

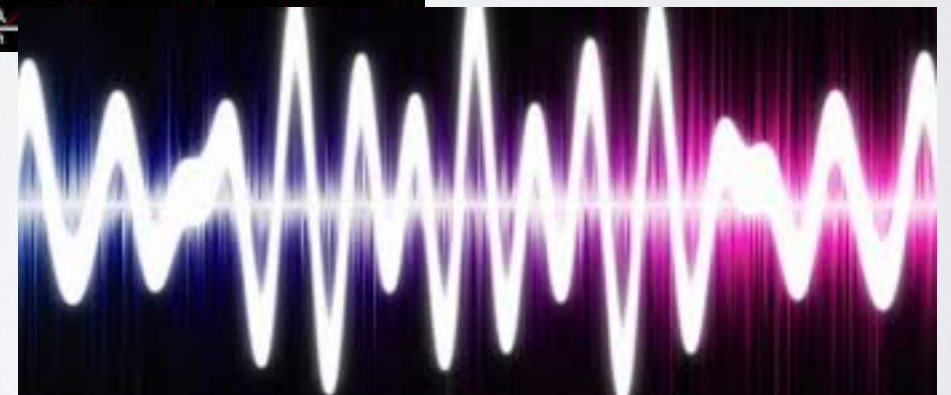
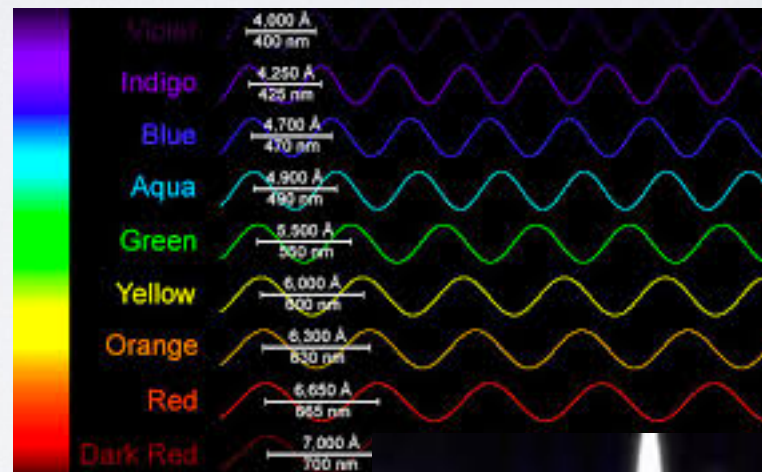
MAPPING AND METAPHORS

DET Chapter 3



MODELING

- Allows users to compare something they understand poorly to something they understand better



- Models can be overextended so don't take it too far

GOOD METAPHOR

- Simple
- Relates well to other relevant metaphors
- Explains much of the subject

