An Introduction to
CS405 "Introduction to Empirical Modelling"

Meurig Beynon
Reader Emeritus in CS

Theme for 2013-14

Towards a new conceptual framework for software development

Questionnaire for CS405

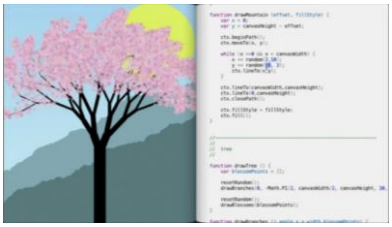
	LINUX	VB, C++, C, Java	Sorting Algorithms	Turing Machines	Functional Programming	JavaScript
Expert	1	5	3	1	2	1
Familiar	9	9	6	7	4	3
Some skill	2	1	5	3	3	6
No skill	3	0	1	4	6	5
	2.47	1.73	2.27	2.67	2.87	3.00

Orientation

To what extent have the problems that underlie the 'software crisis' been resolved? And can current principles and tools for software development meet the additional challenges that contemporary use of computers present?

The problems underlying the software crisis are unresolved. Current principles and tools for software development are inadequate to meet the challenges of contemporary use.

Bret Victor's *Learnable Programming*



Recall the *Inventing in Principle* video: editing of the program text on the right is directly correlated with changes to the picture on the left …

Some of the key ideas …

- Immediate feedback
- Having slider controls at hand to change values of variables interactively
- Having means to connect elements of the display with lines in the code and vice versa

Emphasising the quality of the experience that the programmer is getting … relevant not just to the novice, but to creative programmers

Current principles and tools for software development are inadequate to meet the challenges of contemporary use?

- Does Bret Victor's thinking address:
  - Learning to program?
  - s/w development?
- Deep-seated problems with 50 years history …
- What's changed?
  - The power of digital technology to create artefacts
  - Is this enough?

Bret Victor's stance …

Many questions raised – such as …

- What kind of application are we addressing?
- How to make the correlation of specified and intended behaviours understandable?
- How feasible is it to incorporate / elaborate / extend *existing* programs?
- Is this an approach viable beyond *1-person programs* in David Harel's sense?

Examining Victor's ideas wrt …

The role in software development for …

- Programming paradigms
- Formalism
- The development process
- Learning
- Concurrency
- Multi-agent development

Programming paradigms

Is every principled way to use the computer just a different style of programming a Turing machine? Is it possible to resolve the problems of software development by integrating different styles of programming, and why has this proved to be so difficult?

There is more to computing than programming Turing machines. Programming paradigms fail to account for how computing is experienced.

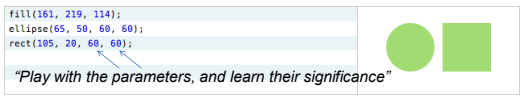
Bret Victor's stance … 1

Does it matter what programming paradigm you use in the 'Learnable Programming' context?

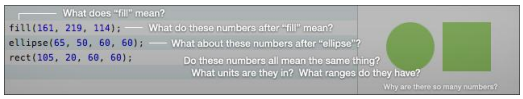
Yes – it even matters crucially how you write and present your procedural programs: consider how Victor reacts to the Khan Academy approach to teaching novice programmers

Bret Victor's Learnable Programming …

The Kahn Academy style of programming exercise:



Victor's initial critique:



Bret Victor's stance … 2

The goals of a programming system should be:

- to support and encourage powerful ways of thinking
- to enable programmers to see and understand the execution of their programs.

Bret Victor's stance … 3

The goals of a programming system should be:

- to support and encourage powerful ways of thinking
- to enable programmers to see and understand the execution of their programs

"A live-coding Processing environment addresses neither of these goals. JavaScript and Processing are poorly-designed languages that support weak ways of thinking, and ignore decades of learning about learning. And live coding, as a standalone feature, is worthless."

Bret Victor's stance … 4

- *How do we get people to understand programming?*
- We change programming. We turn it into something that's understandable by people.

A key question might be: is the 'something' we must turn it into still 'programming'?

Bret Victor's stance … 5

- *How do we get people to understand programming?*
- We change programming. We turn it into something that's understandable by people.

What Victor has in mind by *understanding* here is similar to what a musical performer might mean by "understanding a piece of music"

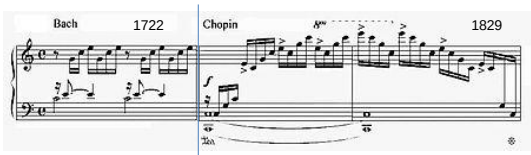
Bret Victor's stance … 6

"… most musicians don't compose entire melodies in their head and then write them down; instead, they noodle around on a instrument for a while, playing with patterns and reacting to what they hear, adjusting and sculpting .. An essential aspect of … a musical instrument is the immediacy with which the artist gets *something there* to react to."

