

Systems Principles

Lorenzo Alvisi

What is this course about?

- The common ideas, techniques, insights, and challenges that arise in the design of a variety of systems that are typically studied separately
 - OSes
 - Databases
 - Distributed Systems
 - Networks
- An attempt at synthesis
- An experiment
- A work in progress

Our goals

- Acquire a broad view of systems
- Develop a sense of taste for what makes a good system design
- Avoid having your next system be used as an example of what **not** to do in a future version of this course

What is a system?

"A complex unit formed of many often diverse parts subject to a common plan or serving a common purpose"

Webster Third New International Dictionary

What is a system?

interconnections

components

"A complex unit formed of ~~many often~~ ~~diverse parts~~ subject to a common plan or serving a common purpose"

Webster Third New International Dictionary

a certain behavior
at the interface
with an environment

"A set of interconnected components with an expected behavior observed at the interface with its environment"

Common systems challenges

- Emergent properties
- Propagation of effects
- Incommensurate scaling
- Trade-offs

Emergent properties

- True of the whole, but not of each component
- Often unforeseen until the system is built



Emergent properties

- Tenex (a time sharing system for PDP-10) had the following individually harmless features
 - A reference to an unassigned virtual page generates a trap to the user program
 - A syscall is seen as a machine instruction for an extended machine, so if it references an unassigned page, a trap is returned to the user program
 - Large arguments to syscall (such as strings) are passed by reference
 - The CONNECT syscall gives access to another directory, with the password as a string argument

Emergent properties

CONNECT's implementation:

```
for i = 0 to Length(directoryPassword) do
  if directoryPassword[i] ≠ passwordArgument[i] then
    Wait three seconds; return BadPassword
  endif
end loop
connect to directory; return Success
```

Suppose strings are made of 7 bit characters:
how many tries to crack a password of length n ?

Arrange *passwordArgument* so its first char is the last char of a page, and the next page is unassigned.

Try a character

If *BadPassword* the guess was wrong; if trap on unassigned page, it was correct

Repeat...

Propagation of effects



To improve ride experience,
change wheels from 14" to 16"

- ❑ Redesign wheel well
- ❑ Redesign trunk for bigger spare wheel
- ❑ Move back seat slightly forward
- ❑ Thin back seat
- ❑ Stiffen rear springs
- ❑ Change speedometer gearing
- ❑ ...

Propagation of effects: fighting malaria



① WHO sprayed villages in N. Borneo with DDT

② Wiped out mosquitos, but...



③ Roaches collected DDT in tissue



④ Lizard ate roaches, and became slower



⑤ Easy target for cats

⑥ DDT cause cats to die

⑦ Forest rats moved into villages

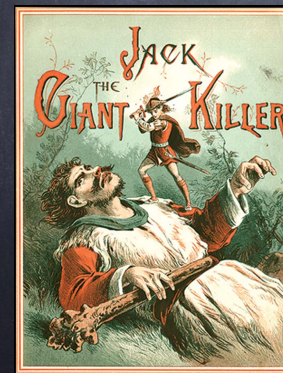


⑧ Rats carried the bacillus of the **plague!**



Incommensurate scaling

As the system increases in size or speed, not all components can manage the scale, and things break down



10x higher than Jack!

but also 10x wider and thicker!

About 1000x Jack's weight – but their cross section of their bones was only 100x Jack's

A human thigh bone breaks at about 10x human weight

The giant would have broken his thighs every time he was taking a step!

On being the right size
J.B.S. Haldane

Trade-offs

Speed vs power in processors

Bandwidth vs computation in compression

Space vs time almost everywhere

A pawn vs better position in chess

...

"A complex unit formed of many often diverse parts subject to a common plan or serving a common purpose"

