Operating Systems COT 4600 – Fall 2009

Dan C. Marinescu
Office: HEC 439 B
Office hours: W, F 3:00-4:00 PM

Lecture 3

- Last time
  - Computer Systems
- Today:
  - Names
  - Computer Systems versus Other Systems
  - Coping with Computer System Complexity
- Next time
  - The project
- Homework 1 due Thursday, September 3, 2009

Names

- Modularity along with abstraction, layering, and hierarchy allow a designer to cope with complexity;
- Names and addresses ➔ provide the means to connect modules.
- A system ➔ a bunch of resources, glued together with names
- Naming allows the designer to:
  - Delay the implementation of some modules; use dummy ones
  - Replace an implementation with another one.
- Binding➔ choosing an implementation for a module
  - Delayed binding; use a place holder.

Names and fundamental abstractions

- The fundamental abstractions
  1. Storage ➔ mem, disk, data struct, File Systems, disk arrays
  2. Interpreters ➔ cpu, programming language e.g. java VM
  3. Communication➔ wire, Ethernet

  rely on names.
- Naming:
  - Flat
  - Hierarchical

Computers a distinct species of complex systems

- The complexity of computer systems not limited by the laws of physics ➔ distant bounds on composition
  - Digital systems are noise-free.
  - The hardware is controlled by software
- The rate of change unprecedented
  - The cost of digital hardware has dropped in average 30% per year for the past 35 years

Analog, digital, and hybrid systems

- Analog systems:
  - the noise from individual components accumulate and
  - the number of components is limited
- Digital systems:
  - are noise-free
  - the number of components is not limited
  - regeneration ➔ restoration of digital signal levels
  - static discipline➔ the range of the analog values a device accepts for each input digital value should be wider than the range of analog output values
  - digital components could fail but big mistakes are easier to detect than small ones!!
- Hybrid systems ➔ e.g., quantum computers and quantum communication systems

Computers are controlled by software

- Composition of hardware limited by laws of physics.
- Composition of software is not physically constrained;
  - software packages of 10^7 lines of code exist
- Abstractions hide the implementation beneath module interfaces and allow the
  - creation of complex software
  - modification of the modules
- Abstractions can be leaky. Example, representation of integers, floating point numbers.

Exponential growth of computers

- Unprecedented:
  - when a system is ready to be released it may already be obsolete.
  - when one of the parameters of a system changes by a factor of
    - 2 ➔other components must be drastically altered due to the incommensurate scaling.
    - 10 ➔ the systems must be redesigned; E.g.; balance CPU, memory, and I/O bandwidth;
  - does not give pause to developers
    - to learn lessons from existing systems
    - find and correct all errors
  - negatively affects “human engineering” ➔ ability to build reliable and user-friendly systems
  - the legal and social frameworks are not ready

Coping with complexity of computer systems

- Modularity, abstraction, layering, and hierarchy are necessary but not sufficient.
- An additional technique ➔ iteration
- Iteration
  - Design increasingly more complex functionality in the system
  - Test the system at each stage of the iteration to convince yourself that the design is sound
  - Easier to make changes during the design process

