Prelude to lecture: a visit to the shops!

We shall explore a script you can find at

http://jseden.dcs.warwick.ac.uk/construit/models/shopping/shop9sbr.js-e

Select all, copy, and go to

http://jseden.dcs.warwick.ac.uk/construit/

Paste the script into the Input window, submit, and follow guidance in lecture.

Computing with Imagination: New Directions for Educational Technology

Steve Russ

Computer Science, University of Warwick, UK

Acknowledgements

The work reflected here is made possible by the pioneering, continuing work of Meurig Beynon since the early 1980's. It could not have been accomplished without the huge contributions, still ongoing, from hundreds of enthusiastic Warwick students.

Empirical Modelling (1)

- A broad unconventional approach to computing developed by Meurig Beynon and co-workers at Warwick over the past 30 years.
- The most recent teaching of EM in module CS405 is linked from the main webpage at
  go.warwick.ac.uk/em
- A large range of resources is available from that page. The research group is still active and the CONSTRUIT! Project is based there.

Empirical Modelling (2)

- The approach involves a ‘stepping back’ from programming to a **modelling** perspective on a domain, based on observation + experiment
- The modelling is of a kind grounded in experience which sets EM on a course complementary to traditional CompSci
- It is kind of modelling natural to humans - based on observables, dependency, agency.

Why ‘step back’ at all?

It’s a pleasant place to be.

May not yet be ready to commit to the ‘action’, or functionality of a program. For example,

domain not yet well understood;

resources, and requirements, not known;

changes, so far unknown, are expected.

Empirical Modelling (3)

- Making hypotheses about an ODA framework for new phenomena has often required scientists and thinkers to use imagination.
- David Gooding adopted the term 'construal' for such imaginative work by Faraday on electromagnetism. EM adopts the same term for a personal, interpretive, understanding of something, as prelude to modelling. As a visual, interactive artefact it can also be 'made' or constructed on a computer.

Making Construals and Learning

- ‘Making construals’ refers to the co-construction by the modeller and on the computer of interactive artefacts which embody understanding.
- Construals have much to do with making sense of things and are thereby connected with learning of all sorts (or knowledge, skills and judgement for example).

