ELEC 875
Design Recovery and Automated Evolution

Architecture Analysis
Next Week


Some Systems do not have documented system architecture

- Extract the system Architecture

Problems keeping documented architecture up to date

- Automation?
**Linux**

- 10 KLOC in 1991 to 1.5MLOC in 1998
  ◊ doubled every year

- Linux Kernel - 800 KLOC
  ◊ documented at individual system level
  ◊ no architectural documentation
  ◊ good guinea pig
Architecture

• Conceptual architecture
  ◊ How developers think about the system
  ◊ only the meaningful links and dependencies
  ◊ Component responsibilities
  ◊ Component interactions

• Activities
  ◊ Capture Functionality
  ◊ Capture Properties
  ◊ Constraints
Architecture

- Concrete Architecture
  - The “real” architecture
  - extracted by some set of tools.
  - Contains extra links required by the implementation

- Neither architecture is documented for Linux
Architecture Change

- Architecture Erosion
  - Conceptual Violations

- Architecture Drift
  - Concrete architecture drifts away from conceptual architecture
Linux Conceptual Architecture

- Read Documentation
  ◇ No conceptual architecture documentation?
  ◇ Some architectural information spread in different documents
  ◇ Some overview documentation
  ◇ Knowledge of other Unix based architectures
    - Tunis, Hector
Linux Conceptual Architecture

- File System
- Network
- Scheduler
- Memory Manager
- IPC
- Init
- Library
File Conceptual Architecture

Figure 2: File System Conceptual Architecture
Linux Concrete Architecture

• Group source files based on directory structure, naming conventions, source code comments and source code examination
• Extract Relations between source files
• Lift relations between source files to relations between subsystems
• Convert to concrete architecture
Linux Extraction

• cfx - predecessor to cppx
  ◊ function level extraction (Middle Model)
• grok used to turn:
  - relations between functions and functions
  - relations between functions and files
  >>>>>>
  - relations between files
• Files manually assigned to subsystems
• grok used to *lift* relations from files to subsystems
Linux Concrete Architecture
Figure 4: Partial Subsystem Hierarchy
File Concrete Architecture

Figure 6: File System Concrete Architecture
MM Concrete Architecture

Figure 5: Memory Manager Dependencies
Linux Concrete Architecture

- Given that concrete architecture one would think that the Linux implementers are horrible coders
- efficiency shortcuts
- expediency
- Debugging (process scheduler depends on file system) mislocation of printk (process -> library)
- Synchronization primitives in IPC
- differences at the subsystem level
What Did We Learn?

- Human Assistance Needed in Analysis
- Concrete Architecture Different from Conceptual
  - why
    - conceptual architecture incorrect
    - efficiency
    - expediency
    - unanticipated dependencies
    - differences in control flow
    - implementation language or environment constraints
    - some subsystems implemented everywhere!!