Building User Interfaces
Advanced HCI
IAT351

Week 2 Lecture 1

9.05.2012

Lyn Bartram

lyn@sfu.ca

Administrivia

- Website and wiki are up!
  - http://www.sfu.ca/siatclass/IAT351/Fall2012/
  - https://wiki.sfu.ca/fall12/iat351
- Remember to make a wiki page please!

Evaluation: Two options - VOTE

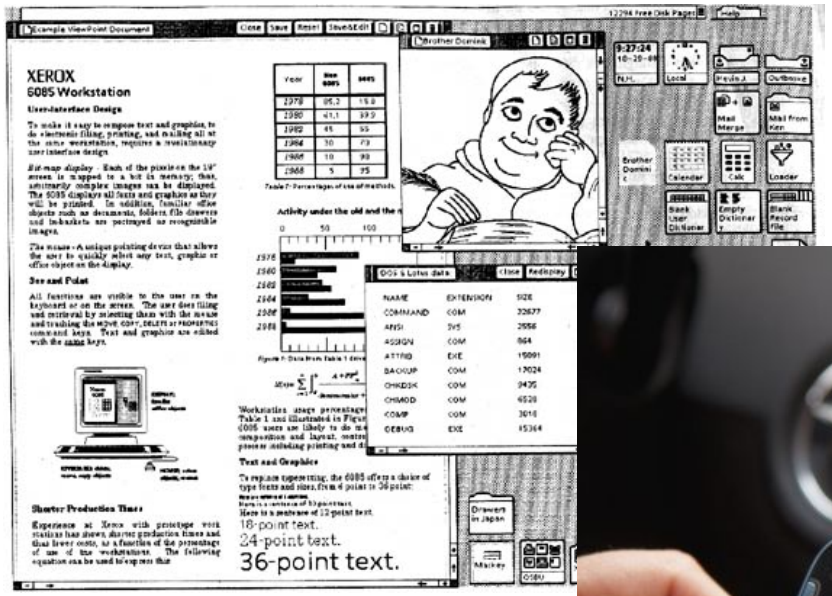
- Five programming assignments
  - Each worth 15 %
  - **Randomised pairs**
  - Spaced pretty evenly throughout the term
- Research project 25%
  - Individual
  - Paper and/or prototype
  - Research and discuss identified issues and approaches in HCI
  - Agreed with instructor
  - Due at end of term

- Four programming assignments
  - Each worth 15 %
  - **Randomised pairs**
  - Mostly in first 1/2 of term
- Midterm 25%
- Research project 15%

Motivation

- Moore's law has done its job
  - No longer: " can it be built" ?
  - Now: "can they use it?" or, more particularly, "how many ways can it be used?": "will they use it"? --> "Can I sell it"?
- Shift towards usability and user experience as a key product denominator
  - Good interaction design
  - Good visual design
  - iPod/iPhone, distributed systems, web services .....
  - Implicit, explicit, ambient, embedded interaction and display

Think of the UI as part – the touchpoints – of an interactive system



Why study UI software?

- Most systems involve some user somewhere
- Good user interfaces are critical for software survival and economics
- Designing for users is **important** and **difficult**
  - Lots of code devoted to the UI
  - Hard to get right the first time (iteration necessary)

Why are user interfaces difficult to build?

- They are reactive and are programmed from the "inside-out"
  - Event based programming
  - More difficult to modularize

Why are user interfaces difficult to build?

- They generally require multi-processing and concurrency
  - To deal with user actions; aborts
  - Displayrefresh
  - Different displays and windows system as a different process
  - Multiple input devices
  - Performance balancing
  - Real-world events

Why are user interfaces difficult to build?

- There are real-time requirements for handling input events
  - Output 60 times a second
  - Keep up with mouse tracking
  - Video, sound, multi-media
- Need for robustness
  - No crashing, on any input
  - Helpful error messages and recover gracefully
  - Aborts
  - Undo

Why are user interfaces difficult to build?

