

time
Mad ants
↑

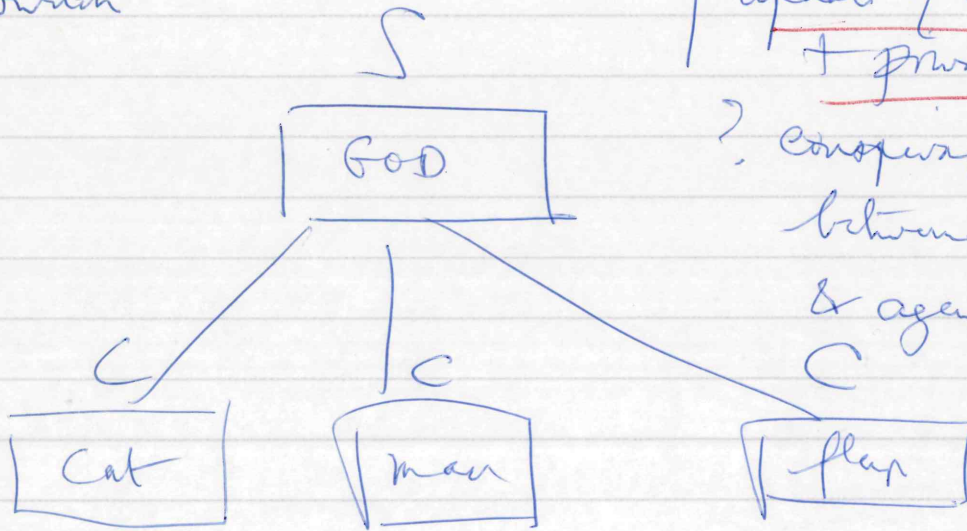
A. Concurrent Systems Simulation

- Examples:
- juggling
 - modelling of financial systems ← sources FT
 - traffic light system (cf. ?)
 - pedestrian
 - linking traffic lights
 - fire engine with banglades

railway / jugs. cat flap.

Cox study fly

switch



How about simultaneous updates of environmental + private?
 ? conspiracy between god & agent

Valery Adzhiev

LSD - engine

OOP C++ code

+ Richinsky

table tennis

Use Venn diagram to discriminate between diff kind of observables — LSD artifact.

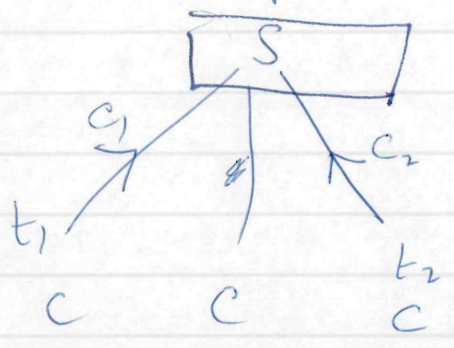
B. Concurrent Room Design

Guy design justifications like explanations cultural.

? auto propagate with concurrency

auto propagate with self-delivery

table to fill into corner.



What is current state of system?

constraints

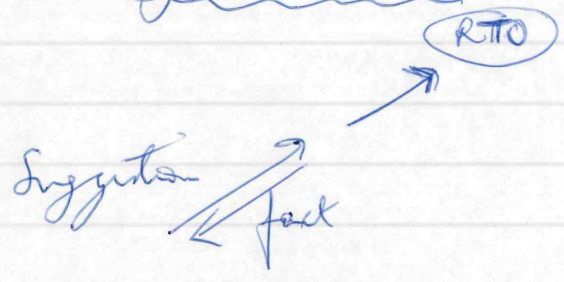
What kind of runtime error imp. is possible?

feasibility question

decomposition by object function

kind of design (specialism)

~~System~~ System state + reason (store system state?)



dependencies
reason = fn of state

absolute constraint on two deps of door.
→ update door ⇒ update door type

constraint if has all dts of RTO

voting system — not a specialism view.

parallel development of models

out of phase
with self-thing

out of phase
with movement
... to fill out more

what is meant by
of system?
Environment



What kind of review can you do in parallel?
feasibility studies

be comparable to light
function

kind of change / specification

state system state + sensor
(state system state?)

dependencies
... = ... of state



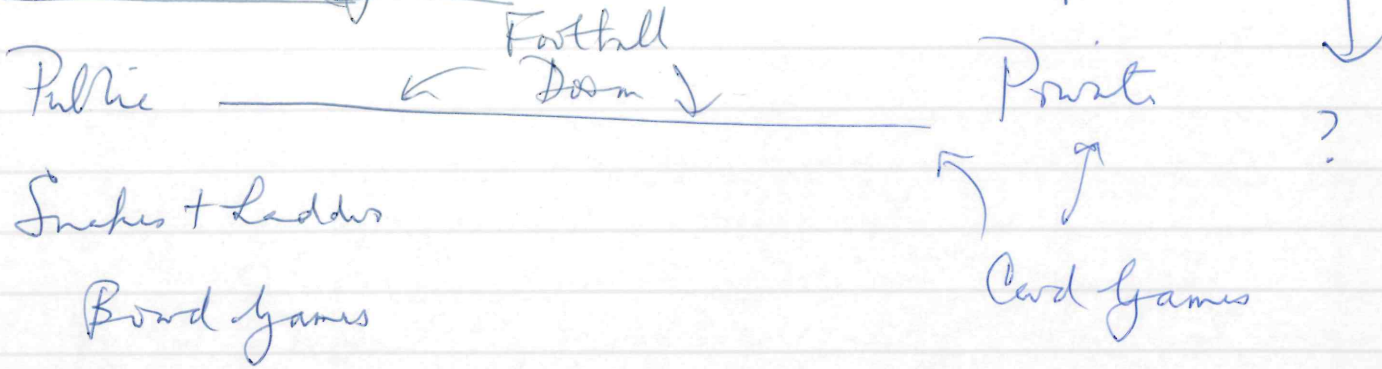
abstract conditions on the basis of data
→ update base ⇒ update base type

contributing of all bits of RTO

not a sequential view
vector system

not entirely private
impossible?

C. Multi-Player Games



Conventions arbitrate whether public or not

Football < private / public orientation of player determines memory

Can shift categories

Strategy could be private initially
as you recognize private strategies migrate

As more strategy, more chance

cf. NIM:

rules spread out among players. (emigrant?)

Conway's strategy

Public
|
rules
assumptions

Private
|
board config
strategy

layout board1
kit board1

board2
board2

slow-propagate

? private

D. Definitive Programming \leftrightarrow 9D mod

programming as modelling? \leftrightarrow 9D

Conventionally

Vars

Procs.

Flow of control

variables

info re agents modelling

+ other not independent

- DP
- L&D.
1. What are agents in system?
 2. Which are observables of each agent?
 3. How does behaviour of each agent depend on its observables?

! 4. Write a script for the above.

Two examples at present

Spreadsheet HW

UNIX make

[cf. state of a program in problem solution conceived]
via family of observables of data structures

Conventional abstractions are implicit variants of agent management strategies

Issues DP \leftrightarrow modelling port interface

DP \leftrightarrow distributed programming

DP \leftrightarrow sequential prog. abstraction

procedures	calling strategies
for-loops	iteration
assignment	record agent effects
recursion	distribute receipt
data structure	

Disadvantages

1. efficiency vs flexibility trade-off
2. interfacing with conventional prog.
3. data structures

• Implementation in concurrent environment problematic m/c dependence

of state of a program or folder within a network
are forms of abstraction or implicit control of system
management strategies