Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:

Input Box:

Accept %eden %donald %scout

Dependency by definition in Imagine-d Logo: applications and implications

Chris Roe
The Centre for New Technologies Research in Education
Meurig Beynon
Computer Science
The University of Warwick, Coventry, UK

EuroLogo 2007
Bratislava

Acknowledgements

Dave Pratt and Antony Harfield

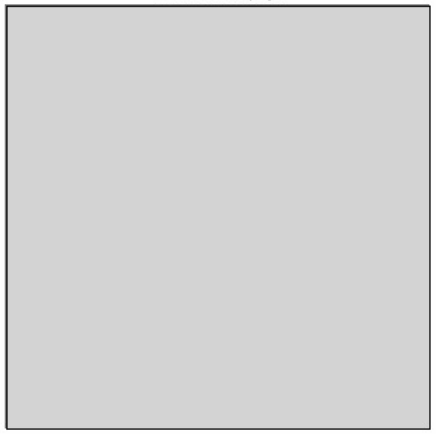
< Hide | Show tkeden | Copy Definitions | Quit

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

Dependency: Changing one value propagates change to other values in a predictable way as if in one and the same action.

The agenda raised by studying dependency:

- **Practical programming:** What advantages / drawbacks does introducing dependency have?
- **Pedagogy:** What is the significance of dependency in relation to learning?
- **Computing Science:** How can we make conceptual sense of "programming with dependency"?

These general questions relate in particular to Logo.

< Hide | Show tkeden | Copy Definitions | Quit

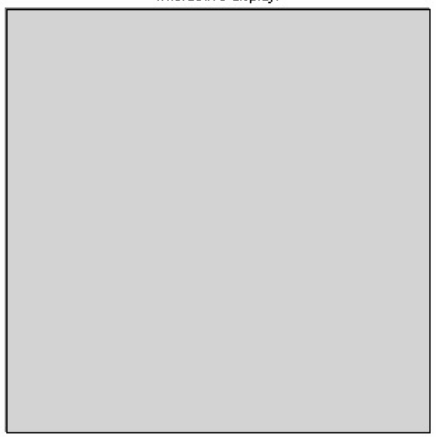
Slide 10 of 22

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept ○ %eden ○ %donald ○ %scout

Chris Roe's Imagine-d Logo prototype ...

... adds dependency to Imagine Logo, in such a way that:

- Programmer / model maker just writes the formula
  - *"dependency by definition"*
  - *how dependency is implemented not a concern*
- Programmer / model maker is able to **see** formula
  - *consider how the definitions of spreadsheet cells are inspected*
- Dependencies beyond go what a basic spreadsheet does, as when:
  - *presentation is dependent on values*
  - *formulae can use sliders*
  - *cells can be named variables and can be freely laid out*

< Hide | Show tkeden | Copy Definitions | Quit

Slide 11 of 22

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:

Input Box:

Accept %eden %donald %scout

Some related previous work ...

- **Peter Tomcsanyi , 2003**
  Implementing object dependencies in Imagine Logo
- **Sendov and Dicheva, 1988**
  Geomland as a 'mathematics laboratory'
- **Erich Neuwirth**
  Implementing spreadsheets in Logo

< Hide | Show tkeden | Copy Definitions | Quit

Slide 12 of 22

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:

Input Box:

Accept %eden %donald %scout

Tomcsanyi:

- dependencies implemented by the programmer through embellishing the underlying classes in Imagine Logo
- technically challenging, but very general in scope

... doesn't meet the end-user need as a spreadsheet does

... doesn't provide an easy way to inspect dependencies

< Hide | Show tkeden | Copy Definitions | Quit

Slide 13 of 22

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:

Input Box:

Accept %eden %donald %scout

Roe:

- attributes can be given values using *definitions* that can be specified and edited dynamically
- applies in particular to geometry, in some ways similar to Geomland
- more general, less specialised in scope: cf. *Visual Fractions* and *Cabri geometry*

< Hide | Show tkeden | Copy Definitions | Quit

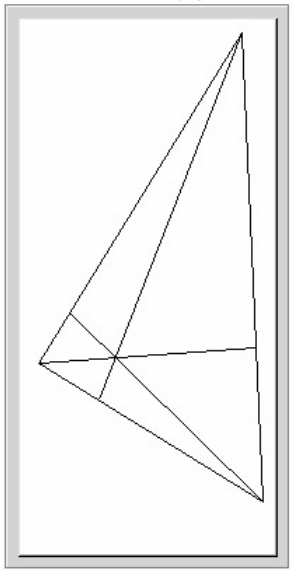
Slide 14 of 22

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

A simple illustrative example

Tomcsanyi implements

"a kind of canonical construction of dynamic geometry"

