CONSTRUIT!

A Curriculum for Making Construals

(A work in progress)

Disclaimer

- *This is a prototype curriculum that follows a general plan for learning to make construals*
- *The examples used are not yet up-to-date and will be revised appropriately*
- *The out-of-date version of the environment for making construals is also to be revised*

Session 1

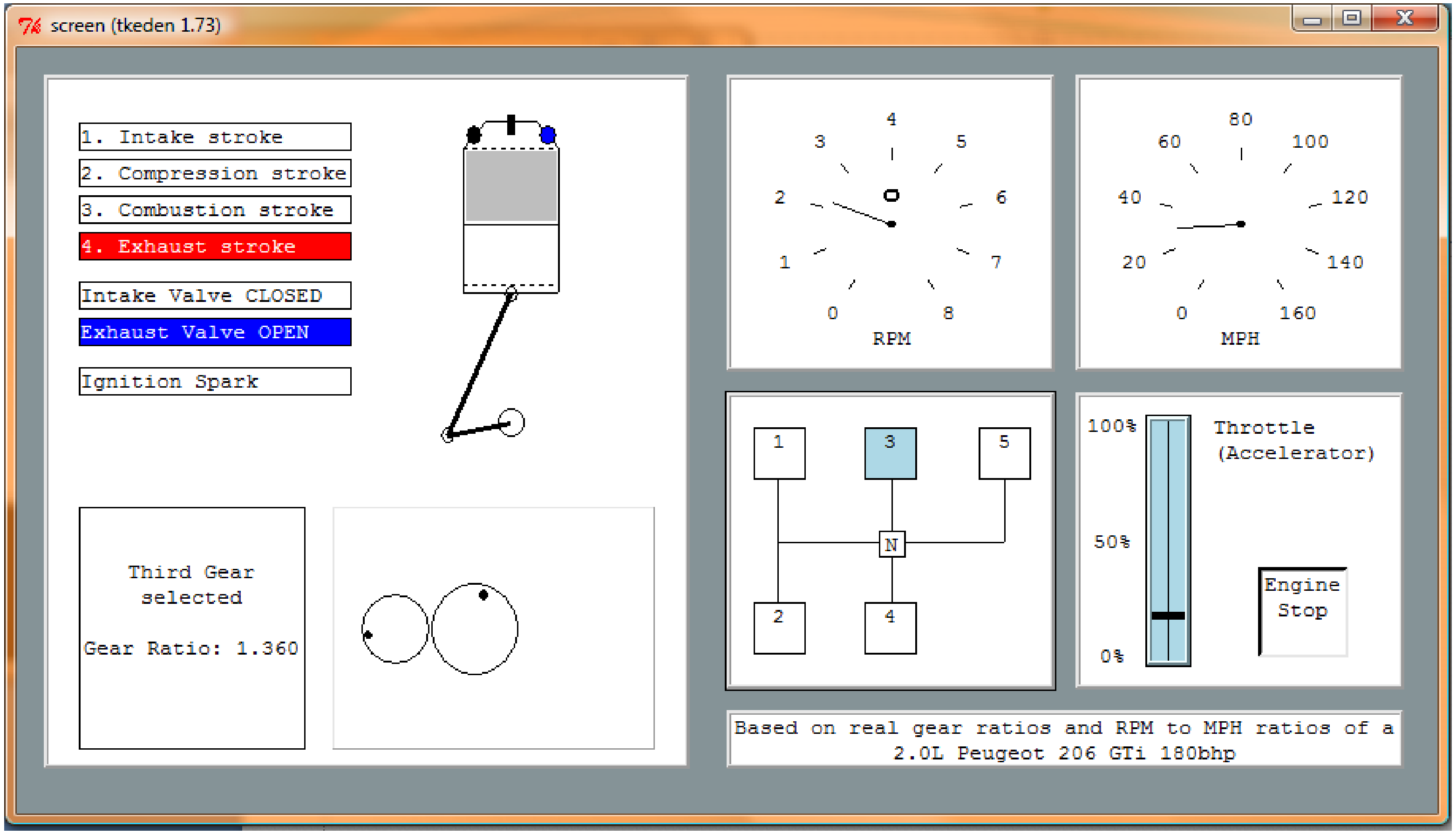
- Orientation on Making Construals
- Examples of construals

Examples of construals

- A car engine [enginewithgearsSidbury2010]
- Playing noughts-and-crosses [oxoGardner1999]
- A room of your own [roomdemolabShao2012]
- Adventures in a lift ...

*The **ICE, OXO Lab** & **Adventures in a Lift** construals have counterparts in the Construit archive*

An engine with gears construal



screen (tkeden 1.73)

1. Intake stroke
2. Compression stroke
3. Combustion stroke
4. Exhaust stroke

Intake Valve CLOSED

Exhaust Valve OPEN

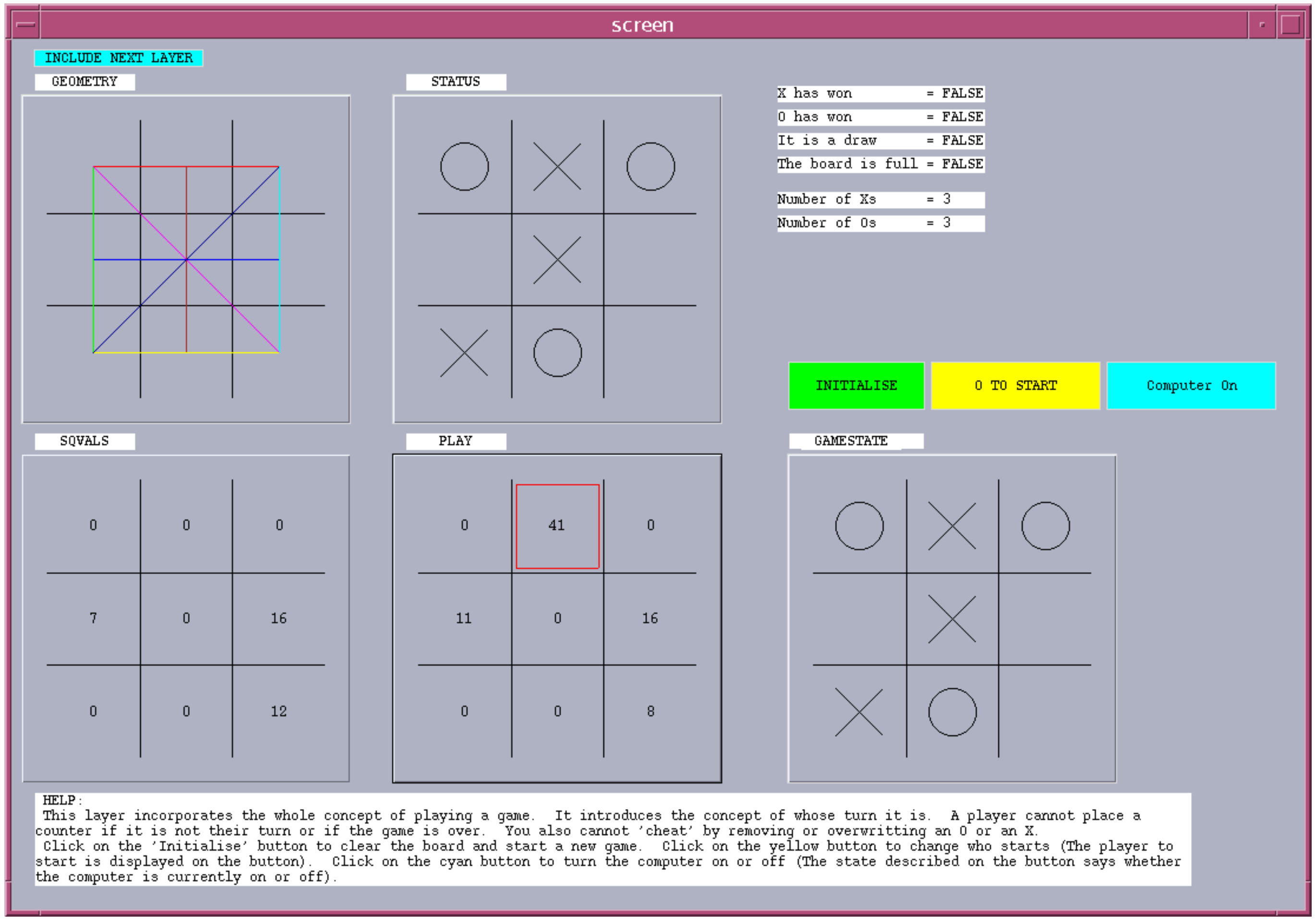
Ignition Spark

Third Gear selected

Gear Ratio: 1.360

Based on real gear ratios and RPM to MPH ratios of a 2.0L Peugeot 206 GTi 180bhp

Playing noughts-and-crosses



```
X has won          = FALSE
0 has won          = FALSE
It is a draw       = FALSE
The board is full  = FALSE

Number of Xs       = 3
Number of 0s       = 3
```

