Design, Again

Invention vs. Architecture

Factoring Policy from Mechanism

Exercise

Invention vs. Architecture

- **Architectural design concerns organization**
  - **Dividing the solution into the "right" pieces**
  - **Placing prices on design commitments**
    - **What can be changed, at what cost**
- **Architecture complements invention**
  - **Not an alternative to creative solutions**
  - **May also be creative**

Heuristics for Invention

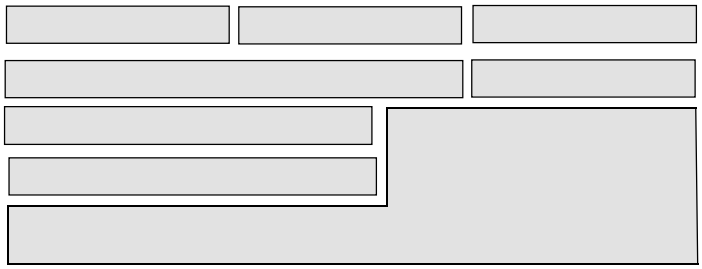
- Explore the problem space first
  - Understand the problem thoroughly before proposing solutions
- Generate solutions before filtering them
  - Bad solutions are raw materials for good
- Enumerate constraints before applying them
- Postpone unnecessary design commitments

Chicken or Egg?

- Some invention must precede architecture
  - Architectures describe solutions, not problems
- Some architecture must precede invention
  - Big problems must be sub-divided
- So ... there will be iteration and interplay
  - Invention + Factoring at several layers
  - Revising each as the other becomes clearer

Layered Systems

- An approach to architectural design
  - developed primarily in OS & networking
  - "virtual machine" abstractions



Mechanism vs. Policy

- A particular way to layer virtual machines
- Mechanism: simple, application-independent layer of functionality
- Policy: an application-specific use of mechanism

Virtual Memory Paging

Mechanism vs. Policy

- Mechanism: Page in, page out
- Policy: Page replacement policy
- Page replacement policy can be changed without altering paging mechanism
- Paging mechanism can be changed (e.g., page caching) without altering replacement policy

Knowledge based systems

Mechanism vs. Policy

- Mechanism
  - "Inference engine" is a domain-independent mechanism for selecting and executing rules
- Policy
  - A particular knowledge-based system combines a highly application-specific "knowledge base" (collection of rules and facts) with the inference engine.
- In this case, mechanism is a "virtual machine"

Little Languages for Policy

- Quake/Zork/et al. engines vs. dungeons
- Postscript
- Visio symbol/function sets
- Mail filtering patterns
- Unix termcap/terminfo
- Spreadsheets
- SGML & XML document definitions
- ...

Exercise: Collaborative Spam Filter

- Task: Allow group members to determine whether each incoming email message was received by others in the group, *without revealing message contents*.