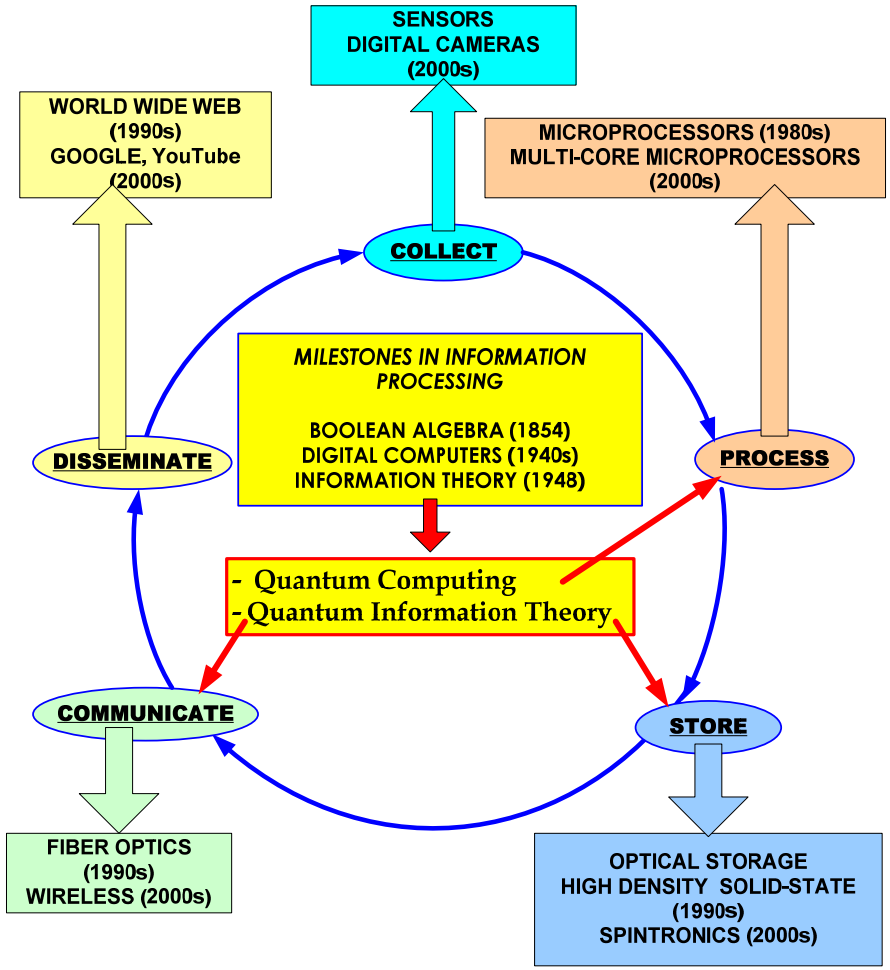
Iteration – design principles

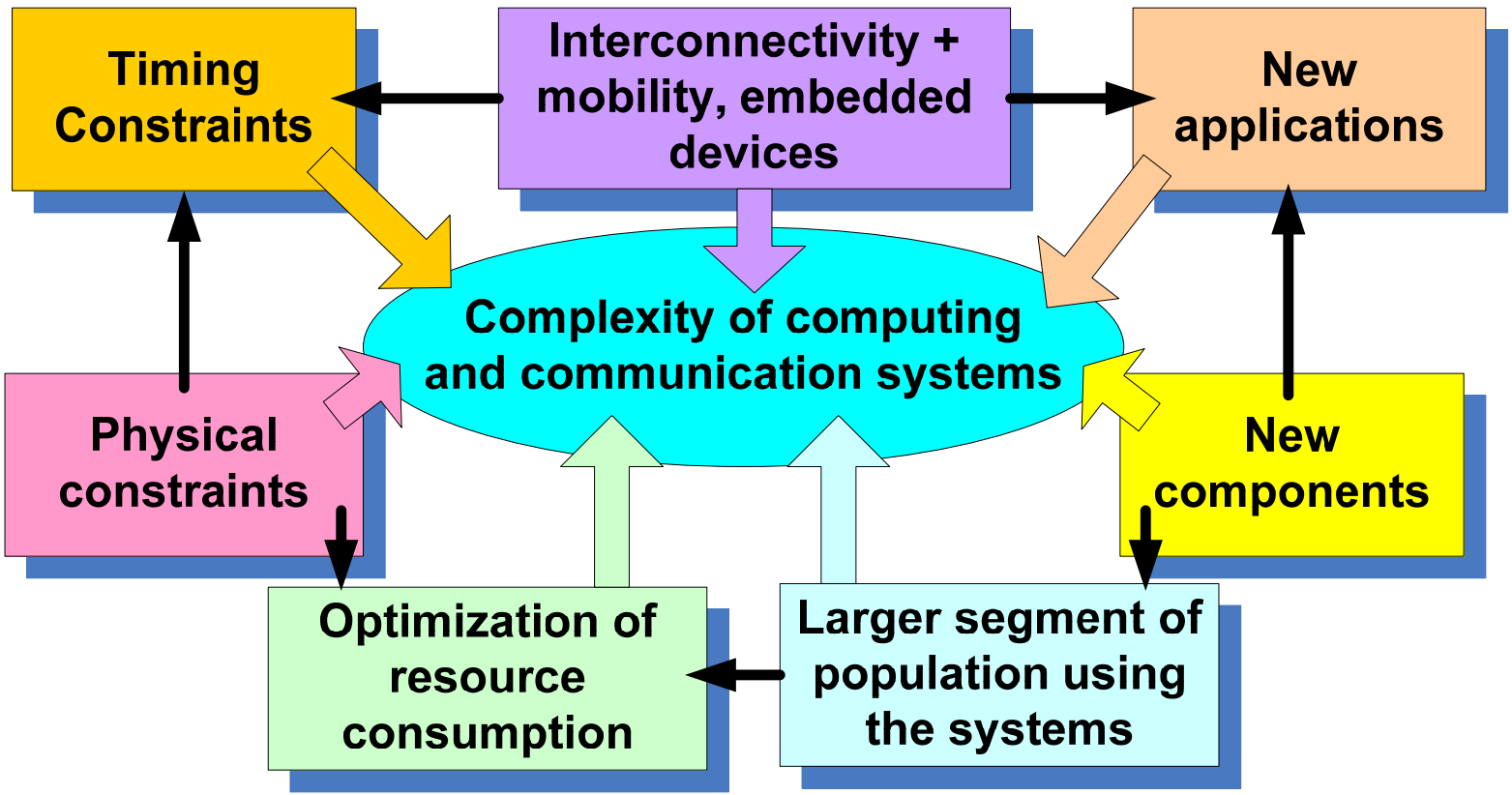
- Make it easy to change
  - the simplest version must accommodate all changes required by successive versions
  - do not deviate from the original design rationale
  - think carefully about modularity ➔ it is very hard to change it.
- Take small steps; rebuild the system every day, to discover design flaws and errors. Ask others to test it.
- Don’t rush to implementation. Think hard before starting to program.
- Use feedback judiciously➔
  - use alpha and beta versions
  - do not be overconfident from an early success
- Study failures ➔ understand that complex systems fail for complex reasons.