SQVALS

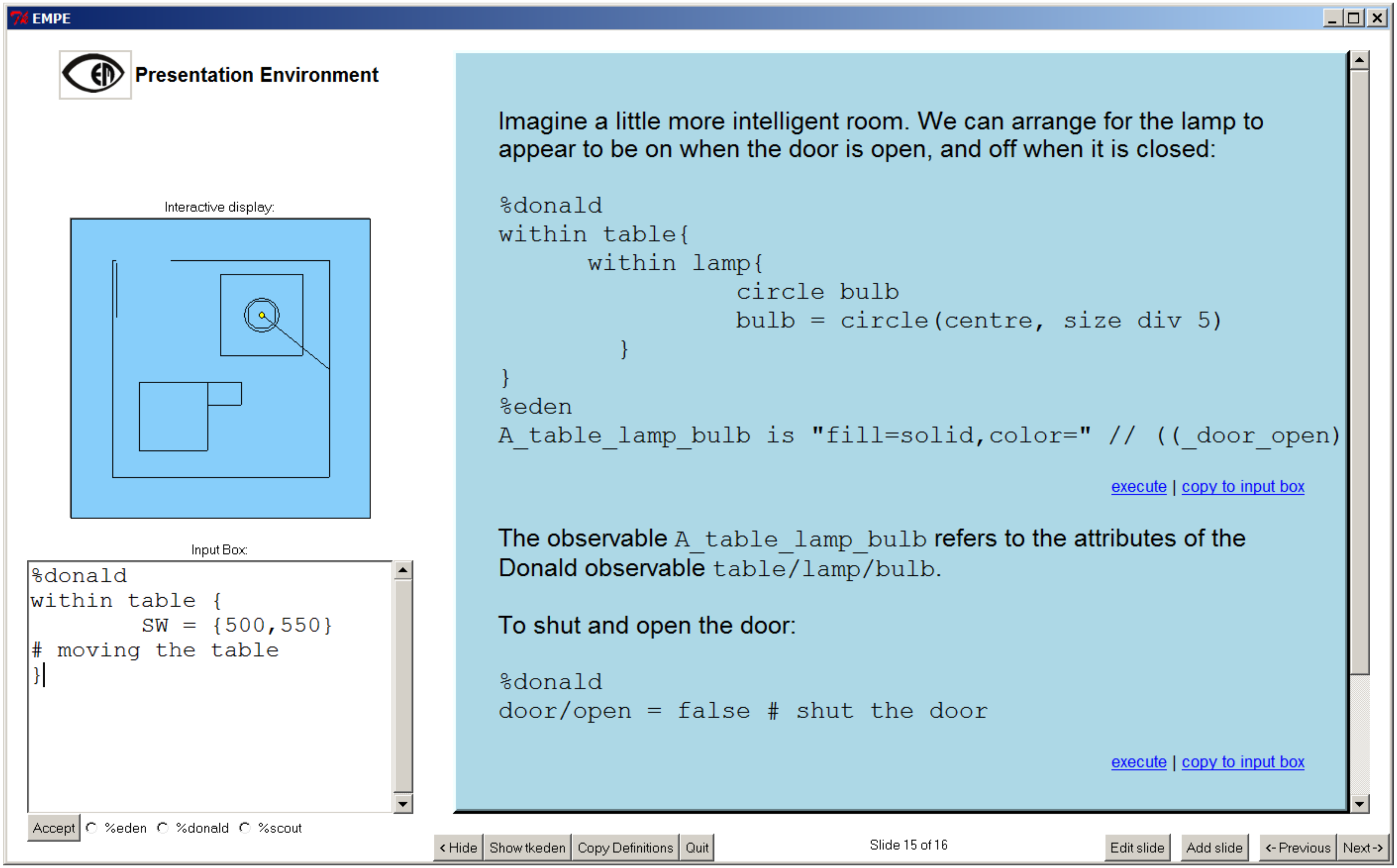
0	0	0
7	0	16
0	0	12

PLAY

0	41	0
11	0	16
0	0	8

HELP:
This layer incorporates the whole concept of playing a game. It introduces the concept of whose turn it is. A player cannot place a counter if it is not their turn or if the game is over. You also cannot ‘cheat’ by removing or overwritting an 0 or an X.
Click on the ‘Initialise’ button to clear the board and start a new game. Click on the yellow button to change who starts (The player to start is displayed on the button). Click on the cyan button to turn the computer on or off (The state described on the button says whether the computer is currently on or off).

A construal of a room



```
%donald
within table {
        SW = {500,550}
# moving the table
}
```

Imagine a little more intelligent room. We can arrange for the lamp to appear to be on when the door is open, and off when it is closed:

```
%donald
within table{
      within lamp{
                circle bulb
                bulb = circle(centre, size div 5)
        }
}
%eden
A_table_lamp_bulb is "fill=solid,color=" // ((_door_open)
```

execute | copy to input box

The observable `A_table_lamp_bulb` refers to the attributes of the Donald observable `table/lamp/bulb`.

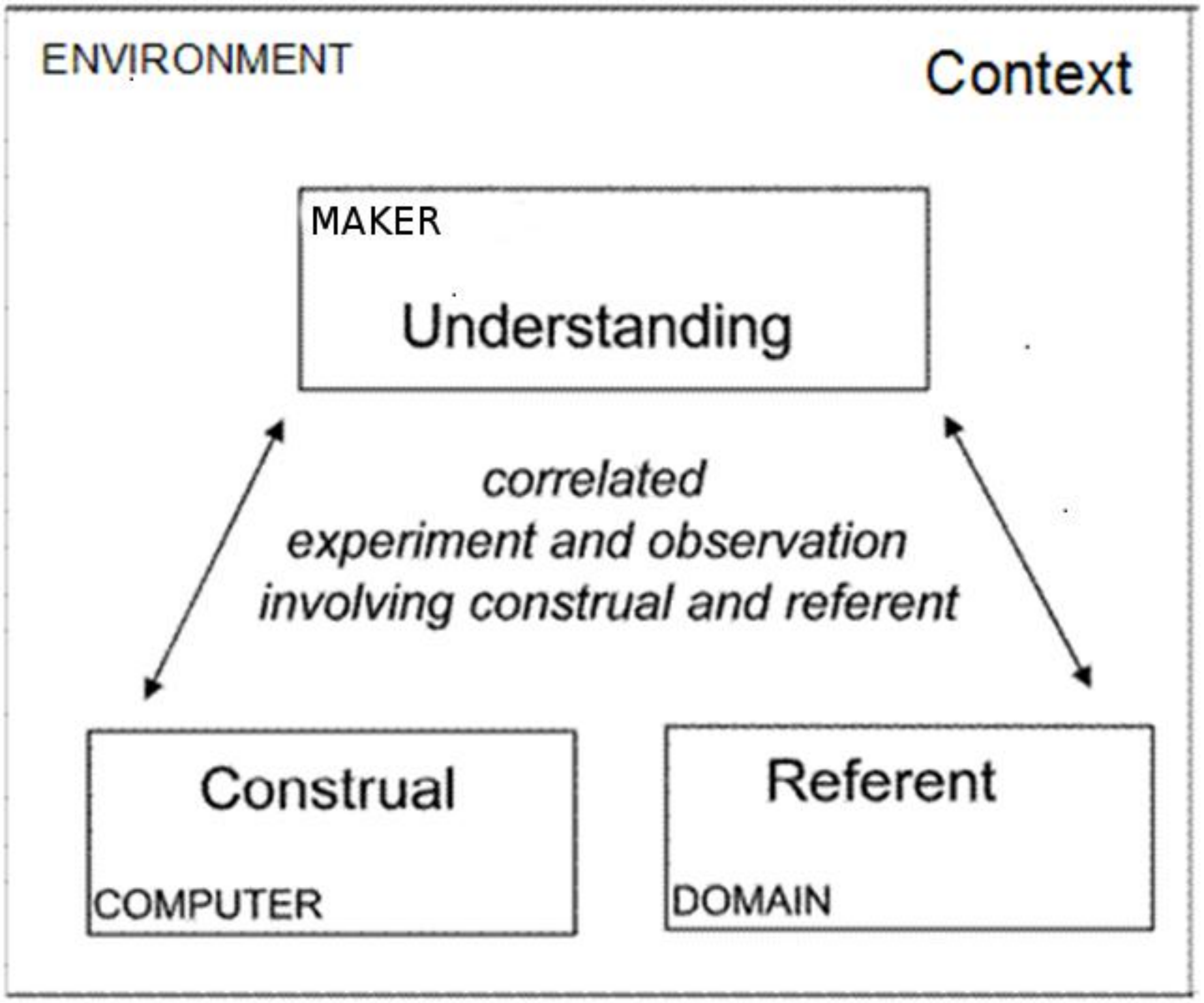
To shut and open the door:

```
%donald
door/open = false # shut the door
```

execute | copy to input box

Based on roomdemoShao2007

Making construals



Session 2

- Basic background to JS-EDEN, the prototype environment for making construals
- A first tutorial on making construals

cf. the Construit environment, and the introductory video on its use with reference to the Dog construal