- Good UI design and construction requires **system thinking**
- They are non-deterministic
  - (translation: you will NEVER know everything that can happen!)
- Frequent early prototyping and refactoring is critical
- Needs good architecture from the outset

Some good news

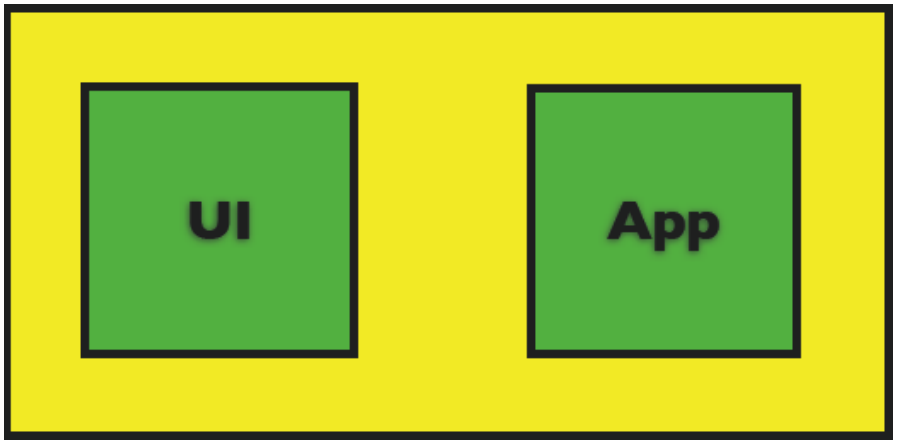
- Large set of tools and middleware layers that abstract the lowest level details away from application programmer
  - .NET
  - Java , Swing
- Don't write a device driver any more, write an extension of some device object to meet new criteria
- UI Builders : we will NOT be relying on these in this course
  - Useful for prototyping
  - Often generate bad software engineering :)

What's the user interface?

- Since mid-40's
  - Display (paper terminal, CRT, LCD, ...)
  - Keyboard
  - "command line", Unix interaction
- Since late '60's
  - Pointing device
  - WIMP/GUI style of interaction
  - "direct manipulation"
  - The desktop
- Since early '90's
  - An extension of our physical environment
  - Sensing, inferencing
  - Different modalities : sound, touch

Programmer's perspective

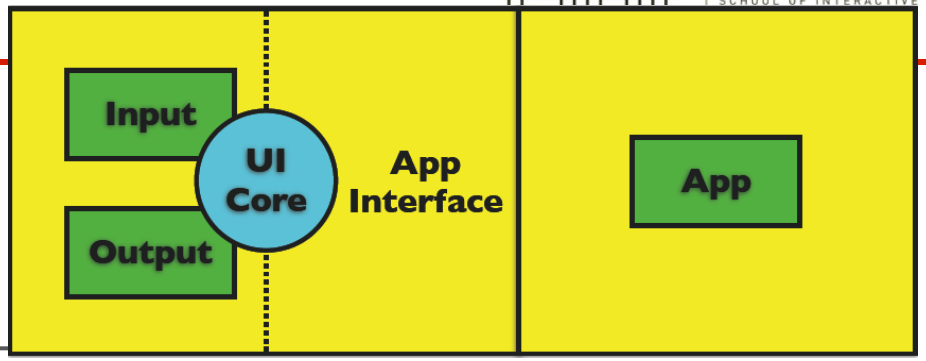
- The "UI" was/is typically viewed as one component of the overall system
  - The part that "deals with the user"
  - Separate from the "functional core" (the application)



The Software engineering approach

- Advantages of “separation of concerns”
  - Keep UI code separate from app code
  - Isolate changes
  - More modular implementation
  - Different expertise needed
  - Don’t want to iterate the whole thing
  - Design in parallel

Hmm, in practice



This is very hard to do in real-world application environments

- More and more interactive programs are “tightly coupled” to the UI
  - Programs are structured around the UI concepts and the flow of behaviour
  - Lower level support needs to be present to enable higher-level behaviour
- Not always bad
  - Tight coupling can lead to better performance and feedback

UI History : A Quick View
IAT351

Week 2 Lecture 2

10.09.2012

Lyn Bartram

lyn@sfu.ca

Some slides adapted from K. Edwards, Georgia tech

Evolution of computing

- Digital computer grounded in ideas from 1700s & 1800s
- Technology became available in the 1940s and 1950s
- 1950s - 1960s
  - Computers appeared on the commercial scene
  - Difficult to use, cumbersome
  - Ran in “batch-mode”, experienced operators
  - Cards
- Early 1960s - 1980s
  - Timesharing systems
  - Manual command line

Evolution of INTERACTIVE computing

- 1970s
  - First personal computers
  - Raster graphics-based networked workstations
  - Mouse pointing devices, desktop metaphor, windows, icons
  - WIMP
  - Widespread adoption
  - Man-machine interface (MM!)
- mid 1980s - now
  - Human-Computer interaction (HCI)

Vannevar Bush

Vannevar Bush

- "…publication has been extended far beyond our present ability to make real use of the record."
  - "As We May Think" - 1945 *Atlantic Monthly*

- Postulated **Memex** device
  - Can store all records/articles/communications
  - Large memory
  - Items retrieved by indexing, keywords, cross references
  - Can make a **trail** of **links** through material, etc.