... a triangle together with the common point of intersection of the perpendiculars dropped from its vertices onto the opposite side

```
%eden
include("link.angel");
include("triangledisplay.s");
```

execute | copy to input box

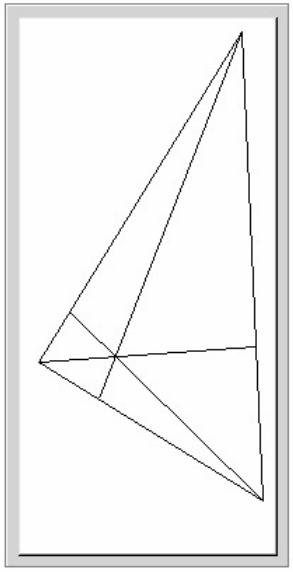
< Hide | Show tkeden | Copy Definitions | Quit

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

```
%donald
line AB, BC, AC
point A, B, C
AB=[A,B]
## the line joining points A and B
BC=[B,C]
AC=[C,A]
A = {20, 180}    ## the point p4
B = {260, 490}   ## the point p1
C = {285, 50}    ## the point p3
line perpA, perpB, perpC
perpC = perpend(C, AB)
##  perpendicular from C on to AB
perpB = perpend(B, AC)
perpA = perpend(A, BC)
point D
D = intersect(perpA, perpB)
```

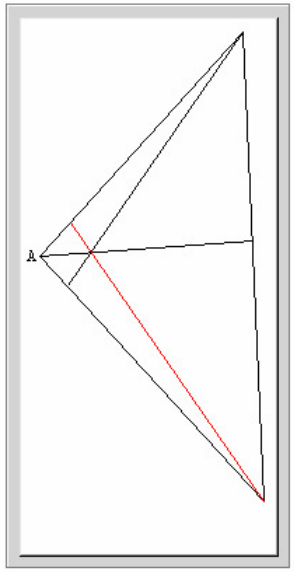
< Hide | Show tkeden | Copy Definitions | Quit

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

```
%donald
A = {20,280}
```

execute | copy to input box

```
%eden
A_perpC = "color=red";
```

execute | copy to input box

```
%donald
label LA
point px
LA = label("A", A-px)
px = {10,0}
```

execute | copy to input box

< Hide | Show tkeden | Copy Definitions | Quit

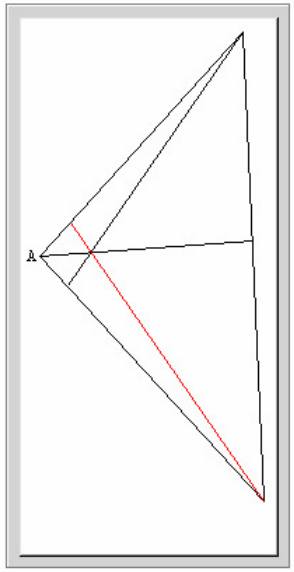
Slide 17 of 22

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

The agenda raised by studying dependency:

- **Practical programming:** What advantages / drawbacks does introducing dependency have?
- **Pedagogy:** What is the significance of dependency in relation to learning?
- **Computing Science:** How can we make conceptual sense of "programming with dependency"?

Central themes in the Empirical Modelling project
- see **http://www.dcs.warwick.ac.uk/modelling/**

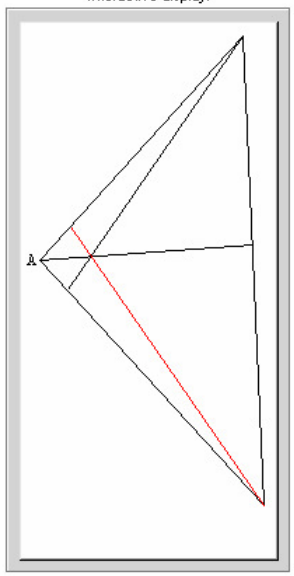
< Hide | Show tkeden | Copy Definitions | Quit

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

Practical programming: What advantages / drawbacks does introducing dependency have?