Basics of the prototype MCE

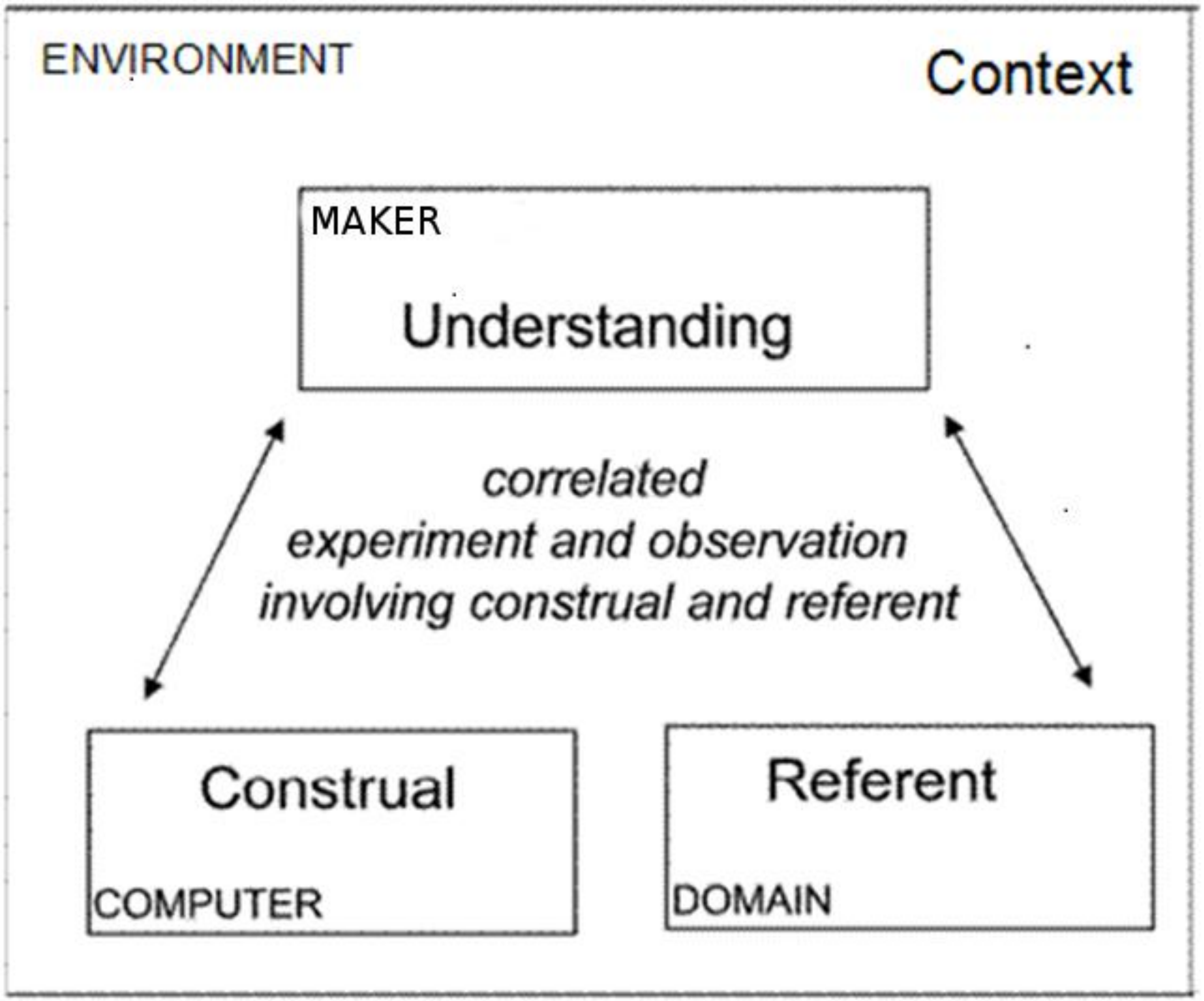
- The EDEN handbook as it applies to JS-EDEN
- Prerequisite knowledge and skill
- Configuration issues – use of an editor etc
- Basic exercises and general technical guidance
- Some initial practical exercises

- *the EDEN handbook is no longer as topical*
- *configuration issues are now irrelevant*

Session 3

- A fundamental diagram in making construals
- Basic concepts in making construals
- Principles for making construals
- Further illustrative examples of construals

Making construals



Orientation

- Experience
  - Awareness of experience [Dewey]
- Classification of experience
  - observables / dependency / agency

Concrete and situated examples informing key abstractions in making construals

Abstractions from experience

Ingredients common to all three examples:

- you as maker
- your construal
- its referent
- your context

... the fundamental diagram

Character of the diagram

A slice through an ongoing interactive experience:

- the **construal**
- its **referent**
- the maker’s **understanding**
- the **context** are all co-evolving

Session 4

- From ODA to definitions, functions, actions
- Scripts as static, dynamic, versioning texts
- An illustrative practical study via bubblesort

Construit now offers radically different options

- using the with-construct in place of functions

- using the when-construct in place of actions

An environment for making construals

Symbol list comprising

- Definition list – observables + dependencies
- Function list – framing dependencies
- Action list – automating agency

Abstract dependency relationships

dependency map

These viewing features have been revised in Construit

An environment for making construals

Symbol lookup table: Explicit dependencies

Script manipulation

- history / script generator / state timeline
- restoring state
- merging state

These mechanisms have been revised in Construit

Session 5

- JS-EDEN introductory lab
- Environments, instruments and tools
- Simple interactive activities using JS-EDEN
- Harfield's Numberline model

The ***Solar System*** *and* ***Construing the Moment*** *construals are Construit exercises of similar complexity*

Session 6-9

Construction through conjunction as seen in:

- relating construals to programs (S6)
- identifying observables through interaction (S7)
- realising understanding as a stream-of-thought (S6)
- commonsense perception of concurrence

as horizontal, vertical and orthogonal relations illustrated by sample construals (S8-S9)

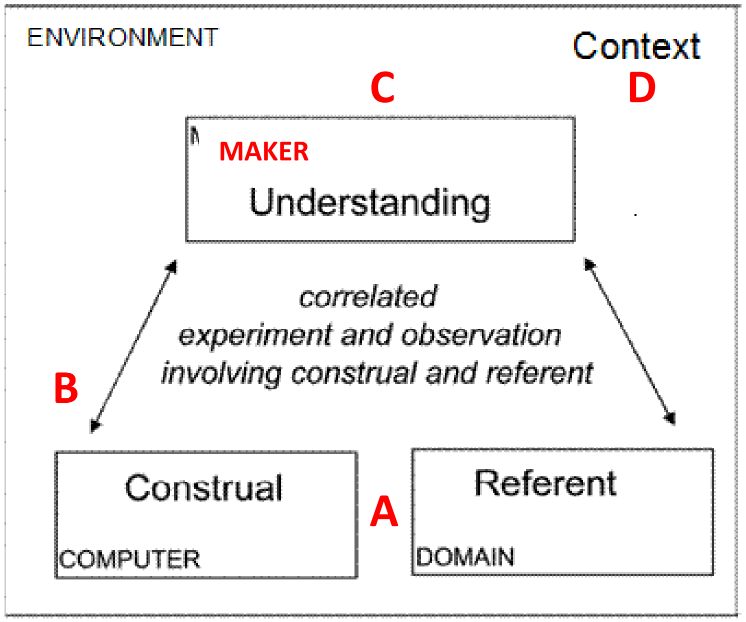
The "Fundamental Diagram of EM"

A - correlation in experience

B - construal as embodied in latent patterns of meaningful interaction

C - understanding as awareness of patterns of meaningful interaction

D - context subject to evolve, or to be revised by the maker at will



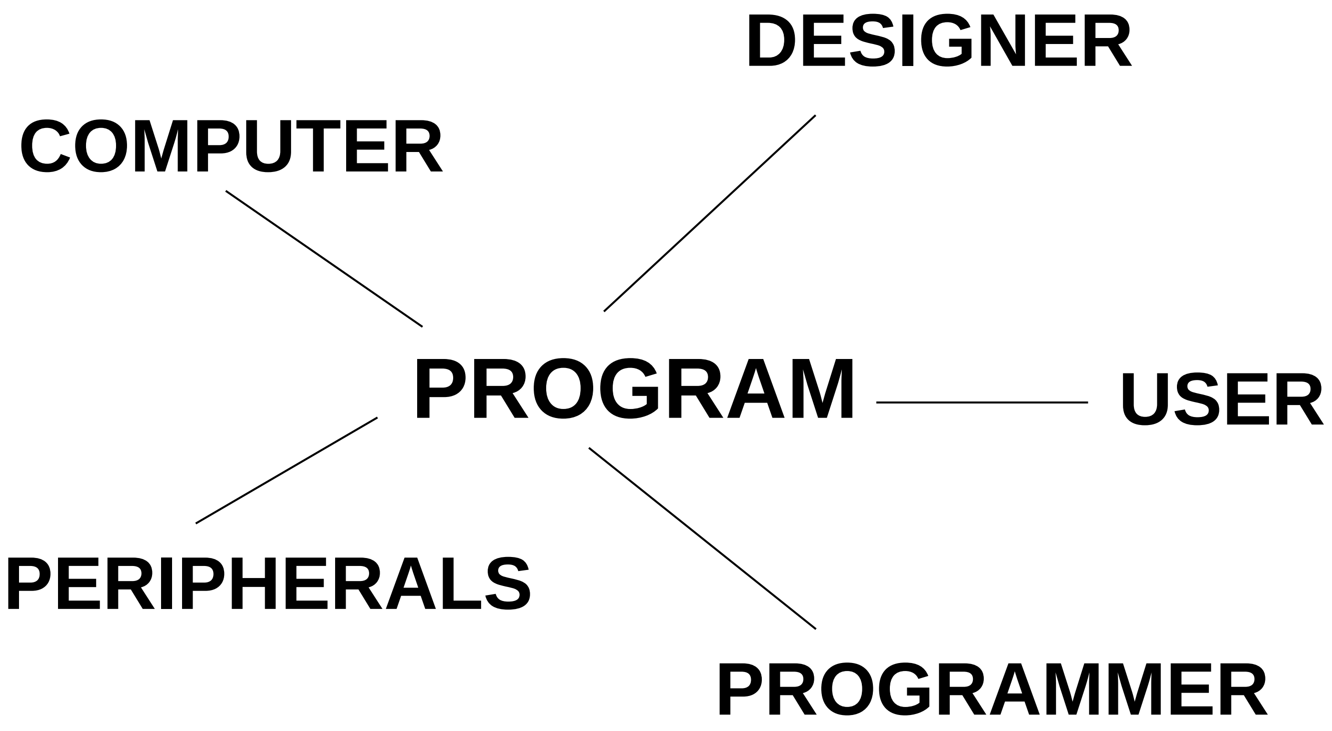
A –the semantics of construals cf. digit-cabinet, lines
B – cf. malaria / lift adventure
C – what it means to play a game of noughts and crosses / using vi editor
D – the experimental paradox / making the transition from construal to program