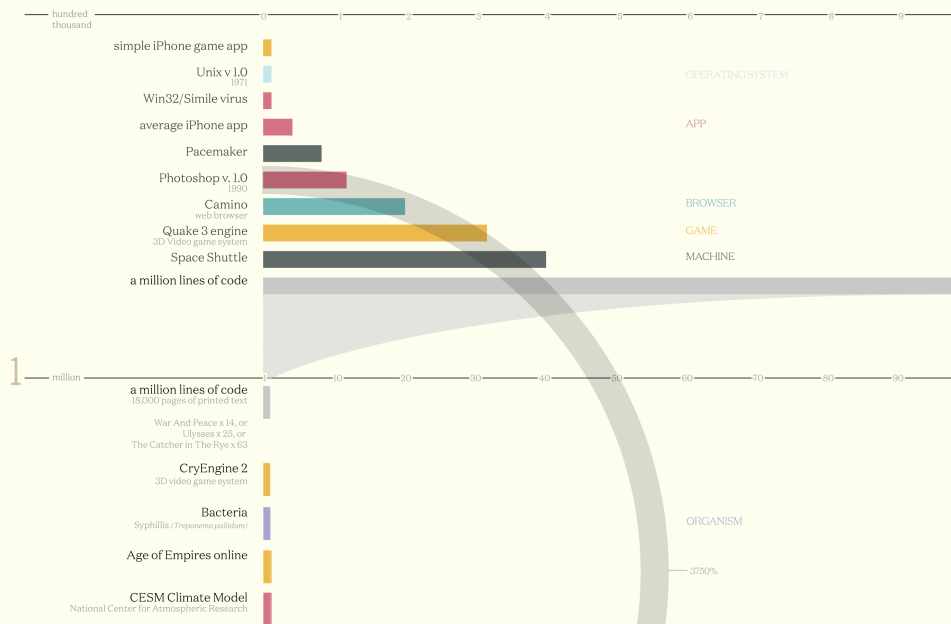
"A **complex** unit formed of many often diverse parts subject to a common plan or serving a common purpose"

Complexity: the symptoms

- ④ Many components
- ④ Many interconnections
- ④ Many irregularities
 - many regularly interconnected components may still be fine...
- ④ Long description
 - hard to understand
- ④ Team of designers/implementors/maintainers

Codebases

Millions of lines of code

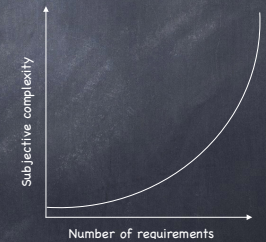


How do systems get complex?

🌀 Lots of subtly interacting requirements

- ❑ one may seem innocuous, but its interactions with the other formerly "innocuous" requirements are not!

Principle of escalating complexity
Adding a requirement increases complexity out of proportion



How do systems get complex?

🌀 Requirements change

- ❑ Changing architecture is hard: system is patched locally
- ❑ Full impact of patch is often unclear
- ❑ When it becomes visible, more patches!

How do systems get complex?

🌀 To address incommensurate scaling

J.B.S. Haldane

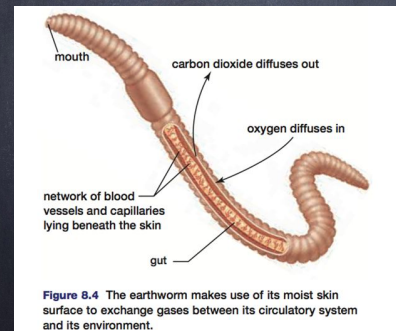
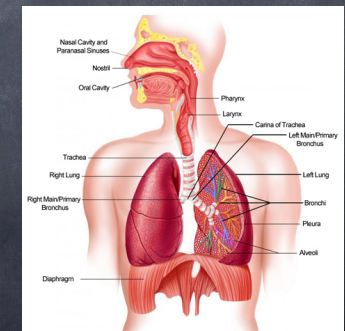


Figure 8.4 The earthworm makes use of its moist skin surface to exchange gases between its circulatory system and its environment.



The pitfall of generality



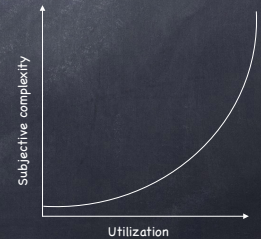
Do one thing at a time, and do it well. Don't generalize
B. Lampson

How do systems get complex?

⦿ To improve utilization of a limited resource



The law of diminishing return
The more one improves, the more effort
the next improvement will take



Managing Complexity: Modularity

⦿ Divide and conquer: modify a module without having to think what is inside others

□ Can save work

$$\text{Debug time} = \overset{\sim N}{\# \text{bugs}} \times \overset{\sim N}{\text{time to debug}}: \sim N^2$$

$$\text{By dividing in } k \text{ independent modules: } \sim \left(\frac{N}{k}\right)^2 \times k = \frac{N^2}{k}$$

Managing Complexity: Modularity

⦿ Divide and conquer: modify a module without having to think what is inside others

□ Can save work

$$\text{Debug time} = \overset{\sim N}{\# \text{bugs}} \times \overset{\sim N}{\text{time to debug}}: \sim N^2$$

$$\text{By dividing in } k \text{ independent modules: } \sim \left(\frac{N}{k}\right)^2 \times k = \frac{N^2}{k}$$

□ Makes it easier to respond to change...