- Envisioned as microfilm, not computer

J.R. Licklider

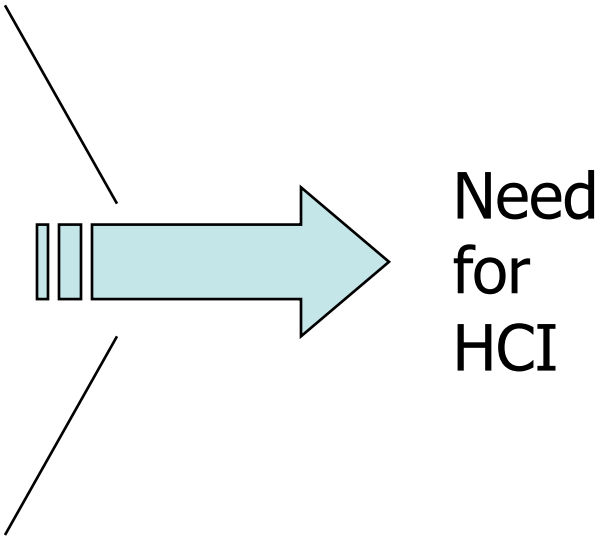
- 1960 - Postulated “man-computer symbiosis”
- Couple human brains and computing machines tightly to revolutionize information handling

The Memex Vision/Goals

- Immed
  - Time sharing
  - Electronic I/O
  - Interactive, real-time system
  - Large scale information storage and retrieval
- Intermed
  - Combined speech recognition, character recognition, light-pen editing
- Long-term
  - Natural language understanding
  - Speech recognition of arbitrary users
  - Heuristic programming

Mid 1960's

- Computers too expensive for individuals -> timesharing
  - increased accessibility
  - interactive systems, not jobs
  - text processing, editing
  - email, shared file system
  - Single, dedicated task



Ivan Sutherland

- **SketchPad** - ‘63 PhD thesis at MIT
  - Hierarchy - pictures & subpictures
  - Master picture with instances (ie, OOP)
  - Constraints
  - Icons
  - Copying
  - Light pen as input device
  - Recursive operations

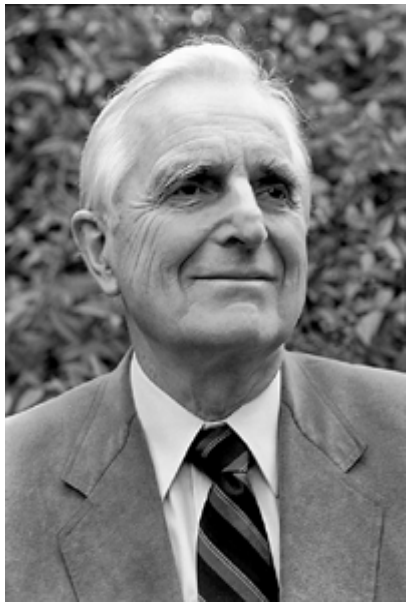
Video Display Units

- More suitable medium than paper
- Sutherland's Sketchpad as landmark system
- Computers used for visualizing and manipulating data

Douglas Engelbart

- Landmark system/demo:
  - hierarchical hypertext, multimedia, mouse, high-res display, windows, shared files, electronic messaging, CSCW, teleconferencing, ...