Learnable Programming – Create by Reacting

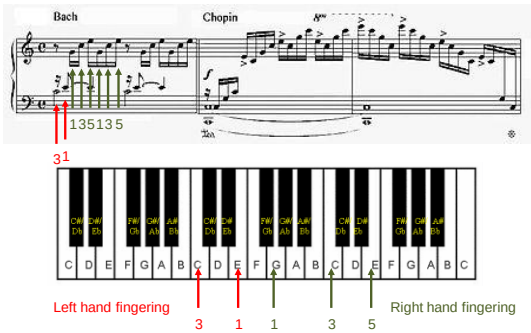
Connections in direct experience

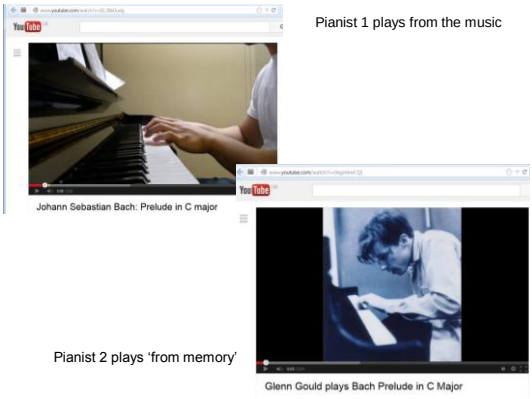
Playing the piano … two contrasting piano pieces:



Prelude in C major | Study in C major

Playing the Bach prelude …





Pianist 1 plays from the music

Pianist 2 plays 'from memory'

Two types of direct connection …

Pianist 1 plays Bach:

http://www.youtube.com/watch?v=IEL2IIkOudg

- This pianist is looking intently at the pattern of notes on the page and translating it into a pattern of movement of the fingers
- The pianist is experiencing a direct connection between precisely which note he is looking at and precisely which finger he is moving

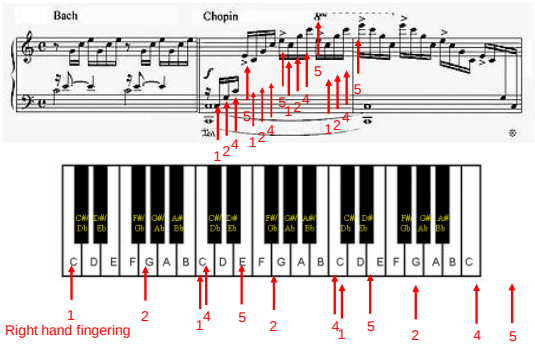
Two types of direct connection …

Pianist 2 plays Bach:

http://www.youtube.com/watch?v=0egJr6nvCQI

- This pianist is playing from memory - without music. He is looking intently at the keyboard and the pattern of movement of his fingers.
- The pianist is experiencing a direct connection between precisely which note he hears in his mind and precisely which finger he is moving.

Playing the Chopin study …



What's it like to play this … ?

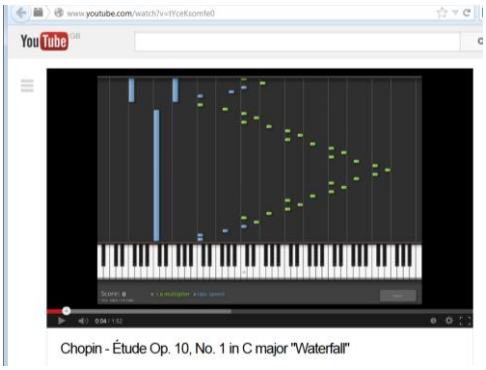
Can you play this the way Pianist 1 plays Bach?

http://www.youtube.com/watch?v=tYceKsomfe0

Does Ashkenazy play from memory?

http://www.youtube.com/watch?v=WpZr_cbYbXo

pongrr: I can play half of this. Now I only need to learn to play the right hand ;P



Matters arising …

- Does it help to teach the beginner the names of the notes on the piano?
- Can *anyone* play Chopin's Study in C and follow it on the music at the same time?
- We all make immediate connections just as amazing as any musician's in ordinary communication with language …

Central thesis of EM

- **Connections** between things we can experience independently, such as
  - JavaScript line of code / feature on the display
  - dot on the page / note on the keyboard
  - English words / things they signify

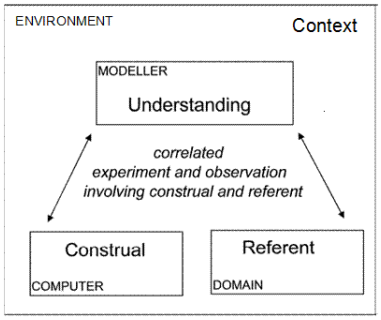
  can also be *given in experience*

(This idea is the basic tenet of William James's philosophy of Radical Empiricism, dating from more than 100 years ago.)

