Building Robust Systems

A Whirlwind Tour by Neil Menne

Evolvable Systems

The Sussman Way

- → Degenerate
- → Modularized
- → Regenerative

Building Immoral Systems

Building Flexible Systems

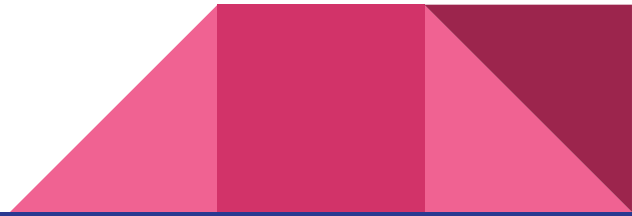
Degeneracy

Stealing lessons from biology

degeneracy occurs when structurally dissimilar components/modules/pathways can perform similar functions (i.e. are effectively interchangeable) under certain conditions, but perform distinct functions in other conditions.

Degeneracy in Practice

- Partition your work!
  - Services can/should make progress when partial failures occur
  - Services can effectively replace other services
- Abstraction
  - Reusable components are the “cells” that make a good service tick
  - Reusable components can be harder to write
- Continuous progress
  - Failure doesn’t always mean failure
- Enabling Exploratory Behavior
  - Gotta have that REPL
  - Need clear API Boundaries



Modularization

From Small Beginnings

- ➔ Establish boundaries for processes
- ➔ Establish an environment specific to a process
- ➔ Establish known locations for where processes are

Modularization in Practice

- Subsystem boundaries
  - Isolate failure
  - Highlight speedup/scaling possibilities
  - Allow for subsystem-specific scaling
- Modular code
  - Subsystems are just specialized “machinery”
  - Mechanisms are testable separate from business logic

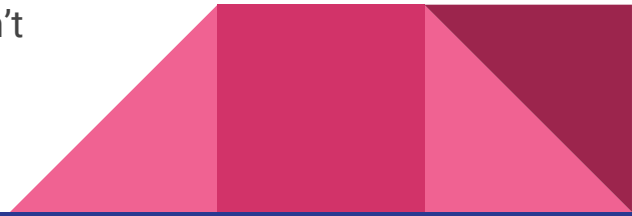
Regeneration

Recovery and Defense

“Biological systems have substantial abilities to repair damage, and, in some cases, to regenerate lost parts. This ability requires extensive and continuous self-monitoring, to notice the occurrence of damage and initiate a repair process. It requires the ability to mobilize resources for repair and it requires the information about how the repair is to be effected.”

Regeneration in Practice

- Identifying Failure
  - Latency is a kind of failure
  - How do you identify failure?
- Recovering from Failure
  - How/when do you alert the engineer?
  - Designing for recovery
- Diverse systems
  - Just because you've built something similar in the past doesn't mean you should build it the same way *NOW*



Questions and Comments

Neil Menne
@the1evilgenius