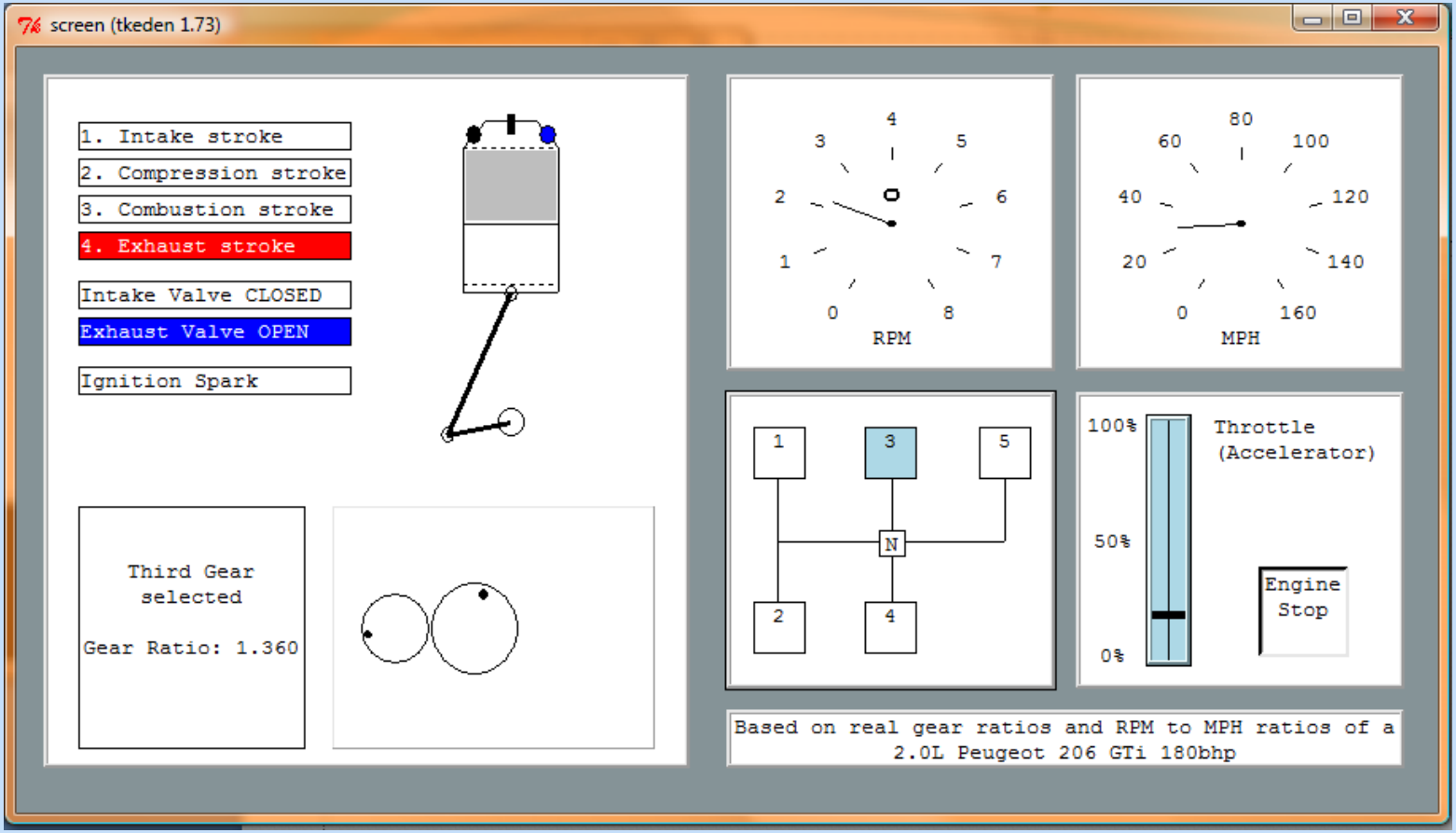
EM and Educational Technology

- See the EM webpage for more on this theme
- See PhD theses by Chris Roe and Antony Harfield for particular contributions
- See many of the publications for further material on links of EM and education
- Look for ‘constructionist learning’ and ‘constructivist computing’ or similar
- And participate in CONSTRUIT!