Session 6

- From construals to programs
- Contexts established at the maker's discretion

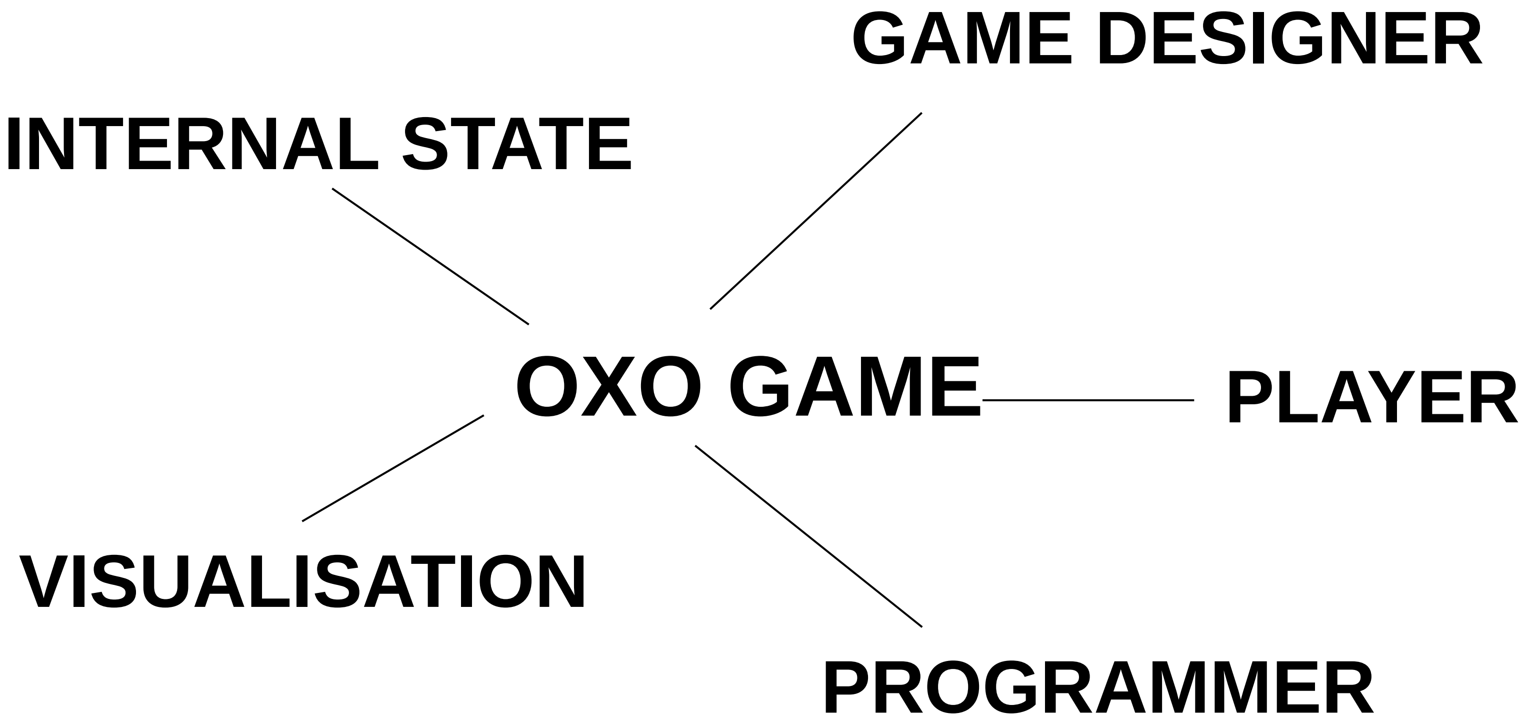
- *There is not yet a good counterpart in Construit for the heapsorting construal*

- *The **Giving Change** and **OXO Lab** construals are relevant*



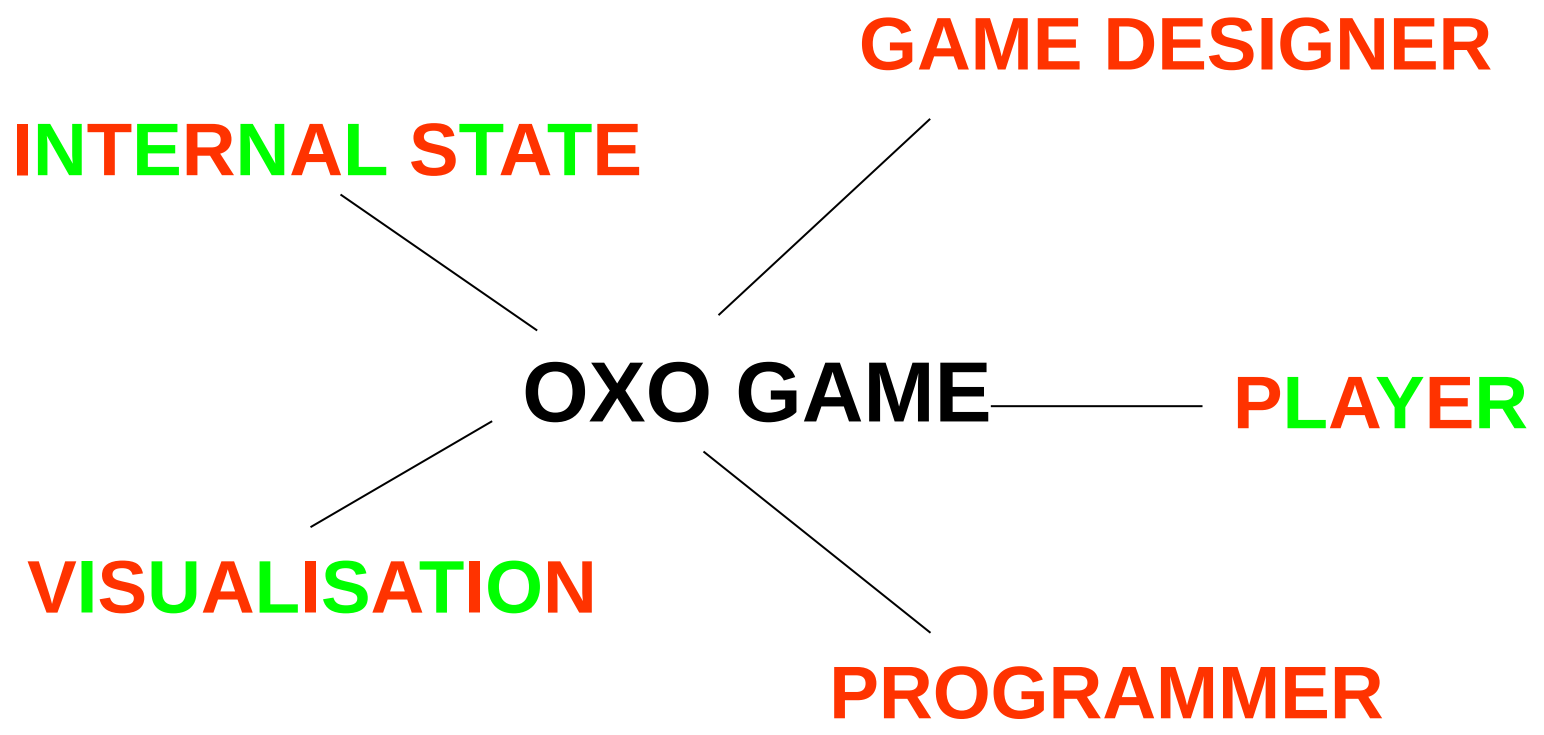
Diverse relations / representations in a traditional program

… compare this with the OXO laboratory

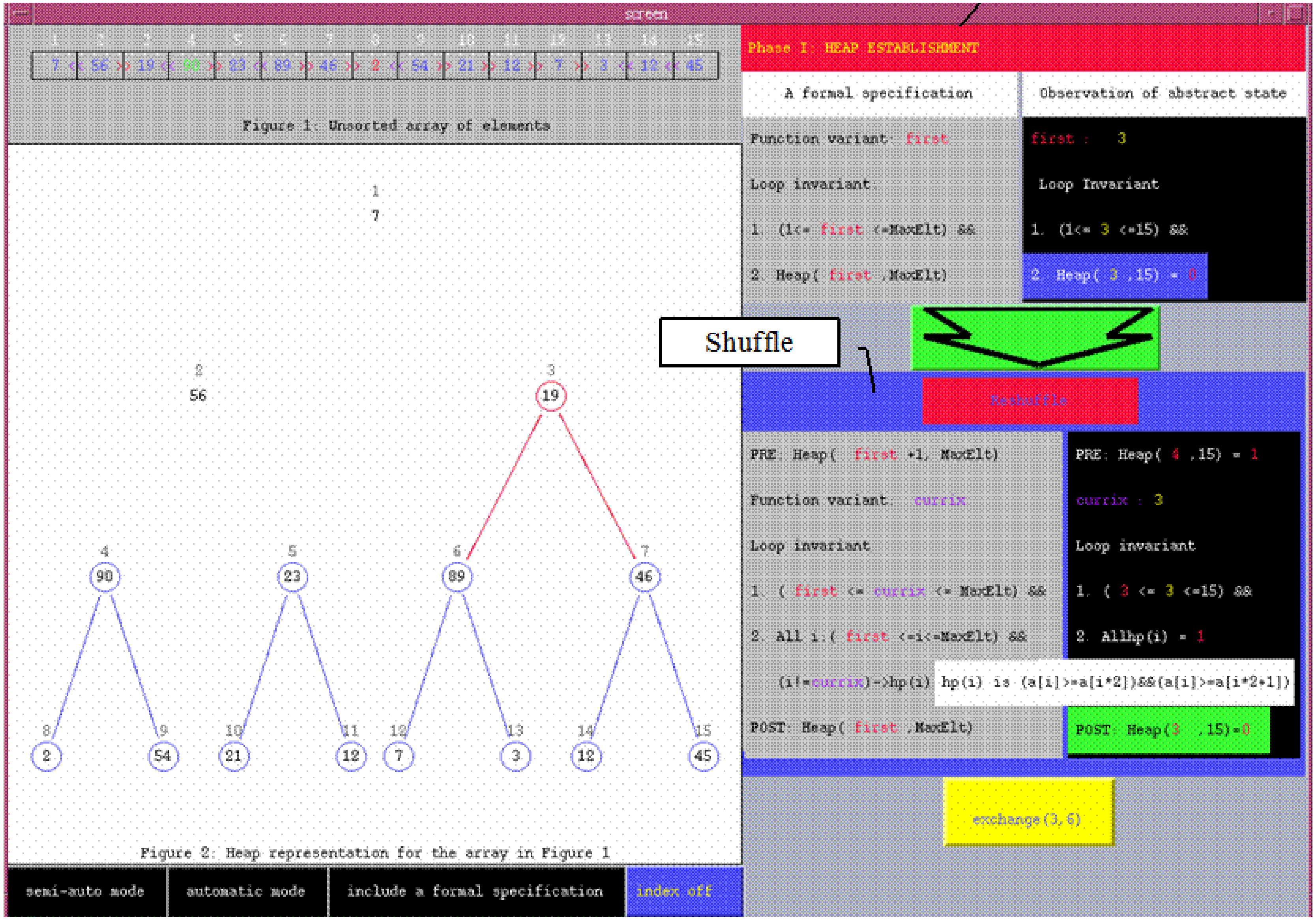


… all relations mediated by definitions

… Behaviour as programmed state change



Static and dynamic elements of state



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	56	19	90	23	89	46	2	54	21	12	7	3	12	45