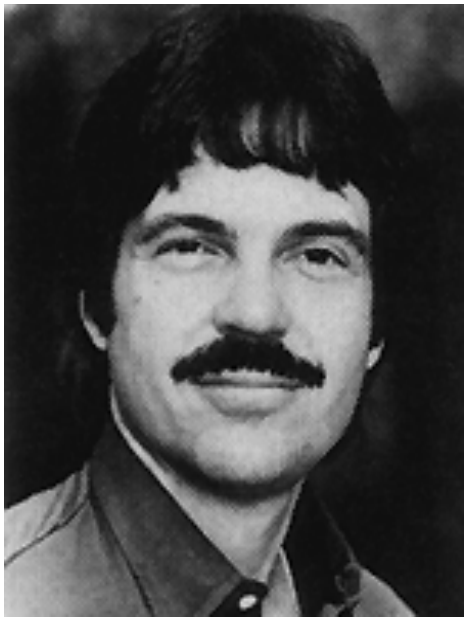
Inventor
of the mouse



Alan Kay

- Dynabook - Notebook sized computer loaded with multimedia and can store everything

Personal computing



Desktop interface

Personal Computing

- System is more powerful if easier to use
- Small, powerful machines dedicated to individual
- Importance of networks and time-sharing
- Kay's Dynabook, IBM PC

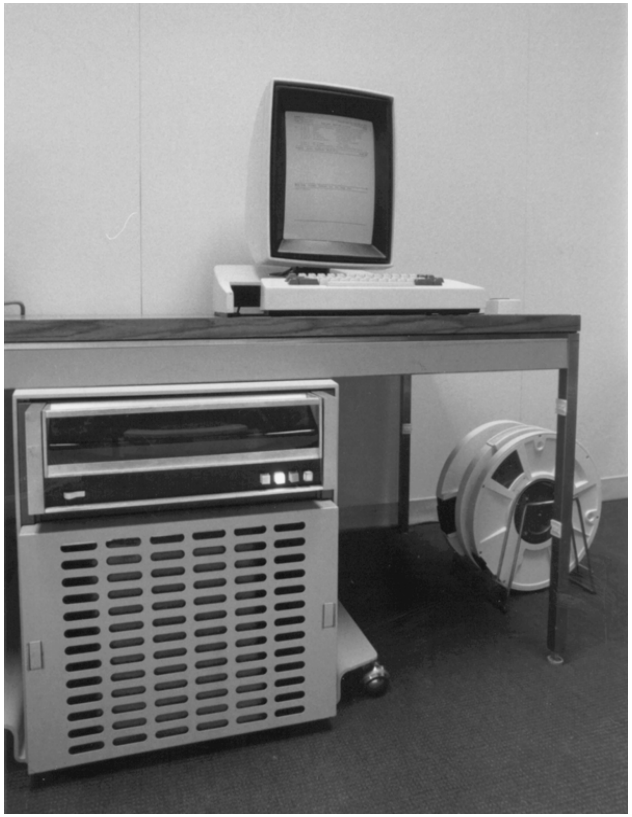
Personal Computers

- '70's IBM PC and the command line UI
  - Text and command-based: symbolic input
  - Monochrome
  - Recall not recognition
  - Required new control mappings and modes
    - Different across applications
  - Single input modality, serialised effort
  - Hard to learn - but efficient for experts
  - Small spatial/discrete capability (Rogue?)
  - Modal input

PCs with GUIs

Xerox PARC - mid 1970's

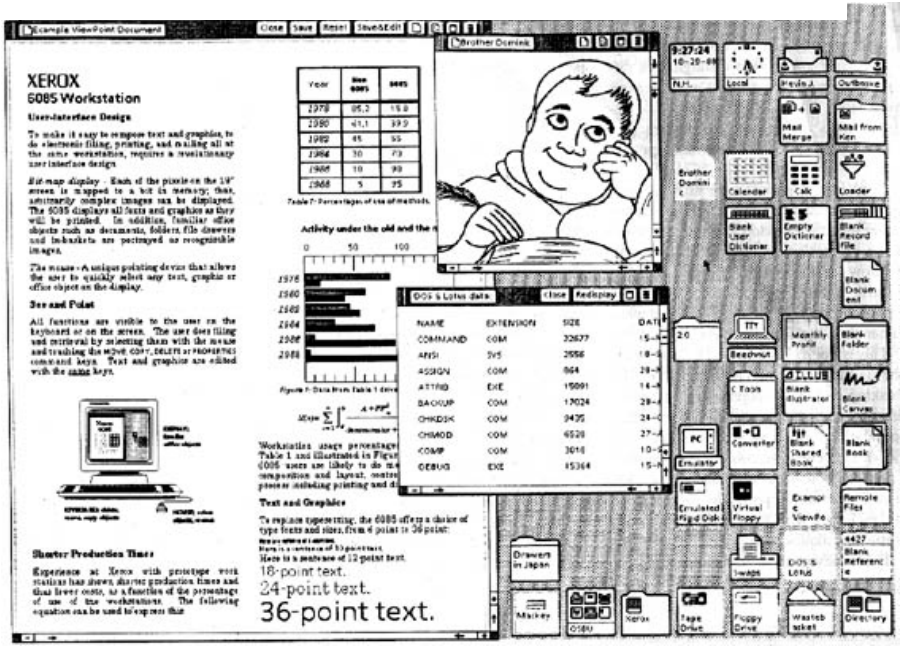
- **Alto**
  - local processor, bitmap display, mouse
  - Precursor to modern GUI, windows, menus, scrollbars
  - LAN - ethernet



