**Lecture Th2: Observations, Continuity and Events**

Is the ADM a machine?

Is the ADM only suitable for discrete event simulation?

Need

- to represent continuous variables cf VCCS
- to consider implications of modelling real-time processes
- to interpret events, cf "station master has whistled" in RSA

event <--> instantaneous changes of state

Cycles of the ADM as clocked like conventional machine cycles?

.

**Time is not a primitive?**

In definitive programming, time is not a primitive concept:

e.g. in script to model relationships in a Hookes' Law experiment

– not concerned about *when* observations are made

In ADM: transitions between machine states

<--> transition between states of the experiment

NOT instantaneous, but

experimentor not concerned with "state between observations"

Cf railway station animation:

activities modelled as atomic, BUT take time. If unpacked

- might involve activities with hidden potential for interference

  e.g. synchronisation between communications

- might indicate how interference is resolved

  e.g. physical restrictions on access to a door handle

**More appropriate interpretation**

**for ADM computational states?**

each state = family of conceptually instantaneous observations

=> ADM execution defined by *a convention for observation*.

Is discrete nature of the ADM computational model a limitation?

NO? processes can be continuous, but observations are discrete

cf classical mathematics

*define* continuity by quantifying over sets of discrete observations:

"f(x) continuous at point t if given ε, there exists a δ such that ..."

i.e. continuity refers to *properties f is seen to have*

*when we are free to choose the convention for observation:*

"Tell me what you want to see – I will show you how to observe it".

Don't interpret ADM as fixed duration machine cycle:

ADM is modelling discrete observations made

"at points of time as close as the programmer wishes to choose"

programmer then exploits

a very strong privilege to intervene in computation:

"as if to change the mode of machine execution

in response to the results of computation"

Danger of circularity re use of time here ........

**Circularity re use of time?**

Proposed *observation* as more primitive concept than *time*

don't invoke time in modelling Hookes' Law

YET prescription for more frequent observation via

"instants of time that can be arbitrarily close".

To eliminate circularity

introduce time as a particular kind of experimental observation

cf experiments that intrinsically involve time

IE include observations of a clock in ADM model:

make observations relative to current observed value of clock.

**Semantics of analogue variables**

Analogue variables are defined via:

x is some function of its previous value x'

Conceptually x' is the value of x "just before now"

the ' values are defined relative to the clock

Involves approximation but no cyclic definition

value *now* is defined with reference to value *then*

faithfulness to observation subject to

"if the clock step for observation is small enough"

events are conditions of the form "x is this and y was that":

x==1 and y'==0.

cf modelling of analogue geometric entities, as in lines demo

analogue of the clock step size = arithmetic precision

Interpretation of the lines demo presumes

*arithmetic precision* can be adjusted on-line

in response to proximity of singular configurations.

**Illustrative examples** are analogue variables in the VCCS

cf sampling rate for the integrator in the speed transducer

*designer's choice based on engineering considerations*

step-size for the integration to model Newton's Second Law

*parameter chosen arbitrarily small as observation demands*

**Case study**

**The Billiards Simulation**

Problem:

simulate smooth collision of two billiard balls A and B

for simplicity, suppose:
A is stationary
B approaches with velocity V

idealised assumption predicts
direction of motion after impact independent of speed

in practice, not realised by integration with fixed step-size

e.g. if V is too great, then
balls overlap in simulation to unreasonable degree
*[ may interpret overlap as deformation of balls on impact ]*

whatever fixed step-size, there's a problem
e.g. at some speed, won't even detect impact

need realistically small degree of overlap in simulation
to estimate point of impact adequately

can change the step-size adaptively to cope with higher speed

in principle, we can cope with "arbitrary" speeds this way
if we don't mind how slowly the simulation runs!

in reality, there is a maximum limit on V

– the speed of light (?)
– the realistic maximum speed of a billiard ball

Choice of parameters in the adaptive strategy reflects:

- how fast billiard balls travel
- how accurately we need to estimate point of impact
- how fast we want the simulation to run

.... illustrates importance of interactive / incremental development