Figure 1: Unsorted array of elements

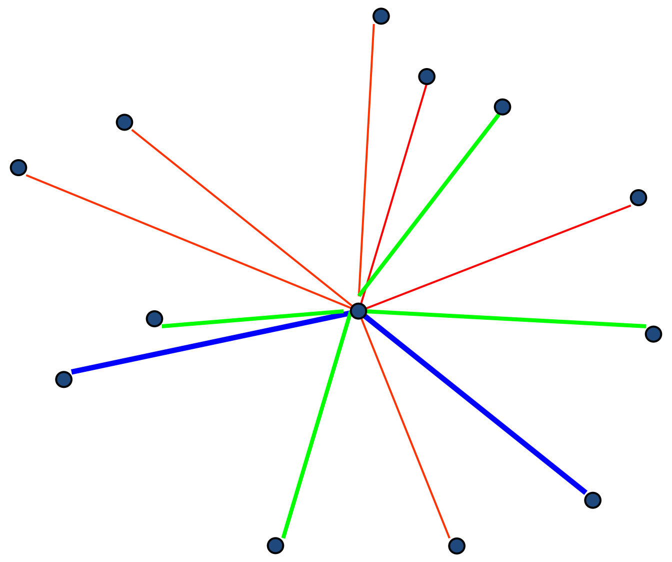
Figure 2: Heap representation for the array in Figure 1

Phase I: HEAP ESTABLISHMENT

A formal specification	Observation of abstract state
Function variant: first	first : 3
Loop invariant:	Loop Invariant
1. (1<= first <=MaxElt) &&	1. (1<= 3 <=15) &&
2. Heap(first ,MaxElt)	2. Heap(3 ,15) = 0
PRE: Heap(first +1, MaxElt)	PRE: Heap(4 ,15) = 1
Function variant: currix	currix : 3
Loop invariant	Loop invariant
1. (first <= currix <= MaxElt) &&	1. (3 <= 3 <=15) &&
2. All i:(first <=i<=MaxElt) &&	2. Allhp(i) = 1
(i!=currix)->hp(i)	hp(i) is (a[i]>=a[i*2])&&(a[i]>=a[i*2+1])
POST: Heap(first ,MaxElt)	POST: Heap(3 ,15)=0

"Formal specification from an observation-oriented perspective"

Definitive scripts as "germs of a construal"



● ≡ a definitive script

╲ ≡ a nonsense redefinition

╱ ≡ a plausible redefinition

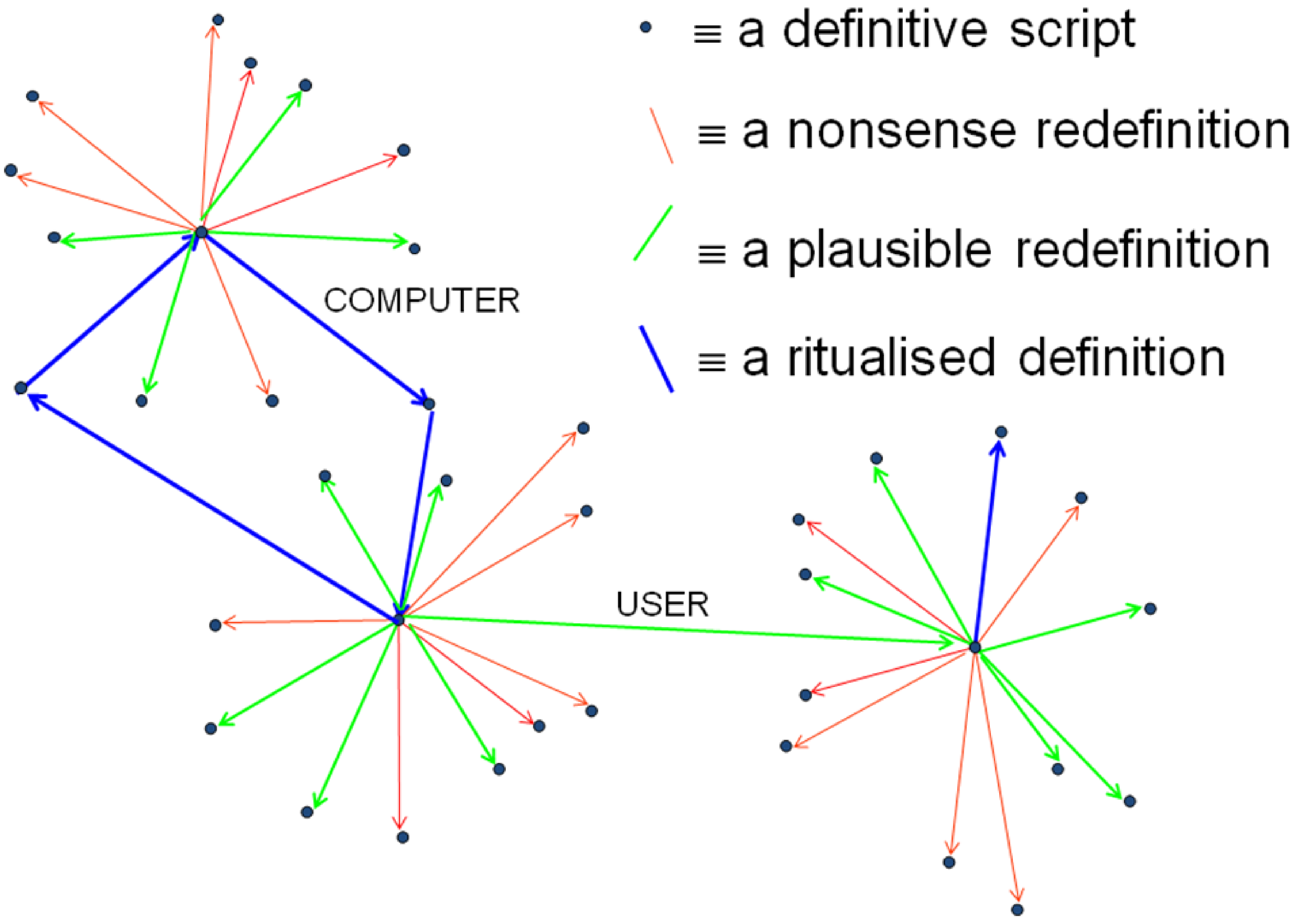
╲ ≡ a ritualised definition

Plausible : *could* open the desk drawer

– note continuous spectrum of redefinitions

Ritualised : door *automatically* closes after being opened

Nonsense : opening the drawer makes the room smaller


● ≡ a definitive script
╲ ≡ a nonsense redefinition
╱ ≡ a plausible redefinition
╲ ≡ a ritualised definition
COMPUTER
USER

3 ingredients in construal development:

- engineering the states within which the agency of the user and the computer operate;
- crafting the behaviours which these agents then play out;
- projecting meanings on to the agent actions

"Vertical", "horizontal" and "orthogonal" dimensions of state

Different kinds of conjunction

- Perceived as concurrent – ‘vertical’ dimension
- Flowing one into another – ‘horizontal’
- Evoking associations with a referent – ‘orthogonal’

Relate to the annotated fundamental diagram: resp. developing context cf. D, patterns of interaction B +C, and semantic link A

Key features of making a construal

- opens up such a profusion of possible interpretations, stimulating the model-builder's imagination and creativity.
- is an open-ended activity that resembles organic growth rather than building to a specification

Session 7

- Observables as conjunctions in experience
- Construction as conjunction

A famous quote from Heraclitus

“No man ever steps in the same river twice, for it's not the same river and he's not the same man.”

- In its proper context, this is great wisdom ...
- ... on the other hand, how perverse it would be to disregard the perceptions of sameness in men and rivers ....
- We can choose (“have discretion”), and because we have a choice we *construct* our context

Fundamental perspective in EM

Perceived connections

= connections *given-in-experience*

= conjunctive relations (William James – 1910)

What is meant by *experience* here? (Dewey)

Key concepts

The ODA framework

- observables, dependency and agency
- different varieties of perceived connection

LSD: “language for specification and description”

- Classification of observables
  - states, oracles, handles, derivates, privileges

Perceived connections ...

An **observable**: same identity different status

Cluster of observables resembles an object

Changes to observables connected by **dependency**

Part of same stream-of-thought ...

- successive positions “in the same game”
- lectures in the same module

Perceived connections ...