□ ...but the specific modularity can become sclerotic



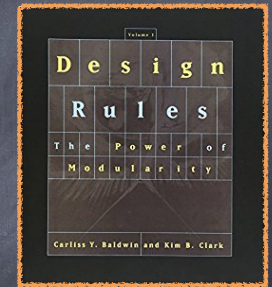
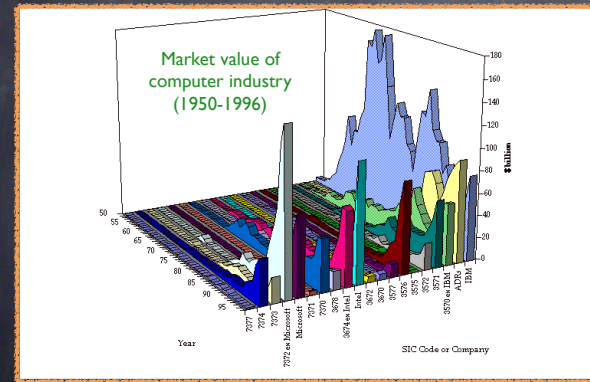
It must be remembered that there is nothing more difficult to plan, more doubtful of success, nor more dangerous to manage than the creation of a new system.

For the initiator has the enmity of all who would profit from the preservation of the old institutions and merely lukewarm defenders in those who would gain from the new ones

The Prince, 1513

The unyielding foundation rule
It is easier to change a module than the modularity

Modularity in action



Baldwin and Clark
Harvard

Managing Complexity: Abstraction

- Modules should treat other modules solely on the basis of their external specification
- Lots of ways to puncture modular abstractions:
 - Unwanted: bugs
 - Deliberate: looking for performance
- Three key abstractions to organize hardware:
 - Memory, Interpreter, Communication link

Suppressing noise

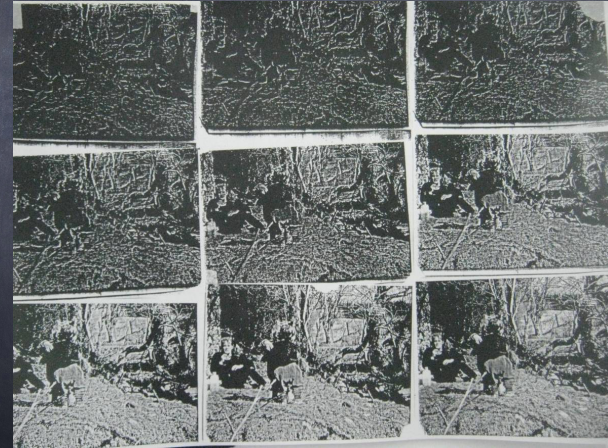
The robustness principle
Be tolerant of inputs and strict on outputs

...but raise a flag when values are out of range even if you can handle them

Why computer systems are more equal

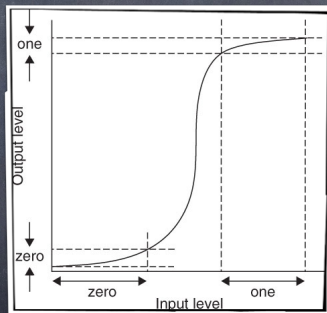
- ③ The complexity of a computer system is not limited by physical laws
- ③ The rate of change of computer systems is unprecedented

Analog systems have noise



Digital systems do not!

- ③ Signal levels are regenerated



The robustness principle
Be tolerant of inputs and strict on outputs

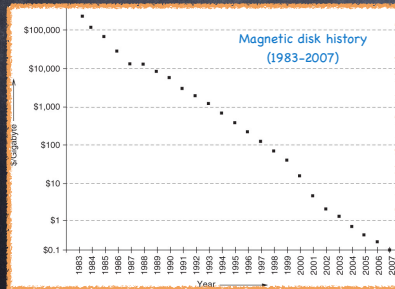
Wait! It gets worse!

- ③ Hardware at least has **some** physical limits
 - Speed of light
 - Heat dissipation
 - Rate of settling of signals
 - Unwanted electrical coupling
 - Space
- ③ Software (virtually) does not
 - The real limit is our ability to understand the system we built
 - Enters feature creep...

The rate of technology change

The cost of digital hardware has
dropped 30% / year

and even more for some components



The incommensurate
scaling rule
A 10X change in a
system parameter usually
requires a new design

What does it mean to engineer systems that change so quickly?

- ⦿ Less emphasis on optimization
 - Can get performance by just waiting
- ⦿ Rough edges
 - Getting usability right takes time
 - Figuring out the best solution takes time
- ⦿ Legal and judicial processes cannot keep up