

Literate Programming

CSE 308: SOFTWARE ENGINEERING
SUNY AT STONY BROOK

A Blast From The Past

“THERE IS A GAME SOMETIMES PLAYED WITH ALGOL 60 PROGRAMS—REWRITING THEM SO AS TO AVOID USING `goto` STATEMENTS. IT IS PART OF A MORE EMBRACING GAME—REDUCING THE EXTENT TO WHICH THE PROGRAM CONVEYS ITS INFORMATION BY EXPLICIT SEQUENCING... THE GAME’S SIGNIFICANCE LIES IN THAT IT FREQUENTLY PRODUCES A MORE “TRANSPARENT” PROGRAM—EASIER TO UNDERSTAND, DEBUG, MODIFY, AND INCORPORATE INTO A LARGER PROGRAM.”

— PETER LANDIN, 1965

The Evolution of Programming

- ☒ PHASE 1: “JUST GET IT DONE”
 - ☒ THIS OFTEN LED TO LOTS OF “SPAGHETTI CODE”
- ☒ PHASE 2: STRUCTURED PROGRAMMING
 - ☒ AT MINIMUM, THIS MEANT THAT EVERY FUNCTION HAS EXACTLY ONE WAY OUT (RETURN STATEMENT)
- ☒ PHASE 3: OBJECT-ORIENTED PROGRAMMING
 - ☒ PHASE 3.5: REFACTORING, DESIGN PATTERNS

The Evolution of Programming

- ☐ PHASE 1: “JUST GET IT DONE”
 - ☐ THIS OFTEN LED TO LOTS OF “SPAGHETTI CODE”
- ☐ PHASE 2: STRUCTURED PROGRAMMING
 - ☐ AT MINIMUM, THIS MEANT THAT EVERY FUNCTION HAS EXACTLY ONE WAY OUT (RETURN STATEMENT)
- ☐ PHASE 2.5(?): LITERATE PROGRAMMING
- ☐ PHASE 3: OBJECT-ORIENTED PROGRAMMING
 - ☐ PHASE 3.5: REFACTORING, DESIGN PATTERNS

Knuth's Philosophy

- ☐ IF WE WANT TO PRODUCE BETTER-DOCUMENTED PROGRAMS, WE SHOULD CONSIDER THEM TO BE “WORKS OF LITERATURE”
- ☐ THE OLD ATTITUDE
 - ☐ MAIN TASK: INSTRUCT A COMPUTER WHAT TO DO
- ☐ THE NEW ATTITUDE
 - ☐ MAIN TASK: EXPLAIN TO HUMAN BEINGS WHAT WE WANT A COMPUTER TO DO

In Knuth's Words...

“THE PRACTITIONER OF LITERATE PROGRAMMING CAN BE REGARDED AS AN ESSAYIST, WHOSE MAIN CONCERN IS WITH EXPOSITION AND EXCELLENCE OF STYLE. SUCH AN AUTHOR...CHOOSES THE NAMES OF VARIABLES CAREFULLY AND EXPLAINS WHAT EACH VARIABLE MEANS. HE OR SHE STRIVES FOR A PROGRAM THAT IS COMPREHENSIBLE BECAUSE ITS CONCEPTS HAVE BEEN INTRODUCED IN AN ORDER THAT IS BEST FOR HUMAN UNDERSTANDING...”

“Oh what a tangled web we weave...”

- ▣ KNUTH: “[A] COMPLEX PIECE OF SOFTWARE IS, INDEED, BEST REGARDED AS A WEB THAT HAS BEEN DELICATELY PIECED TOGETHER FROM SIMPLE MATERIALS. WE UNDERSTAND A COMPLICATED SYSTEM BY UNDERSTANDING ITS SIMPLE PARTS AND THEIR IMMEDIATE NEIGHBORS. IF WE EXPRESS A PROGRAM AS A WEB OF IDEAS, WE CAN EMPHASIZE ITS STRUCTURAL PROPERTIES IN A NATURAL AND SATISFYING WAY”