Cluster of observables resembling an object co-existing as coming and going ‘at the same time’ – potentially an **agent**

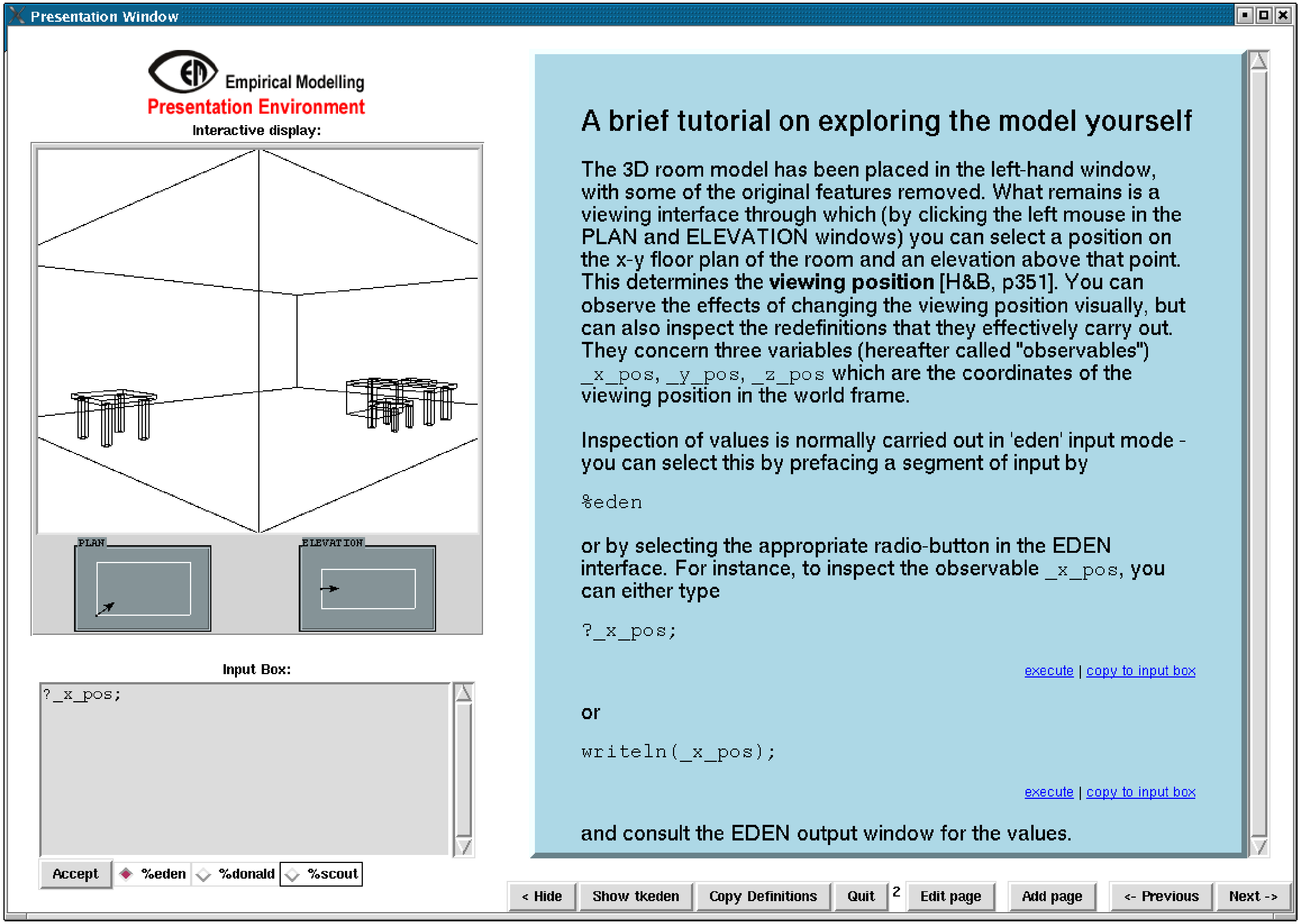
Being concurrent in the present moment

Changes being associated with / attributed to a specific agent

Session 8

- Illustrative examples of construals

The graphics presentation and the vision-related neuron construals have counterparts in Construit. There are other elaborate examples of construals.



```
?_x_pos;
```

A brief tutorial on exploring the model yourself

The 3D room model has been placed in the left-hand window, with some of the original features removed. What remains is a viewing interface through which (by clicking the left mouse in the PLAN and ELEVATION windows) you can select a position on the x-y floor plan of the room and an elevation above that point. This determines the **viewing position** [H&B, p351]. You can observe the effects of changing the viewing position visually, but can also inspect the redefinitions that they effectively carry out. They concern three variables (hereafter called "observables") `_x_pos`, `_y_pos`, `_z_pos` which are the coordinates of the viewing position in the world frame.

Inspection of values is normally carried out in 'eden' input mode - you can select this by prefacing a segment of input by

```
%eden
```

or by selecting the appropriate radio-button in the EDEN interface. For instance, to inspect the observable `_x_pos`, you can either type

```
?_x_pos;
```

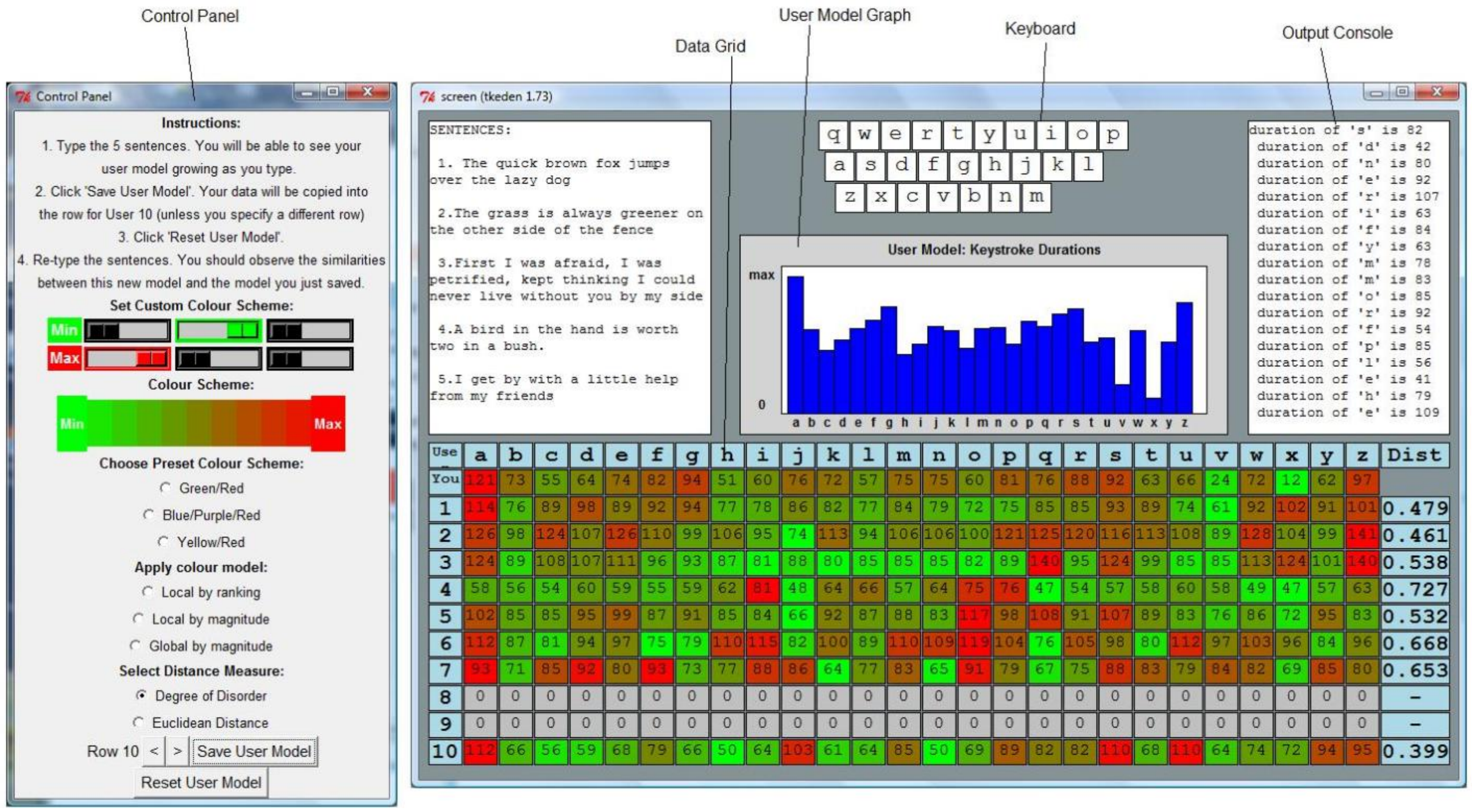
execute | copy to input box

or

```
writeln(_x_pos);
```

execute | copy to input box

and consult the EDEN output window for the values.



Instructions:

1. Type the 5 sentences. You will be able to see your user model growing as you type.
2. Click 'Save User Model'. Your data will be copied into the row for User 10 (unless you specify a different row)
3. Click 'Reset User Model'.
4. Re-type the sentences. You should observe the similarities between this new model and the model you just saved.

