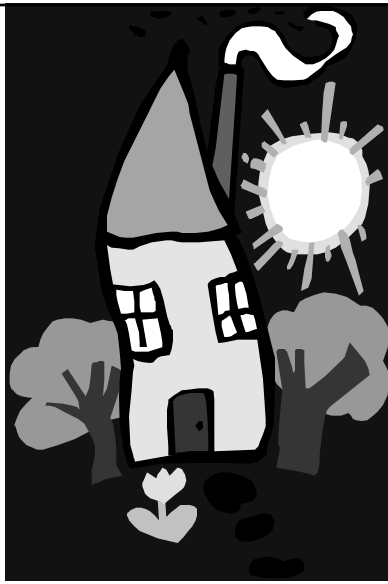


House Of Structure

By Warren Keuffel



Warren Keuffel, "House of Structure," *Unix Review* 9(2), 28-36

House of Structure



- ◆ What is article's purpose?
 - Examine 25 year history of structured systems thinking
 - Show object methods in context of a logical evolution of techniques

Warren Keuffel, "House of Structure," *Unix Review* 9(2), 28-36

Early Methods 1950s and Early 1960s: Unsystematic



- ◆ The process of Systems Analysis was not well understood.
- ◆ The problems were poorly understood.
- ◆ Focus was often on solutions.
- ◆ Difference between efficiency and effectiveness was not well understood
- ◆ Efficiency is doing the job right.
- ◆ Effectiveness is doing the right job!!
- ◆ Before we begin to solve a problem we must clearly understand and define it. This is what analysis is all about, but this was not generally recognized until the late 1960s. . .

Warren Keuffel, "House of Structure," *Unix Review*9(2),28-36

The evolution of structure:



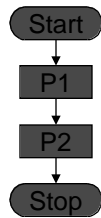
- ◆ 1965 "Programming considered as a human activity" (by Dijkstra) introduced:
 - source of lack of clarity (self modifying code)
 - GOTO problems
 - principle of non-interference - functional decomposition

Warren Keuffel, "House of Structure," *Unix Review*9(2),28-36

The evolution of structure



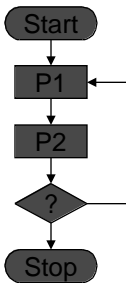
- ◆ 1966 “Flow Diagrams, Turning Machines and Languages with only two formation rules” (by Bohm & Jacopini) proved that any flowchart can be constructed out of the standard three, one-entry/one-exit constructs
 - Sequence



The evolution of structure



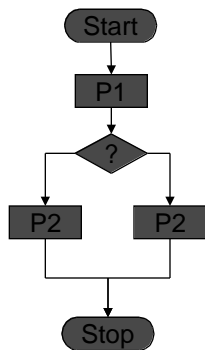
- ◆ 1966 “Flow Diagrams, Turning Machines and Languages with only two formation rules” (by Bohm & Jacopini) proved that any flowchart can be constructed out of the standard three, one-entry/one-exit constructs
 - Iteration



The evolution of structure



- ◆ 1966 “Flow Diagrams, Turning Machines and Languages with only two formation rules” (by Bohm & Jacopini) proved that any flowchart can be constructed out of the standard three, one-entry/one-exit constructs
 - Selection



The evolution of structure



- ◆ 1968 Dijkstra's letter
 - efficiency vs. maintainability and readability
 - who won? (maintainability and readability)

The evolution of structure



- ◆ 1968 NATO introduced the terms
 - *software engineering*
 - *software crisis*

The evolution of structure



- ◆ 1972 "A technique for software module specification with examples" (by Parnas)
 - introduced Pseudocode
 - determined a major problem is lack of techniques for precise specifying without revealing too much information

The evolution of structure



- ◆ 1972 “A technique for software module specification with examples” (by Parnas)
- ◆ first suggestions - desirable to model a program prior to programming
 1. specification must provide all of information needed to use program correctly and nothing more (WHY?)
 2. provide implementor all information required to complete it and nothing more - in particular nothing about the calling program
 3. specification must be sufficiently formal to be machine testable for consistency, completeness, and other desirable properties

The evolution of structure



- ◆ 1974 “Structure Design” by Stevens, Myers, and Constantine
 - graphical notation describing behavior of modules
 - a taxonomy of module types
 - first steps toward a methods for analyzing positions
 - top down decomposition
 - coupling - high coupling=difficult to maintain
 - cohesion - measure of how well module accomplishes a one single task
 - data flow analysis

The evolution of structure



- ◆ 1977 “Structured Analysis for Requirement Definition” by Ross and Schoman
 - SADT (proprietary)
 - First structured analysis
 - First commercial methodology
 - No data dictionary
 - No system behavior description at module level

The evolution of structure



- ◆ 1978 “Structured Analysis and System Specification” DeMarco
 - not proprietary
 - well written
 - lots of publicity for Yourdon
 - data dictionary
 - minispecs = pseudocode

The evolution of structure



- ◆ 1986 “A Rational Design process and how to fake it” Parnas
 - big systems cannot be designed in a top down fashion
 - documentation must be produced as if top down had been done

The evolution of structure



- ◆ 1988 “understanding and controlling software costs” Boehm
 - killed the SDLC
 - advocated iteration between analysis and design
 - concentrate efforts in area of greatest risk

The evolution of structure



- ◆ 1988 “Modern Structured Analysis”
Yourdon
 - introduced object orientation to structured analysis

History Time Line



- ◆ 1989 Coad introduces Object Oriented Analysis