

7 Axioms & Behaviours

- * Why **AXIOMS**? Judiciously chosen **UNQUESTIONED ASSUMPTIONS**
 - * An allusion to Euclid's axioms ..
 - .. defining the basis for constructions
- * Why are axioms **CAUSAL LINKS**?
 - * Because a CPS is a **CONTRIVANCE**
 - * Is 'the **LOGIC** of **CONTRIVING**' [Polanyi] causality?
- * Surely the laws of physics are the necessary axioms?
 - * True? Of course! Useful? sometimes! Sufficient? No!
 - * Scales; shapes; discrete properties; juxtapositions

8 Triplets

- * W axioms support development of system **BEHAVIOUR**
- * Model (axioms) must be **GLOBAL** wrt behaviour activations



- * **TRIPLET**: microcosmic CP system, one constituent behaviour
 - * Triplet $i = \{M_i \text{ program}, W_i \text{ model}, B_i \text{ behaviour } M_i || W_i\}$
 - * B_i combines contributions from both M_i and W_i
 - * W_i model: axioms required to support B_i activation
- * **DEVELOP** a triplet and **COMBINE** with others
 - * Triplet activations are linked by their program texts
 - * Combining is a separate (possibly invasive) task

9 Model Structure

- * **MODELLING-IN-THE-LARGE**: structured wrt **BEHAVIOURS**
 - * W_i model must hold during enactment of behaviour B_i ..
.. so model structure is behaviour enactment structure
- * **MODELLING-IN-THE-SMALL**: **A, B, C** are distinct aspects
 - * **AXIOMS** for W_i are causal links in and between W_i domains
 - * **BEHAVIOUR** $B_i = M_i || W_i$ (eg state machine, trace set, ...)
 - * **CONSEQUENCES** of B_i (satisfying relevant requirements)
- * Modelling as a discipline
 - * Rigorous **DENOTATIONS** in the physical world
 - * Corpus of identified **MODEL FAILURE CONCERNS**

10 Envoi

- * Reliable programming of an unreliable world
 - * CO-DESIGN Mi program and Wi model
 - * Both content and structure
- * A COMPUTING SCIENCE perspective
 - * FORMAL SPECIFICATION for M is IMPOSSIBLE ..
 - .. because W does not support reliable abstraction
- * Are CPS development CHALLENGES relevant for CS?
- * Where is CS most relevant to CPS development?

Thank you