Set Custom Colour Scheme:

Choose Preset Colour Scheme:

- Green/Red
- Blue/Purple/Red
- Yellow/Red

Apply colour model:

- Local by ranking
- Local by magnitude
- Global by magnitude

Select Distance Measure:

- [✓ Degree of Disorder]
- Euclidean Distance

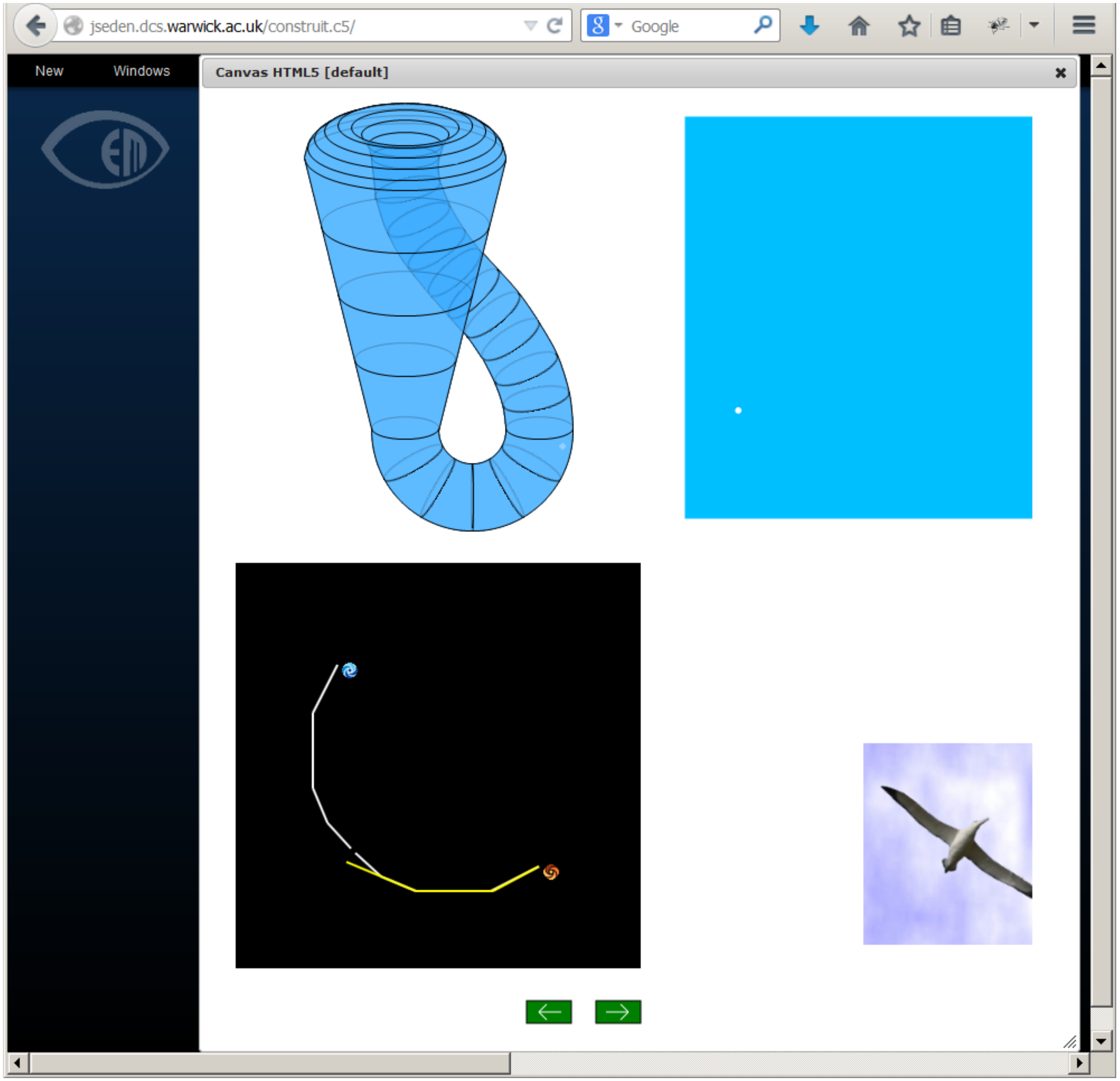
Row 10 < > Save User Model

Reset User Model

```
SENTENCES:

 1. The quick brown fox jumps
over the lazy dog

 2.The grass is always greener on
the other side of the fence

 3.First I was afraid, I was
petrified, kept thinking I could
never live without you by my side

 4.A bird in the hand is worth
two in a bush.

 5.I get by with a little help
from my friends
```

```
duration of 's' is 82
 duration of 'd' is 42
 duration of 'n' is 80
 duration of 'e' is 92
 duration of 'r' is 107
 duration of 'i' is 63
 duration of 'f' is 84
 duration of 'y' is 63
 duration of 'm' is 78
 duration of 'm' is 83
 duration of 'o' is 85
 duration of 'r' is 92
 duration of 'f' is 54
 duration of 'p' is 85
 duration of 'l' is 56
 duration of 'e' is 41
 duration of 'h' is 79
 duration of 'e' is 109
```

Use	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	Dist
You	121	73	55	64	74	82	94	51	60	76	72	57	75	75	60	81	76	88	92	63	66	24	72	12	62	97	
1	114	76	89	98	89	92	94	77	78	86	82	77	84	79	72	75	85	85	93	89	74	61	92	102	91	101	0.479
2	126	98	124	107	126	110	99	106	95	74	113	94	106	106	100	121	125	120	116	113	108	89	128	104	99	141	0.461
3	124	89	108	107	111	96	93	87	81	88	80	85	85	85	82	89	140	95	124	99	85	85	113	124	101	140	0.538
4	58	56	54	60	59	55	59	62	81	48	64	66	57	64	75	76	47	54	57	58	60	58	49	47	57	63	0.727
5	102	85	85	95	99	87	91	85	84	66	92	87	88	83	117	98	108	91	107	89	83	76	86	72	95	83	0.532
6	112	87	81	94	97	75	79	110	115	82	100	89	110	109	119	104	76	105	98	80	112	97	103	96	84	96	0.668
7	93	71	85	92	80	93	73	77	88	86	64	77	83	65	91	79	67	75	88	83	79	84	82	69	85	80	0.653
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	–
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	–
10	112	66	56	59	68	79	66	50	64	103	61	64	85	50	69	89	82	82	110	68	110	64	74	72	94	95	0.399


jseden.dcs.warwick.ac.uk/construit.c5/
Google
New
Windows
Canvas HTML5 [default]

Session 9

- The semantics of construals
- An experiential framework for learning
- Blending the learner, teacher, developer roles
- Sense-making across many disciplines

None of the four examples of construals for sense-making displayed has a counterpart in Construit. There are as yet few similar examples of construals.

private experience / empirical / concrete

interaction with artefacts: identification of persistent features and contexts

practical knowledge: correlations between artefacts, acquisition of skills

identification of dependencies and postulation of independent agency

identification of generic patterns of interaction and stimulus-response mechanisms

non-verbal communication through interaction in a common environment

directly situated uses of language

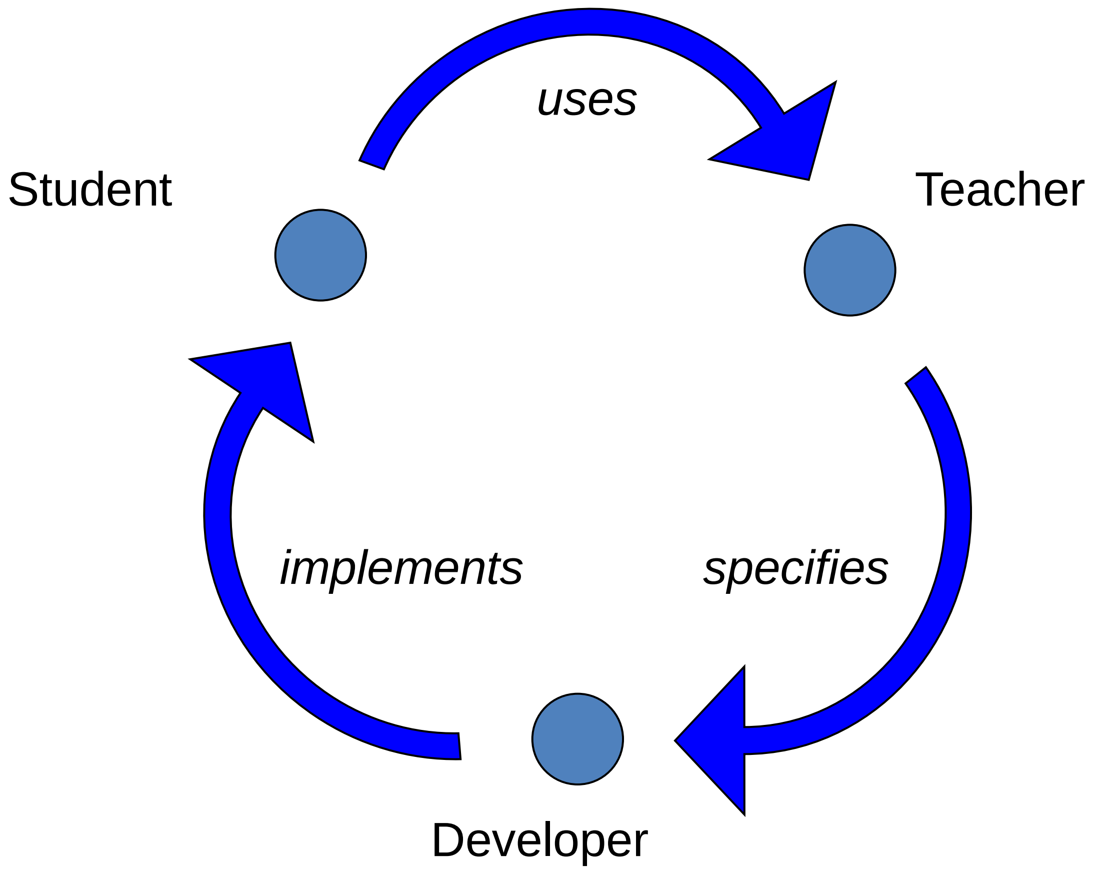
identification of common experience and objective knowledge

symbolic representations and formal languages: public conventions for interpretation