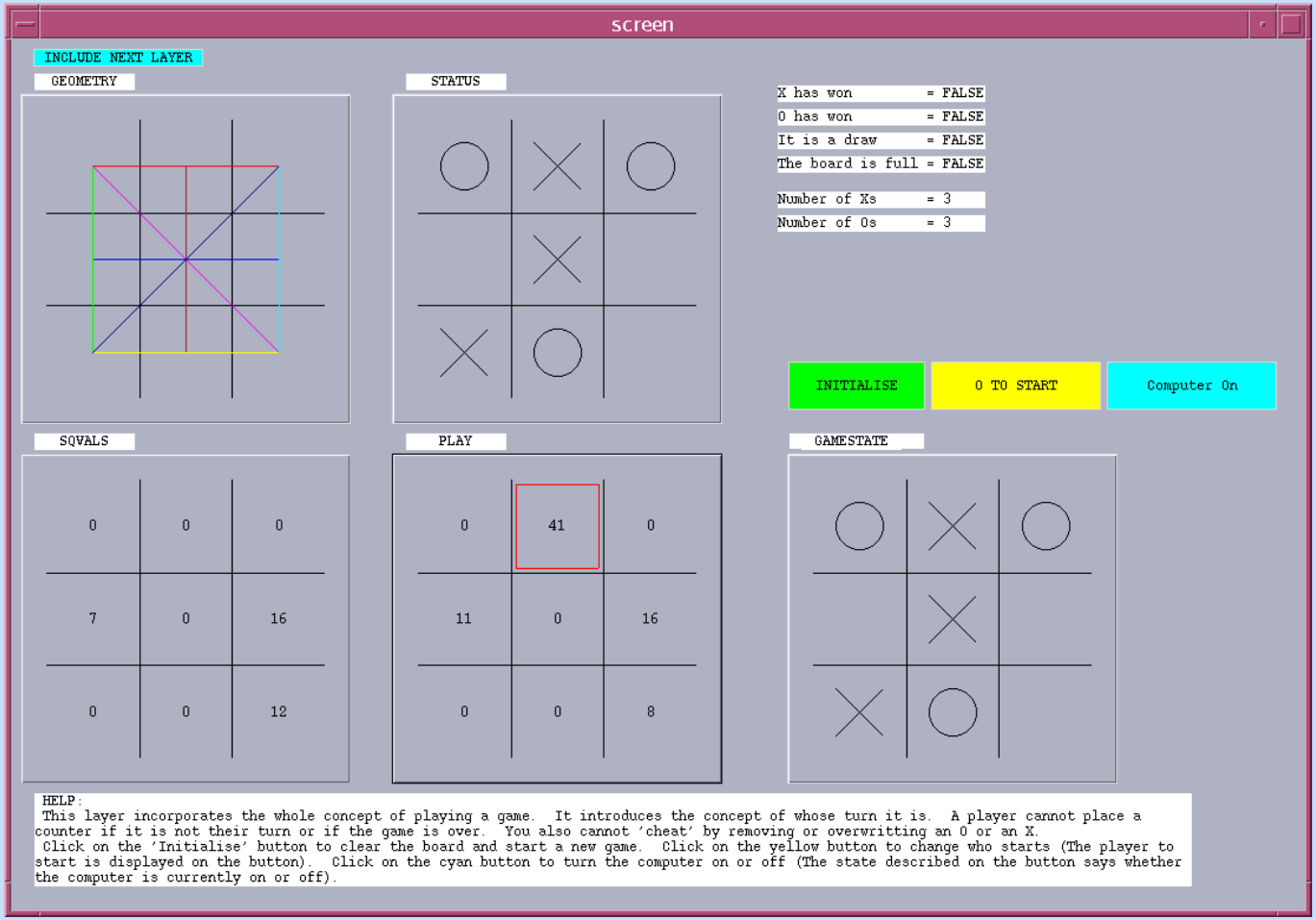
Some examples of construals

- A car engine [enginewithgearsSidbury2010]
- Playing noughts-and-crosses [oxoGardner1999]
- A room of your own [roomdemolabShao2012]
- A lattice of Boolean valued functions
- A planimeter from the early 19C
- An historic railway accident
- The Erlkoenig poem / song (Goethe / Schubert)

An engine with gears construal



Playing noughts-and-crosses



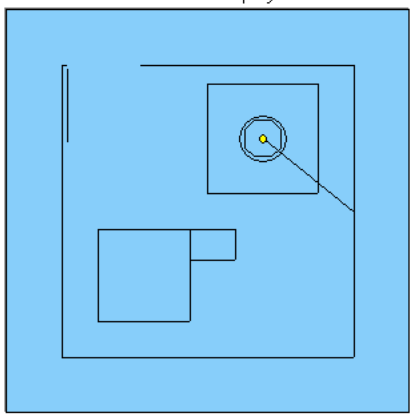
The “playing noughts-and-crosses” construal **oxoGardner1999** in the EM project archive

A construal of a room

EMPE

Presentation Environment

Interactive display:



Input Box:

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

Accept | %eden | %donald | %scout

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
```

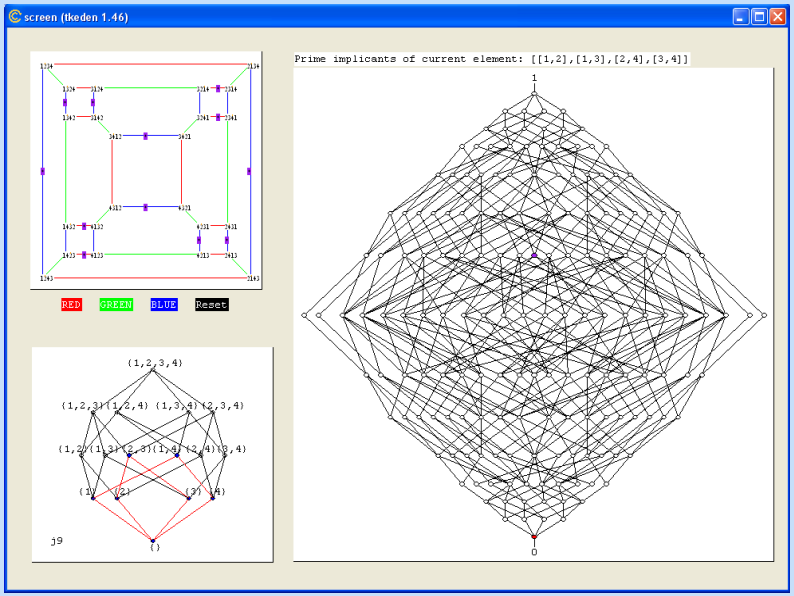
execute | copy to input box

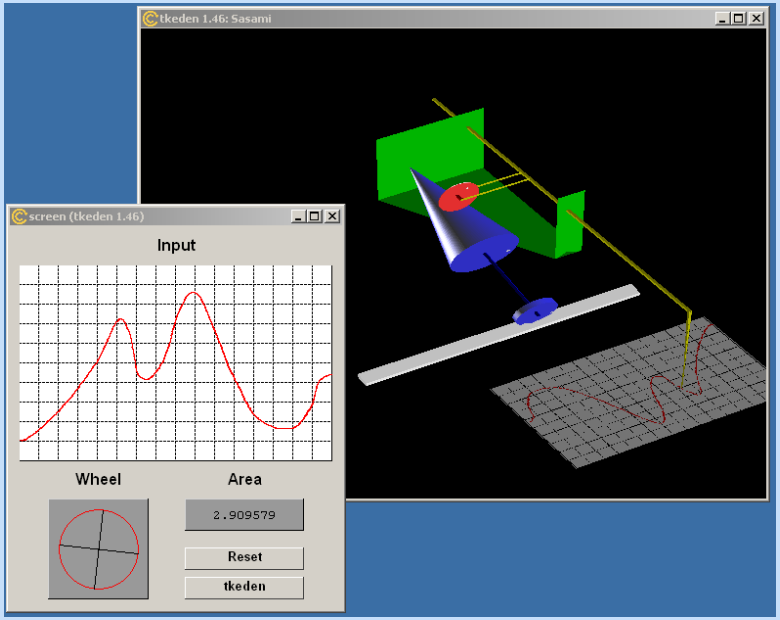
< Hide | Show tkeden | Copy Definitions | Quit

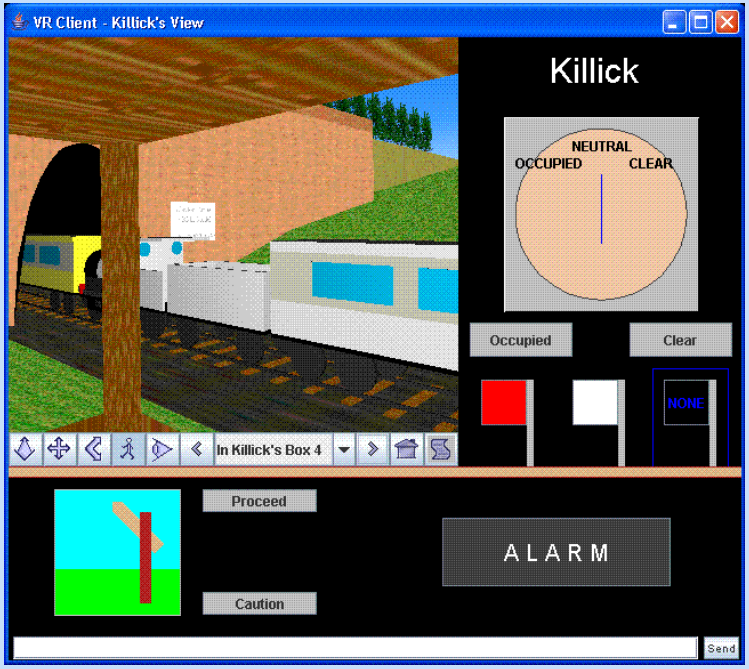
Slide 15 of 16

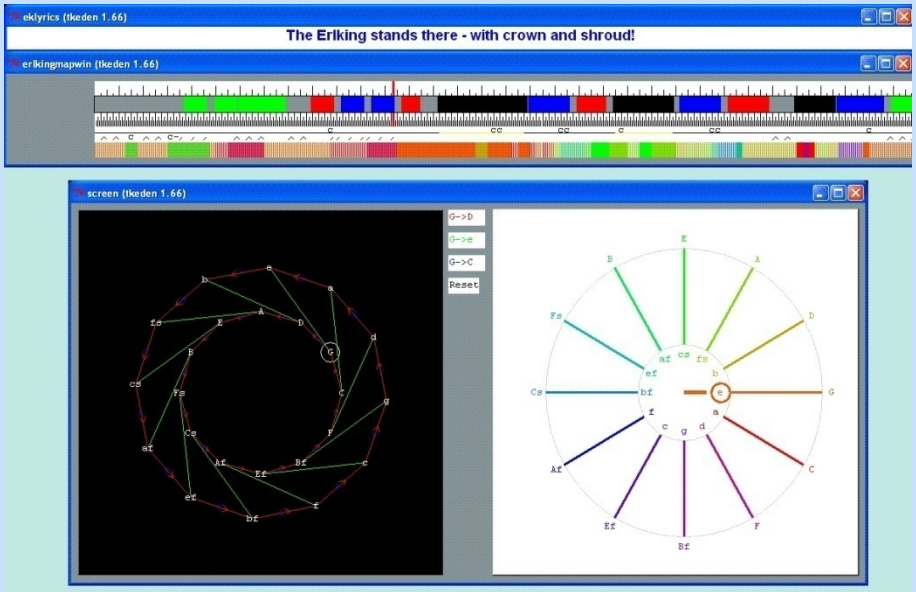
Edit slide | Add slide | <- Previous | Next ->

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









Making your first construal

Settle on a topic that interests you. Open a plain text editor. Study an existing example.

Write down the observables that occur to you. Write down some dependencies holding between them as ‘is’ statements; think how you might visualise the observables and what agencies might change them. Save, copy script.

Go to: http://jseden.dcs.warwick.ac.uk/construit/

Paste script into Input window, submit. That’s it!

Follow-up on your first construal

Possibly it did not go quite so easily!

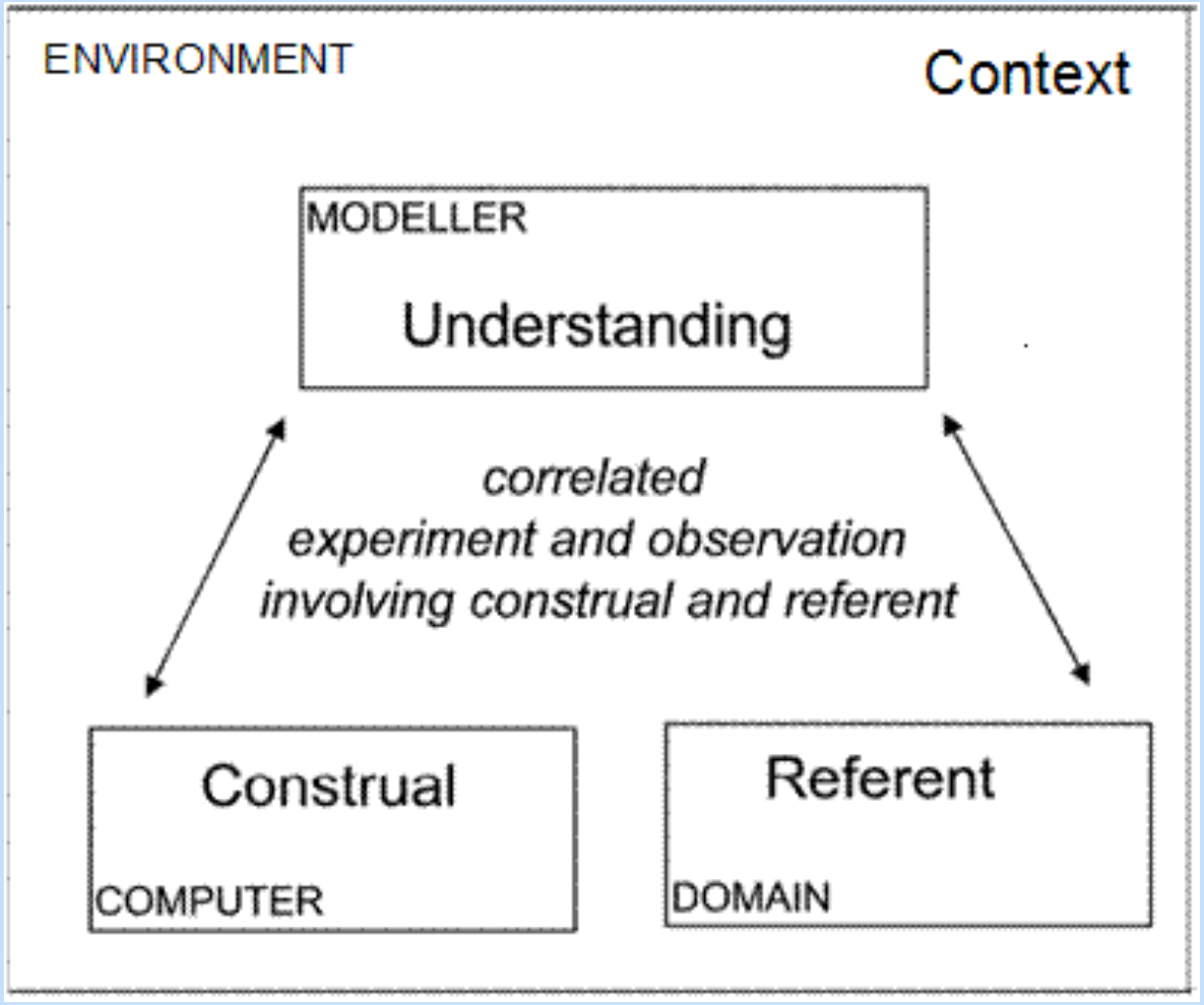
(If it did, please send me a copy of your script, you may win a prize!) Some things are hard to use at present - we are working to improve.

Let's look again at `shop9sbr.js-e`. One thing 'wrong' with it, is that the prices do not update [check this].