Curbing complexity

- In absence of physical laws curb the complexity by good judgment. Easier said than done because:
  - tempted to add new features than in the previous generation
  - competitors have already incorporated the new features
  - the features seem easy to implement
  - the technology has improved
  - human behavior: arrogance, pride, overconfidence…

Critical elements of information revolution!



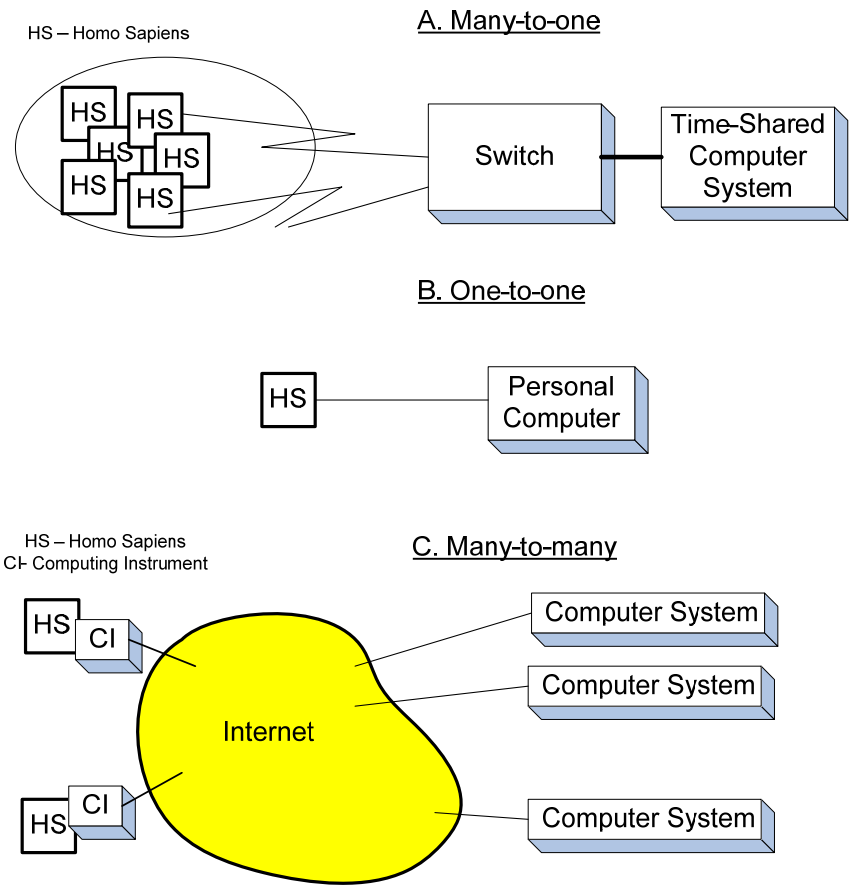

Timing Constraints
Interconnectivity + mobility, embedded devices
New applications
Complexity of computing and communication systems
Physical constraints
New components
Optimization of resource consumption
Larger segment of population using the systems

The relation between homo sapiens and the computers

- The feelings of the homo sapiens:
  - Hate
  - Frustration
  - Lack of understanding
- The Operating System
  - A program to “domesticate” the computer.
  - Transforms a “bare machine” into a “user machine”
  - Controls and facilitates access to computing resources; optimizes the use of resources.
- The relation went through several stages:
  - Many-to-one
  - One-to-one
  - Many-to-many
  - Peer-to-peer

Resource sharing and complexity

- A main function of the OS is resource sharing.
- Sharing computer resources went through several stages with different levels of complexity:
  - Many-to-one
  - One-to-one
  - Many-to-many
  - Peer-to-peer


HS – Homo Sapiens
A. Many-to-one
HS
HS
HS
HS
HS
HS
Switch
Time-Shared Computer System
B. One-to-one
HS
Personal Computer
HS – Homo Sapiens
CI- Computing Instrument
C. Many-to-many
HS
CI
HS
CI
Internet
Computer System
Computer System
Computer System

HS – Homo Sapiens
PC- Personal Computer I