public knowledge / theoretical / formal

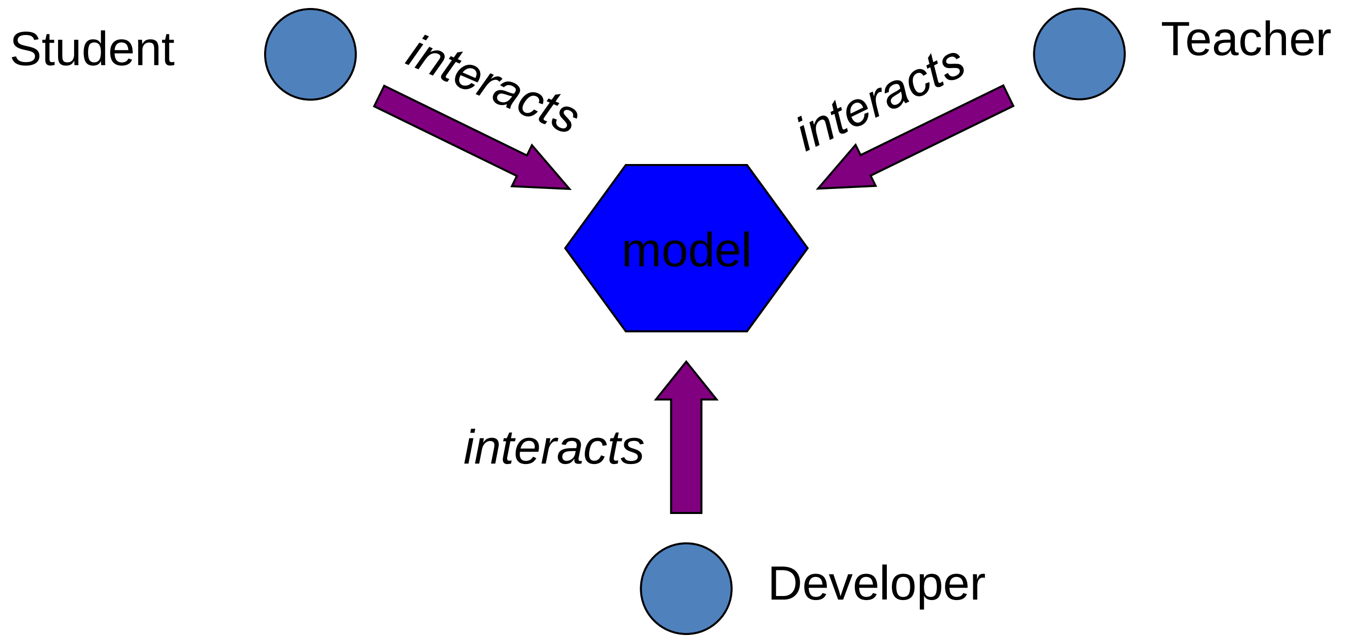
An Experiential Framework for Learning (EFL)

Developing educational software

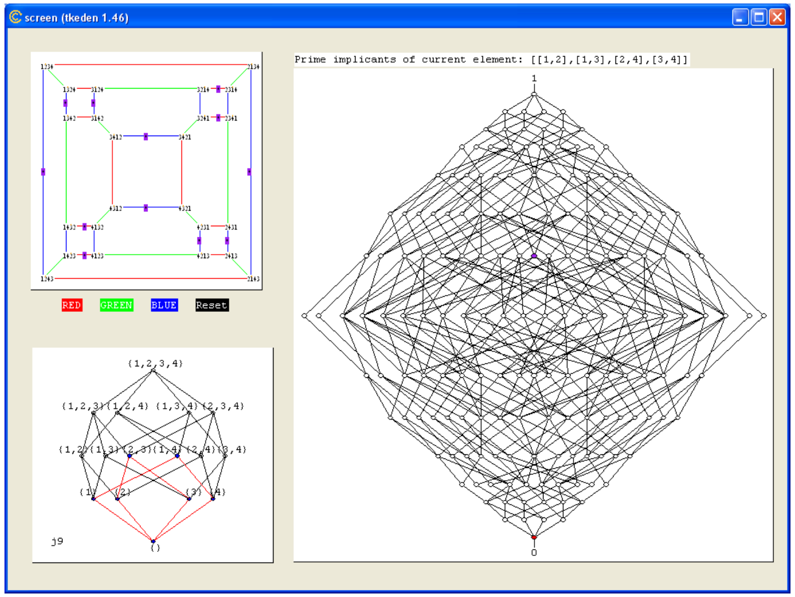


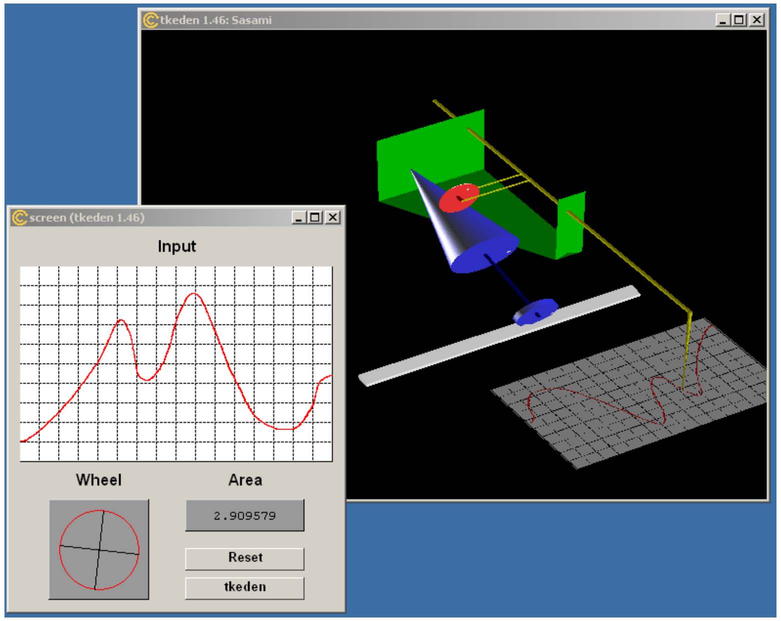
Empirical Modelling (EM)

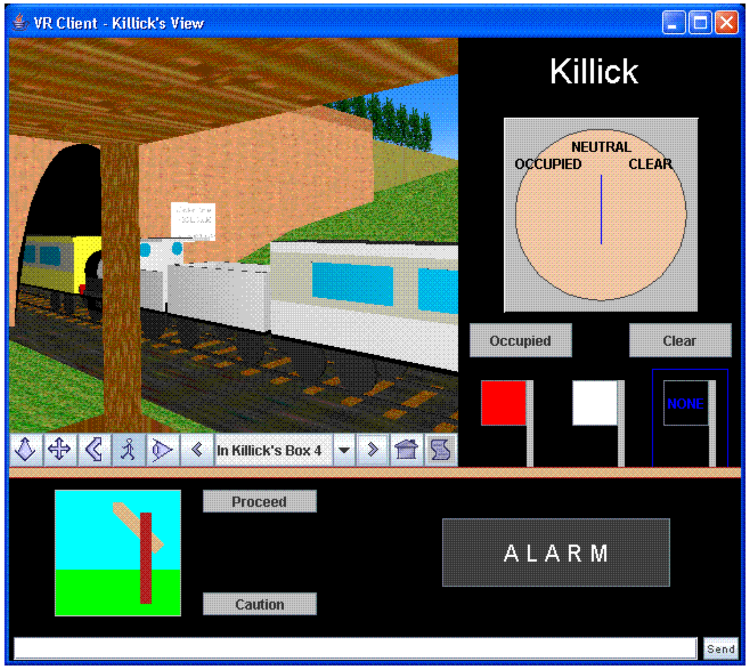
- Offers a set of principles for model building in any of the student, teacher and developer roles:

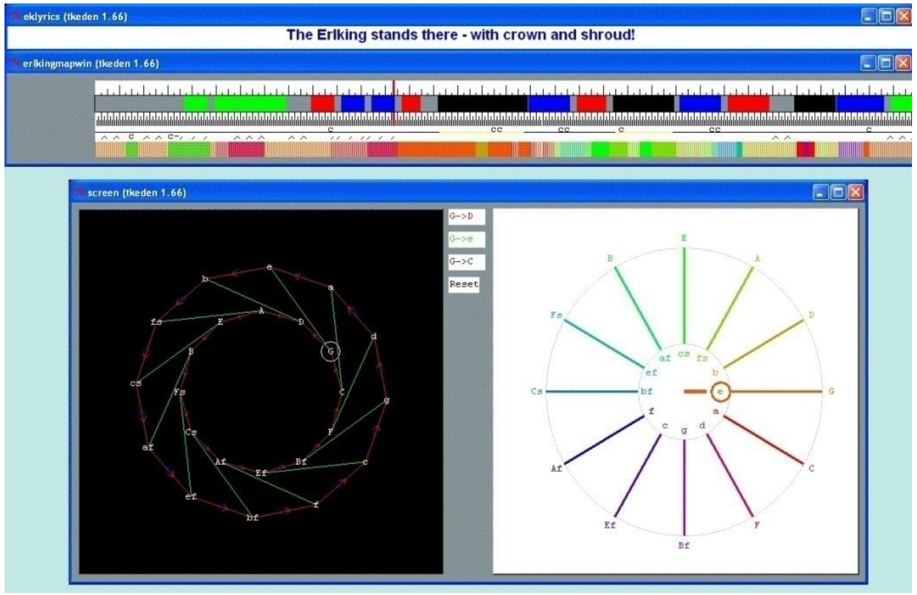


Sense-making in mathematics, in the physical world, social interactions and music ...


screen (tkeden 1.46)
Prime implicants of current element: [[1,2],[1,3],[2,4],[3,4]]
RED
GREEN
BLUE
Reset
j9






eklyrics (tkeden 1.66)
The Erlking stands there - with crown and shroud!
erlkingmapwin (tkeden 1.66)
screen (tkeden 1.66)
G->D
G->e
G->C
Reset

Session 10

Six key claims for construals, relating to:

- Accessibility
- Comprehensibility
- Scope for collaborative development
- Scope for assessment and evaluation
- Serving as a resource for creating OERs
- Wide applicability across disciplines