Check `item1text` and open Function List to examine `pricedisplay()`.

We'll do some editing to make the corrections.

The “Fundamental Diagram of EM”



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

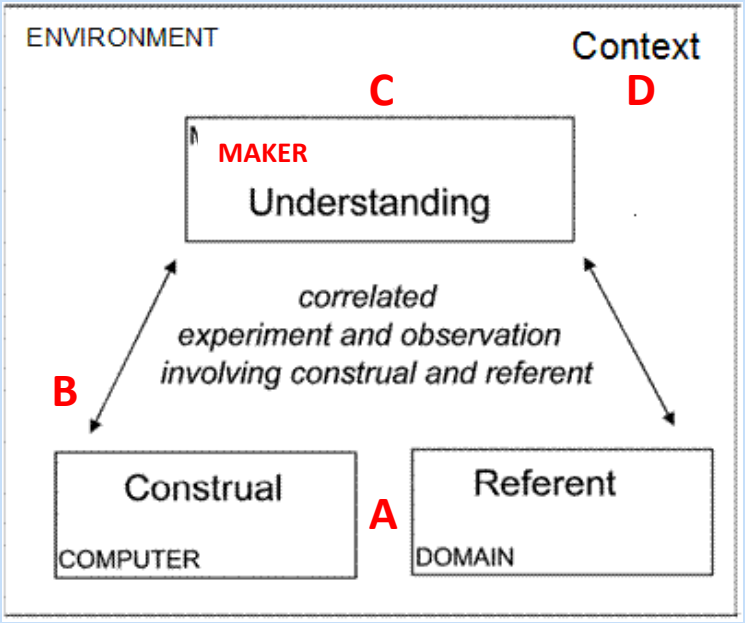
The “Fundamental Diagram of EM”

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

Key features of making a construal

- it is deliberate that the shopping construal is not ‘closed down’ (yet) to be program-like
- 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

CONSTRUIT! Title

Title: Making construals as a new digital skill for creating interactive open educational resources

Part of the Erasmus+ Programme under Key Action 2 Strategic Partnerships : Cooperation for innovation and the exchange of good practices [in IT] and 'across more than one field' [schools and HE].

Learning through making

The one main message of this presentation is that we hope to make a contribution (through CONSTRUIT!) to the resources available for learning:

Learning through making construals

For this we welcome and value your participation and contributions.

CONSTRUIT! Partners

Period: September 2014 – August 2017

Partners:

Warwick: Mike Joy, Meurig Beynon, Steve Russ +…

Edumotiva: (Patras) Dimitris Alimisis + …

UEF: (Joensuu) Erkki Sutenin + …

Edinburgh: Hamish MacCleod + …

Comenius: Peter Tomscanyi + …