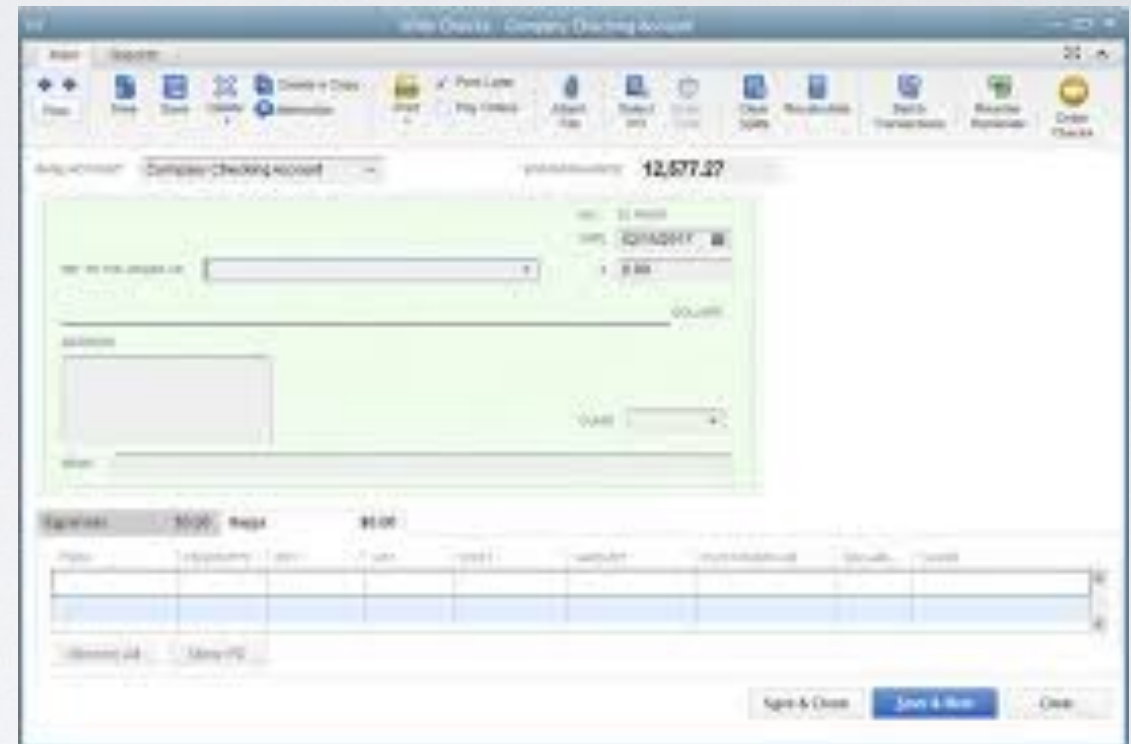


Aristotelian: falling object that comes to rest

Galileo: pendulum, repeating the same motion, almost perfectly

METAPHORS IN DESIGN

- Signifiers
- Constraints
- Natural mappings



METAPHORS IN SOFTWARE

- 1973 Turing Award lecture: Charles Bachman
 - Sun revolves around the Earth > Earth revolves around the sun
 - Early 1970s...data processing changed from a computer-centered view of information systems to a database-centered view

USING METAPHORS IN SOFTWARE

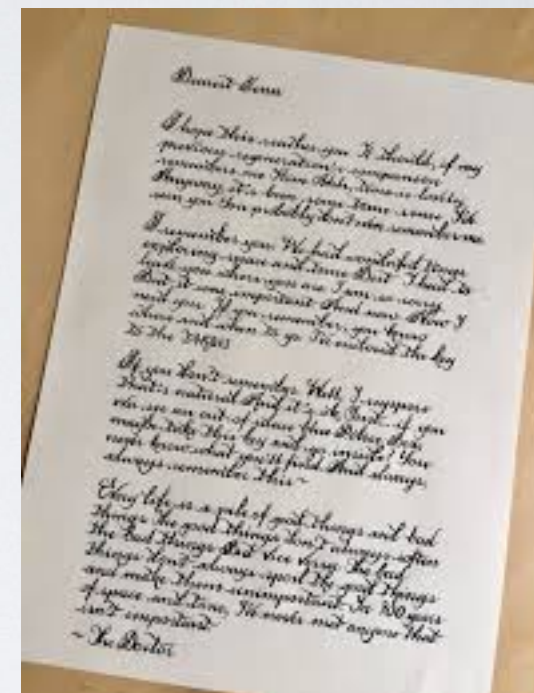
- Algorithm: predictable, deterministic, and not subject to chance, direct path from A to B
- Heuristic: technique that helps you look for an answer, subject to chance; How to look not what to find
- Most challenging part of programming is conceptualizing the problem - many errors in programming are conceptual errors

WRITING SOFTWARE...

