File 20111027.1007: Notes from systems engineering webinar: ‘The Software Challenge of Robots’ by Terry Bollinger (IEEE Computer Society).

> ‘Software engineering is anticipating the future of code.’

Five code universes: think of these as nested circles in a Venn diagram.

- U1 (‘all code’) This is the universe of all executable strings.
- U2 (‘bad code’) Code that mostly works.
- U3 (‘slow code’) Code that always works, but slowly.
- U4 (‘fast code’) Code that always works and is fast, but overly optimised.
- U5 (‘adaptable code’) Code that is easily adapted.

Deep quality uncovers:

- Broader rules (algorithmic quality)
- Provably safe ways to compose
- Universal rules
- Knowledge of its own limits
- Good heuristics

> ‘Deep quality defines the Hilbert space axes.’

‘Architecture is the superhighways of the scale-free network. Architecture is the central sorting Post Office of the statistical multiplexing network that connects a huge number of potential problems to a huge number of potential solutions without the need for an $n \times m$ fully connected network in between.

Consider U5 (above) as Hilbert spaces. Proximity in higher dimensions. A well defined Hilbert space places related items close together; makes near neighbourhoods ‘safer’. An example of an axis in Hilbert space is whenever you define a constant in code.

‘Aspects’ in aspect-oriented programming are explicitly the axes in Hilbert space. I emailed the presenter to praise his explanation of this, and said I want to read the paper he’s going to publish on it some day, right?

References