Beyond Metaphor

In Search of Software Development's Identity

Kevlin Henney

kevlin@curbralan.com

Before

analogy *a comparison between one thing and another, typically for the purpose of explanation or clarification.*

- *a correspondence or partial similarity.*
- *a thing which is or is represented as being comparable to something else in significant respects.*
- *a process of arguing from similarity in known respects to similarity in other respects.*

metaphor *a figure of speech in which a word or phrase is applied to an object or action to which it is not literally applicable.*

- *a thing regarded as representative or symbolic of something else, especially something abstract.*

identity *the fact of being who or what a person or thing is.*

The New Oxford Dictionary of English

Analogies, Metaphors & Identity

- It is natural for any discipline to define itself by analogy with others
  - It is a human trait to use direct and oblique points of triangulation for communication and debate
- However, metaphor is not identity
  - The notion of *is like* is not the same as *is*
- Metaphor shear occurs when simplicity, utility and reality all part company

The Hardness of Software

- Software and code exist in an abstract space
  - The absence of a physical dimension makes it difficult to gauge
  - Any use of physical space for exposition or representation is by necessity metaphorical
- Software is digital and discontinuous, not analogue and continuous
  - Failure behaviour is more surprising
  - Degradation is less graceful

Software Engineering

engineering *the branch of science and technology concerned with the design, building, and use of engines, machines, and structures.*

The New Oxford Dictionary of English

Structural engineering is the science and art of designing and making, with economy and elegance, buildings, bridges, frameworks, and other similar structures so that they can safely resist the forces to which they may be subjected.

The Institution of Structural Engineers

The Nature of Disorder

- Each engineering domain is defined in terms of forces that must be resisted and balanced
  - Most of these forces are physical and are clearly grounded in physics
- What laws of physics govern code?
  - The second law of thermodynamics applied informationally: entropy — things get messier
- So just one law... and it is normally ignored
  - This may explain quite a lot!

Physical versus Informational

- Software development is empirical with theoretical underpinnings
  - It is mathematical, but it is not mathematics
- The domain of engineering can be broadly divided into informational and physical
  - Structural and chemical are found under physical engineering, software under informational
  - Therefore, software as engineering is a cousin not a sibling of the physical engineering disciplines

Software Architecture

Architecture is concerned with the creation of order out of chaos, a respect for organization, the manipulation of geometry, and the creation of a work in which aesthetics plays a far greater role than anything likely to be found in a humdrum building.

The Oxford Dictionary of Architecture

I call architecture frozen music.

Johann Wolfgang von Goethe

Architecture is the art of how to waste space.

Philip Johnson, New York Times

Masterplanning versus Growth

- There are two schools of thought on what software as architecture should be
  - Each related to building-architecture metaphors
- Masterplan architecture focuses on software as blueprint based
  - Formal, fixed, abstract and established up-front
- Piecemeal growth rejects this view, focusing on informed change over time
  - Evolutionary, empirical and concrete

Architecture and Marketecture

- The built environment offers an attractive metaphor for software
  - Aspects of scale, form and function
- However, the work of 'real' architects is often misunderstood
  - Although there are parallels to be drawn and lessons to be learnt, false analogies are often drawn
- Architecture addresses an aspect of software development, but not its whole

Software Craft

Craft and craft-related activities are pleasurable, and the pleasure derives from doing something well that you know well how to do. The criteria for doing the job well and judging whether the job has been well done are known in advance. The best craft jobs are those where the mind and the body are engaged entirely as one in the process.

Peter Dormer, Design After Modernism

People in high tech take pride in their work. They are individuals who see the details of the things they produce in the light of the trials and triumphs they experience while creating products. In the courage of creation, they find a place to hang their individuality. Programmers and techno types appreciate elegant, spare code and the occasional well-turned architectural hack.

Rick Levine, The Cluetrain Manifesto

Love's Labours Lost?

- Craft is not mutually exclusive with either art or engineering
  - Emphasis is on skill and creativity for purpose
  - Focused on individuals and small teams
  - Learning from failure plays an important role, as does learning from success
- Software's relationship with craft comes from the skill and role of the individual
  - Not because of the maturity of the discipline

Craft versus Industry

- Falsifiability, testing and so on distinguish software development from being a pure craft
  - This is an engineering trait
- And some of the notions of craft are also false
  - There are a number of idealistic fairy tales associated with the positive image of craft
- However, the often cited tension between craft and industry is often a false one
  - Industrialisation has already occurred

Software Composition

composition *a work of music, literature, or art.*
- *the action or art of producing such a work.*

The New Oxford Dictionary of English

Substance doesn't change. Method contains no permanence. Substance relates to the form of the atom. Method relates to what the atom does. In technical composition a similar distinction exists between physical description and functional description. A complex assembly is best described first in terms of its substances: its subassemblies and parts. Then, next, it is described in terms of its methods: its functions as they occur in sequence. If you confuse physical and functional description, substance and method, you get all tangled up and so does the reader.

Robert Pirsig, Zen and the Art of Motorcycle Maintenance

Writing and Music

- Style has a strong role to play in all forms of creative composition
- Software development has very close parallels with technical writing
  - But is necessarily more precise
  - And it suffers similar variance in quality
- Music and its performance have much in common with software and its development
  - Informational with a variety of performance styles

Differentiation

- Software development is like all these forms of composition... but it is not one of them
  - Art, whether free or functional, is judged by very different criteria
- Subjectivity has a supporting rather than a primary role in software development
  - Yes, aesthetics matter — more than is currently recognised — but there is a lot more to software development than aesthetics

Beyond

The ability to quote is a serviceable substitute for wit.

W Somerset Maugham

Although lots of money has been spent over the years trying to get people out of the software development process, none of the attempts has really worked. The old dreams of natural-language specifications being read by computers that then automagically create your software are just that – dreams. The only way currently to make this process work is to give the computer an extremely detailed specification in a formal language – an activity we all know as "programming a computer."

Pete McBreen, Software Craftsmanship

Are We Nearly There Yet?

- It is a design-based discipline in its own space
  - And this space connects, by metaphor and application, to other spaces
- To engineer is human
  - If you accept the separation of informational from physical engineering, then this is where it lives, otherwise you have just software development
  - Maturity (or lack of it) is only a small part of what separates it from others: rest is its inner nature