- is a science - David Tries (1981)
- is an art - Donald Knuth (1998)
- is a process - Watts Humphrey (1989)
- is like driving a car - P.J. Plauger (1993), Kent Beck (2000)
- is a game - Alistair Cockburn (2002)
- is like a bazaar - Eric Raymond (2000)
- is like gardening - Andy Hunt and Dave Thomas (1994)
- is like filming Snow White and the Seven Dwarfs (1994)
- is like farming, hunting werewolves, or drowning with dinosaurs in a tar pit (1995)

SOFTWARE PENMANSHIP: WRITING CODE

- Writing metaphor suggests that developing a program is like writing a casual letter
- You should be able to sit down by the fire with a glass of brandy, a good cigar, and your favorite hunting dog to enjoy a “literate program” the way you would a good novel - Jon Bentley



WRITING CODE

- Accuracy of the metaphor?
- Problems with the metaphor?

WRITING CODE

- Accuracy of the metaphor?
 - individual work
 - small-scale projects
- Problems with the metaphor?
 - doesn't describe software development fully
 - involves one person where writing typically is a one-person activity
 - software is never complete (66%-90% of development efforts happens after initial
 - not original (reuse of design ideas, code, and test cases)

WRITING CODE



- Fred Brooks's *The Mythical Man-Month* says to plan to throw one away
- Unreasonable...cost is enormous
- Get it the first time and take chances when the cost is the least

SOFTWARE FARMING: GROWING A SYSTEM

- design a piece, code a piece, test a piece, and add it to the system in an attempt to minimize the trouble you can cause
- Incremental technique is valuable, farming metaphor...not so much



SOFTWARE OYSTER FARMING: SYSTEM ACCRETION

- Accretion: any growth or increase in size by a gradual external addition or inclusion
- Incremental, iterative, adaptive, evolutionary
- Incremental development: simplest version of the system that runs (skeleton)
 - Slowly add the muscle and skin



SOFTWARE CONSTRUCTION: BUILDING SOFTWARE

- Building software implies various stages of planning, preparation, and execution varying in degree depending on what's being built
- Building process can be more complicated depending on the project and poor design/mistakes have a larger impact



BUILDING SOFTWARE

- What kind of house - problem definition
- Architect must get house approved - software architectural design
- Blueprints - detailed software design
- Prepare building site, lay foundation, frame, siding, roof, plumb/wire - software construction
- Landscaping, painting - software optimization
- Inspectors - software reviews and inspections (QA)

BUILDING SOFTWARE

- Making changes to a house is comparable to making changes in software
- Load bearing wall will cost more to change than a simple dividing wall, the same is true for code. It's harder to make structural changes to code than to add or delete peripheral features

BUILDING SOFTWARE

- Builders make and inspect plans carefully
- Better to pay 10 percent more for stronger material than to have a skyscraper fall
- Timing is important
 - When the Empire State Building was built, each delivery truck had a 15-minute margin in which to make its delivery or the project would be delayed

BUILDING SOFTWARE

- A software system with one million lines of code requires an average of 69 kinds of documentation (Capers Jones, 1998)
- Requirements specification: 4000-5000 pages long
- Design documentation: 2-3 times as extensive



APPLYING SOFTWARE TECHNIQUES: THE INTELLECTUAL TOOLBOX

- Effective, high-quality software development requirements dozens of techniques, tricks, and magic incantations
- Not rules but analytical tools
- You have to know the right tool for the job
- Don't buy into one methodology 100%, you'll miss better methods suited to the current problem



CONCLUSION

- Metaphors are heuristics, not algorithms - can be fuzzy
- Metaphors help you understand the software-development process by relating it to other activities you already know about
- Some metaphors are better than others
- Treating software construction as similar to building construction suggests that careful preparation is needed and illuminates the difference between large and small projects
- Thinking of software development practices as tools suggest further that every programmer has many tools. Choosing the right tool for each problem is one key to being an effective programmer
- Metaphors are not mutually exclusive. Use the combination of metaphors that works for you