Helix5: (Twente) Piet Kommers + …

CONSTRUIT! Outputs

Intellectual Outputs

O1: Open Online Course for principles of making construals, comprising

MCC: curriculum [next slide]

MCE: environment (instrument) [JS-Eden +]

MCM: materials [examples and applications]

O2: evaluation of the Course

O3: evaluation of the six claims for construals

Curriculum for making construals

Scope of the curriculum ("six claims"):

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

The Challenge of Accessibility

Our hopes of building on teachers' basic knowledge of programming – or capacity to 'pick it up' - to help in building construals was optimistic.

Our scripts are currently largely inaccessible to teachers without programming or maths skills.

We are exploring new ways to construct scripts.

Relevance of this for teaching programming ?

JS-Eden environment

Review of some important features you can explore in the environment

See the ‘views’ under the ‘New’ menu:

Dependency map

Script generator

Symbol List with Definitions, functions, actions

Symbol Lookup table

Support for state

Construal, Model, Program

Difference between:

Exploring something

Discovering a ‘law’ about it

Using the ‘law’ to solve problems

Construals	< - >	exploring
Models	< - >	making theories
Programs	< - >	problem solving

New horizons for shopping?

Whole idea of 'shopping' construals is to explore the imaginative world, stimulated by experiences of shopping. Not meant to be a *simulation* of 'real' shopping. See

http://jseden.dcs.warwick.ac.uk/construit/models/shopping/introShoppingConstrual.js-e

for some ideas on the range of games, variations, extensions, possible directions and programs to be derived from the construal.

Example extension

`enoughmoney.js-e`

A very simple ‘program’ that could be useful for children getting to know a new currency – recognising the coins and their values - and practising mental arithmetic. The program itself could easily have multiple variants offering variety, challenge and interest. It has been developed in many variants by Jonny Foss at Warwick Computer Science.

Like to join us in CONSTRUIT! ?

We welcome and value your participation in the work of CONSTRUIT! through:

- Making your own construals and variations …
- Giving feedback, and contributions, on the developing online course, experiments with it and the surrounding evaluations. We expect there to be publications and further research.
- Look out for participation soon at: http://www2.warwick.ac.uk/fac/sci/dcs/research/em/construit/

Thank you for your attention.

Next time you go shopping … remember CONSTRUIT! and make more construals!

A copy of these slides will soon be linked from the EM webpage.