Carlo Pescio

A Physics of Software

an investigation into the true nature of software

Version 0.003

Version 0.003

- Just a Scrapbook
  - Slightly more structured than blog posts
  - Consolidate ideas
  - Provide hooks for further development / investigation
    - Hooks are in red
  - Pointers to relevant blog posts
    - using a callout hyperlink

Why a “Physics”?

- Physics (wikipedia)
  - Physics (Ancient Greek: φύσις physis "nature") is a natural science that involves the study of matter and its motion through space-time, as well as all applicable concepts, such as energy and force. More broadly, it is the general analysis of nature, conducted in order to understand how the universe behaves
- Matter, Motion, Energy, Force?
  - Not really – but software has a nature
  - Properties & Forces – largely undefined

What is Software?

- Software is Encoded Knowledge (Information)
  - Knowledge about processes
  - Knowledge about data

- For a Designer, Software is a Material to be shaped
  - The concept of Form

The Nature of Software

- Some influence from Alexander is unavoidable :)

- Not an Alexandrian clone, anyway
  - There is some overlap, of which I'm happy
  - Surely some inspiring concepts (e.g. the Center)
  - Mostly original work on forces and properties
  - Looking for a few pervasive, unifying concepts

- What is a Center in Software, anyway?

  - A locus of highly cohesive information

What is Software Design?

- Design is an act of partitioning information
  - You can't have cohesive units if you can't keep units apart from each other
- Information is naturally subject to several forces
  - The forcefield
- The Designer can modify the forcefield
  - Decisions
  - The Decision Space

The Dual Nature

- Software Design seeks balance in Two Worlds
  - Artifacts
    - Form
  - Run-Time Instances
    - Function
- In both worlds, we have almost-fractal hierarchies of concepts
- Concepts depends on technology (languages/hw)

Properties & Forces

- Mass / Gravity / Inertia
  - Defined on the Artifacts side
  - Must be extended to the Run-Time side
- Distance
- More already identified (not yet discussed)
- The “right” forces / properties apply to all levels of the fractal hierarchy, ideally in both worlds.
  - Would be nice to extend some concepts in the decision space as well. Distance, mass, inertia...

Distance – the (fractal) Space

- Artifact side 
  - Executable Statement / Unstructured Variable
  - Group of Statements / Group of Variables
  - Function-Procedure / Data Type / Table
    - Named Concept (Strong Centers)
  - Nested Function / Local Variable-Parameter
    - Boundaries
  - Module-Class / Database
    - Collapsed Concepts (gravity, Not-Separateness)
  - Component-Service
    - Range of Sizes
  - Application

Distance – the (fractal) Space

- Run-Time Side
  - CPU Execution Pipeline / Register
    - Instruction scheduling
  - L1 cache (same cache line / different lines)
    - Non shared: Locality
  - L_n cache (same cache line / different lines)
    - Shared: Coherence
  - Main Memory
    - Protection
  - Virtual Memory (same page / different pages)
    - Latency
  - Out-of-process services / data
    - Connection, Format
  - Remote services / data (LAN)
  - Remote services / data (WAN)

More

- My Blog
  - http://www.carlopescio.com
- Notes On Software Design
  - http://www.carlopescio.com/search/label/NOSD
- Previous posts on The Concept of Form
  - Look around in my blog; start from:
  - http://www.carlopescio.com/2008_09_01_archive.html
  - ... and go back in time :)