Top-Down vs. Bottom-Up

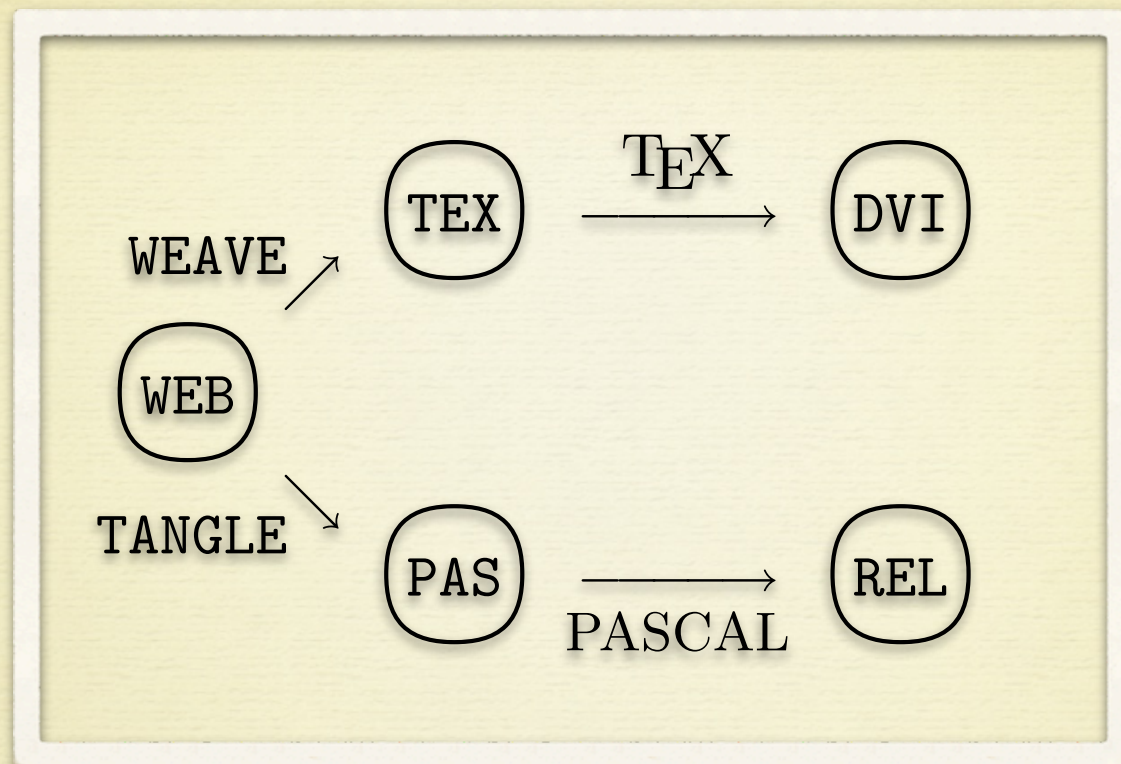
- ☐ WEB SHOWS THAT THESE ARE NO LONGER OPPOSING METHODOLOGIES
- ☐ PROGRAMS ARE WEBS, NOT TREES
 - ☐ HIERARCHICAL STRUCTURE IS PRESENT, BUT STRUCTURAL RELATIONSHIPS ARE MORE IMPORTANT
- ☐ THE WEB SYSTEM LEADS TO “STREAM OF CONSCIOUSNESS” PROGRAMMING
 - ☐ A PROGRAM IS DESCRIBED IN THE BEST EXPOSITORY ORDER; TANGLING UNSCRAMBLES THE CODE INTO COMPILABLE FORM

The WEB System

- ▣ WEB COMBINES TWO OTHER LANGUAGES:
 - ▣ A DOCUMENT FORMATTING LANGUAGE
 - ▣ A PROGRAMMING LANGUAGE
- ▣ THE ORIGINAL WEB PROTOTYPE USED TEX AND PASCAL, BUT WEB WORKS WITH ANY LANGUAGE COMBINATION
- ▣ WEB IS “INHERENTLY BILINGUAL”
 - ▣ USING WEB, COMPONENT LANGUAGES ARE ENHANCED; THEY ARE MORE THAN THE SUM OF THEIR PARTS

The Literate Programming Process

- ▣ WEB PRODUCES A DUAL-USE PROGRAM
 - ▣ THIS PROGRAM IS THE SOURCE LANGUAGE FOR TWO SYSTEM ROUTINES: WEAVING AND TANGLING
- ▣ WEAVING PRODUCES A DOCUMENT THAT CLEARLY DESCRIBES THE PROGRAM
- ▣ TANGLING PRODUCES A MACHINE-EXECUTABLE PROGRAM
 - ▣ “COMPILE & EXECUTE” BECOMES “TANGLE, COMPILE, & EXECUTE”
- ▣ THE PROGRAM AND DOCUMENTATION ARE PRODUCED FROM THE SAME SOURCE, SO THEY STAY CONSISTENT



Dual-Usage, Illustrated

WEB Programming Style

- ☐ WEB PROGRAMS ARE BROKEN INTO SMALL, CONSECUTIVELY-NUMBERED SECTIONS
- ☐ EVERY SECTION BEGINS WITH OPTIONAL COMMENTARY ABOUT THAT SECTION, AND ENDS WITH OPTIONAL PROGRAM TEXT
- ☐ PROGRAM TEXTS EITHER:
 - ☐ USE ACTUAL SNIPPETS OF CODE, OR
 - ☐ USE ANGLE BRACKETS TO REPRESENT ACTUAL CODE THAT APPEARS IN OTHER SECTIONS
- ☐ WEB AUTOMATICALLY GENERATES A “CHUNK INDEX” THAT LISTS ALL PROGRAM TEXTS

The Advantages of WEB

- ▣ WEB IMPROVES PROGRAM STRUCTURE
 - ▣ IT REDUCES THE APPARENT PROPORTIONAL SIZE OF RARELY-USED CODE (E.G., ERROR-HANDLING)
- ▣ COMPILATION TIME IS ESSENTIALLY UNCHANGED BY TANGLING
 - ▣ TYPESETTING TAKES LONGER DUE TO TEX
- ▣ TOTAL PROGRAMMING TIME IS EFFECTIVELY IDENTICAL TO NON-WEB PROGRAMMING TIME
 - ▣ ADDED DOCUMENTATION TIME IS OFFSET BY REDUCED DEBUGGING TIME