

modelling state

looking within

---

---

---

---

---

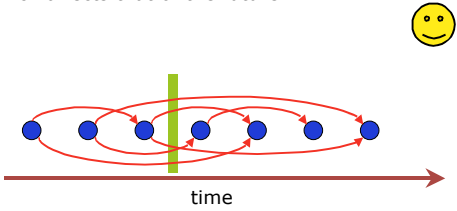
---

---

---

### what is state

that in the present  
of that in the past  
which affects that of the future




---

---

---

---

---

---

---

---

### modelling state

- describe state using variables
- types of variables:
  - basic type:  
x: Nat                    - non-negative integer {0,1,2,...}
  - individual item from set:  
shape: {circle, line, rectangle}
  - subset of bigger set:  
selection: **set** Nat                    - set of integers
  - function (often finite):  
objects: Nat → shape
  - user defined:  
Point = [x, y: Real]                    - e.g. (1.79,-3.2)

---

---

---

---

---

---

---

---

## stages

iteratively define:

- state - what needs to be remembered
- invariants - what is always true
- initial state - how it starts
- actions - what can happen to the state  
(need to relate this to keys etc.)
- display - what the user sees (hears etc.)

use scenarios to check they are what you want

---

---

---

---

---

---

---

---

## four function calculator

- formal description of the state
- define the effect of the following actions:
  - type\_digit(d) - user presses single digit
  - equals - user presses '=' button
  - op(p) - user presses '+', '-', '\*' or '/' button

N.B. will not be right first time ... spot the mistakes

---

---

---

---

---

---

---

---

## calculator state - first attempt

state \_\_\_\_\_  
total: Nat - running total (accumulator)  
disp: Nat - number currently displayed  
*no invariants*

initial state \_\_\_\_\_  
total = 0  
disp = 0

display \_\_\_\_\_  
disp - more complex calculator may show formulae

---

---

---

---

---

---

---

---

### calculator actions - first attempt

type\_digit(d) \_\_\_\_\_  
add d to the end of disp  
total unchanged

equals \_\_\_\_\_  
do last operation "+,-,\*,/" to disp and total  
...

what is it!

---

---

---

---

---

---

---

---

### calculator state - second attempt

state \_\_\_\_\_  
total: Nat - running total (accumulator)  
disp: Nat - number currently displayed  
pend\_op: {+, -, \*, /, none} - pending operation

initial state \_\_\_\_\_  
total = 0  
disp = 0  
pend\_op = none

---

---

---

---

---

---

---

---

### calculator actions - second attempt

type\_digit(d) \_\_\_\_\_  
add d to the end of disp  
total and pend\_op unchanged

equals \_\_\_\_\_  
do pend\_op to disp and total  
put result in both disp and total  
set pend\_op to none

op(o) \_\_\_\_\_  
do pend\_op to disp and total  
put result in both disp and total  
put o into pend\_op

---

---

---

---

---

---

---

---

## calculator - scenario

- user types: 1 + 2 7 = - 3
- start after 1 + 2

action	total	disp	pend_op
	1	2	+
type_digit(7)	1	27	+
equals	28	28	none
op(-)	28	28	-
type_digit(3)	28	283	-



---

---

---

---

---

---

---

---

## calculator state - third attempt

state

total: Nat - running total (accumulator)  
disp: Nat - number currently displayed  
pend\_op: {+, -, \*, /, none} - pending operation  
typing: Bool - true/false flag

- added 'typing' flag  
- user in the middle of typing a number

---

---

---

---

---

---

---

---

## calculator actions - third attempt

type\_digit(d)

if typing then add d to the end of disp  
otherwise clear disp and put d in it  
also set typing to true  
total and pend\_op unchanged

equals and op(o):

- as before except both set typing to false

---

---

---

---

---

---

---

---

## calculator - scenario revisited

- user types: 1 + 2 7 = - 3
- start after 1 + 2

action	total	disp	pend_op	typing
type_digit(7)	1	2	+	yes
equals	1	27	+	yes
op(-)	28	28	none	no
type_digit(3)	28	28	-	no
	28	3	-	yes



---

---

---

---

---

---

---

---

## defining state

two problems:

- too little state
  - elements missing from specification
  - may be deliberate
  - e.g. dialogue level spec.
- too much state
  - too many states, too complex state
  - may be deliberate
  - redundancy, extensibility

---

---

---

---

---

---

---

---

## too little state

- forgotten elements
  - e.g. 'typing' flag for calculator
- checking:
  - dialogue state
    - can you work out current dialogue state?
  - action specification
    - do you have enough information?
  - implicit global variables (see also later)
    - suggest state missing

---

---

---

---

---

---

---

---

## too much state

- unreachable states
  - too few actions (see later)
  - constraints
  - states are not orthogonal
- spare variables: constant/functional dependent
- dependent state
  - e.g. first point of line, number being typed
- indistinguishable states
  - what is observable?

---

---

---

---

---

---

---

---

## defining actions

- framing problems
  - = too little in result state
- unreachable states - insufficient actions
- using 'global' variables
  - implicit in operation definition
- beware extreme cases
  - (e.g. empty document, cursor at end of line)

---

---

---

---

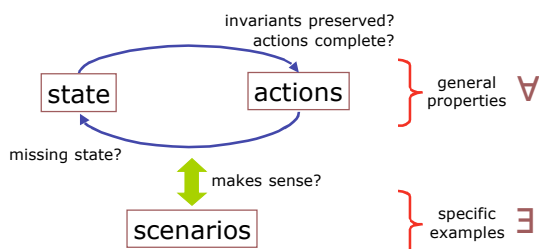
---

---

---

---

## internal and external consistency



---

---

---

---

---

---

---

---