- *programming with dependency is powerful but can promote conceptual confusion*
- *ease-of-use doesn't mean "good vehicle for constructivist learning"*

Empirical Modelling demonstrates:

- an approach to programming that is essentially based on model building using definitions
- a methodology for modelling with dependency associated with a radical rethinking of computing

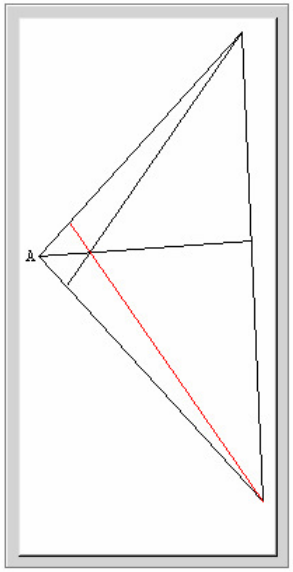
< Hide | Show tkeden | Copy Definitions | Quit

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

tkeden 1.66: Input

File Edit View Help

Accept %eden %donald %scout %sasami %aop %angel Interrupt

Enter EDEN statements:

```
  type: DONALD
  box: [{10, 10}, {190, 390}]
  pict: "TRIANGLE"
  border: 2
  bgcolor: "white"
  xmax: 300
  ymax:500
  sensitive: MOTION
};


%eden
_A is cart(20, trianglepic_mousePos);
*/
```

Practical programming:
introducing dependency ha

- *programming with depe*
  conceptual confusion
- *ease-of-use doesn't mean "good vehicle for constructivist learning"*

Empirical Modelling demonstrates:

- an approach to programming that is essentially based on model building using definitions
- a methodology for modelling with dependency associated with a radical rethinking of computing

< Hide | Hide tkeden | Copy Definitions | Quit

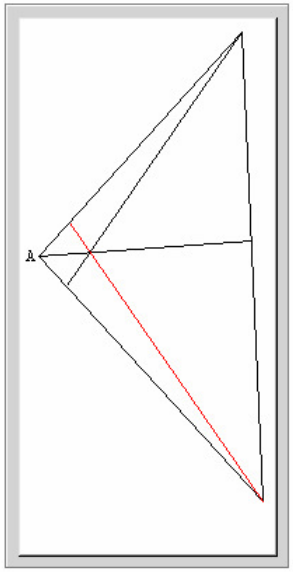
Slide 19 of 22

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

Pedagogy: What is the significance of dependency in relation to learning?

- *di Sessa: "make it experiential is perhaps the single most powerful educational heuristic I know"*
- *di Sessa: why should educational software come in large units so slick and complex they require man-years of effort from highly technically competent software engineers?*

Empirical Modelling exploits dependency:

- to give priority to expressing latent dependency in situations over describing processes and behaviours
- allowing pupil, teacher and developer roles to be synthesised in a homogenous model-building environment

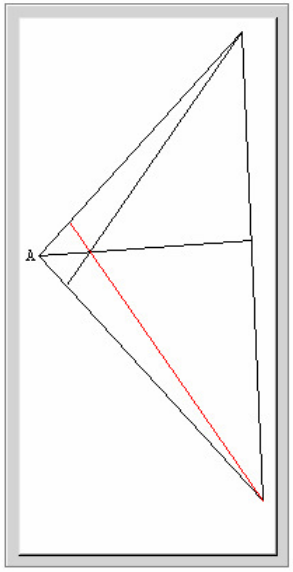
< Hide | Hide tkeden | Copy Definitions | Quit

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

Computing Science: How can we make conceptual sense of "programming with dependency"?

- *Large-scale software development involving radical rather than routine design is an exercise in constructivist learning*

Empirical Modelling activity:

- builds artefacts using principles resembling bricolage and the 'scientific method'
- supports primitive learning for familiarisation and understanding
- is in the spirit of di Sessa's *material intelligence*: building *construals* not programs

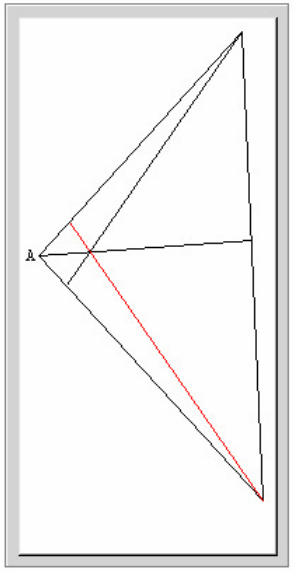
< Hide | Hide tkeden | Copy Definitions | Quit

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

Presentation Window

Empirical Modelling
Presentation Environment

Interactive display:



Input Box:

Accept %eden %donald %scout

Future directions?

- Introduce a dependency front-end to Imagine Logo, for use in a discretionary way - a *mixed-paradigm* approach
- enhancing Empirical Modelling tools by drawing on the excellent qualities of Logo and Boxer where accessibility is concerned

BUT ...

... Empirical Modelling has a Jamesian philosophical stance, for which Peter Naur proposes as a metaphor *an octopus jumping in a pile of rags*

For more background, see:

Meurig Beynon, Computing technology for learning - in need of a radical new conception, *Educational Technology & Society* 10(1), 94-106, 2007

< Hide | Hide tkeden | Copy Definitions | Quit

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