Central thesis of EM

- Such connections are of their nature highly personal and subjective, but can be the foundation for what appear to be (and can be treated as) objective relationships
- Connections between experiences can be engineered by establishing a correspondence between configurations of **observables**, **dependencies** and **agency**.

Empirical Modelling as *Construction*



What makes these connections?

- Doing programming exercises ...
- Language lessons and practice ...
- Piano practice ...

- in a suitably engineered context.

  e.g. put the music on the stand, sit at the keyboard, learn to look at the notes and not your hands etc

Formalism

What is the role and potential for formal specification and verification in software development? Is there any alternative basis on which the quality of software can be assured?

The role of formal methods cannot be dissociated from the identification of machine-like environments for computation. Quality assurance for software must in general rely on the practices observed in engineering.

Victor's *Kill Math* project

Victor attaches enormous importance to the kind of experiences that computers can offer

He believes that connections in direct experience can [to some degree? / totally?!] substitute for mathematics

In EM formalism is appropriate as a complement to - and itself an ingredient of - experience

Formalism in relation to EM

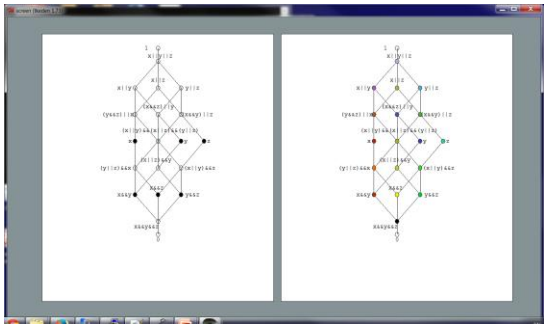
From an EM perspective, formalism is appropriate for specific kinds of experience – there are established objective contexts in which we can use logic for specification, programming languages, written texts, and musical scores, but it is essential in developing and understanding these to explore their experiential roots and counterparts.

Boolean functions with AND in OR

- Can view this symbolically as expressions, such as **x AND (y OR z)**
- Can also make a visual structure for the 18 logically distinct expressions of this form – as a ordered set where p <= q if **p implies q**
- If we assume that **x** is true, then other the proposition **x AND y** is logically equivalent to **y**

- we can demonstrate this visually.

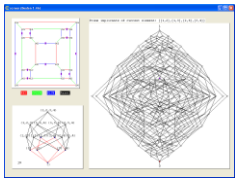
Boolean functions with AND in OR

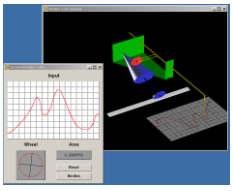


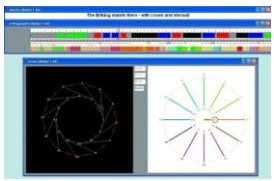
Boolean functions with AND in OR

- There are 166 different propositional functions in 4 variables – this is harder to visualise ...
- There are more than 7000 in 5 variables, and I think no-one has constructed a visualisation
- For most serious study of logical functions, symbolism seems unavoidable, and is itself a highly significant form of experience.

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







Learning

How well is current thinking about software development suited to the constructionist goal of establishing an intimate link between development and domain learning? What can be learned from parallel research in educational technology?

Procedural thinking within a computational framework is ill-suited to a constructionist stance, as is corroborated by problems encountered in developing effective educational technology.

Victor on constructionism ...

- Victor rightly associates making connections in experience with the notion of **constructionism**
- In *Learnable Programming*, he stresses the contribution of Papert's work (with Logo in particular)

EM is sceptical about procedural programming as a vehicle for constructionism ... Behaviours are experienced *moment-by-moment*

More references to Victor's work:

https://groups.google.com/forum/#!topic/qilang/1gt27APS_us

www.fullstack.io/choc/

http://lambda-the-ultimate.org/node/4607