Xerox Star - ‘81

desktop metaphor, pointing, WYSIWYG, high degree of consistency and simplicity

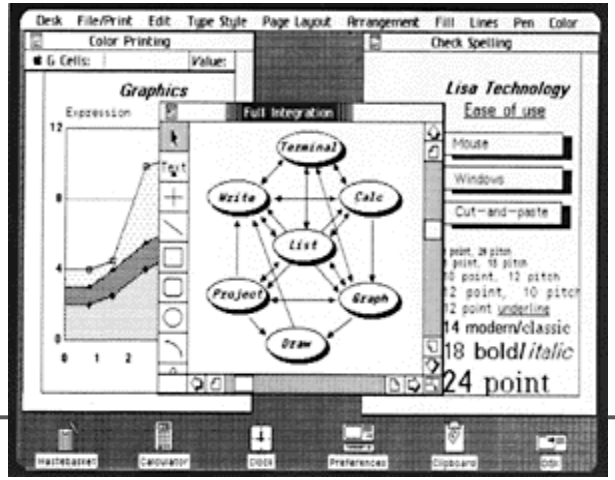
- First commercial PC designed for “business professionals”
  - Desktop publishing
- First system based on usability engineering
  - Paper prototyping and analysis
  - Usability testing and iterative refinement



Advent of the 2D input device
Pointing, selection, control

Apple Lisa - ‘82

- Based on ideas of Star
- More personal rather than office tool
  - Still $$$
- Failure



Apple Macintosh - ‘84

- Aggressive pricing - $2500
- Not trailblazer, smart copier
- Good interface guidelines
- 3rd party applications
- High quality graphics and laser printer

WIMP

- **W**indows, **I**cons, **M**enus, **P**ointers
- Can do several things simultaneously
  - Context switching
  - Start of the religious wars on tiled vs overlapping
- Familiar GUI interface - desktop metaphor
- Xerox Alto, Star; early Apples
- Used a mouse and a keyboard for input

Ben Shneiderman

- Coins and explores notion of direct manipulation of interface
- Long-time Director of HCI Lab at Maryland

Direct Manipulation

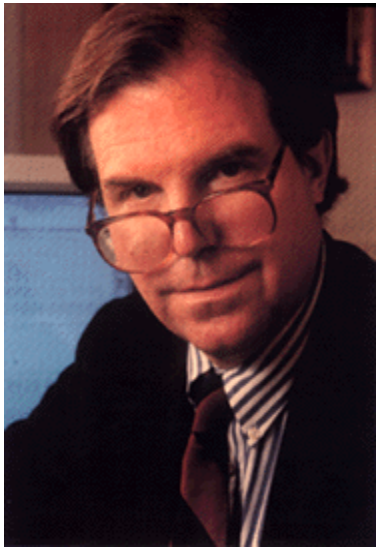
- ‘82 Shneiderman describes appeal of graphically-based interaction
  - object visibility
  - incremental action and rapid feedback
  - reversibility encourages exploration
  - replace language with action
  - syntactic correctness of all actions
- WYSIWYG, Apple Mac

Multimodality

- Mode is a human communication channel
  - Not just the senses, e.g., speech and non-speech audio are two modes
- Emphasis on simultaneous use of multiple channels for I/O
- Fragmentation and integration across many interaction channels
  - Multimodal != additive ??
- More intuitive ?

Nicholas Negroponte

- MIT machine architecture & AI group '69-' 80s
- Ideas:
  - wall-sized displays, video disks, AI in interfaces (agents), speech recognition, multimedia with hypertext



Language (Agents)

- Actions do not always speak louder than words
- Interface as mediator or agent
- Language paradigm
- Different communication mapping
  - Clifford Nass – learn from human-human communication

CSCW

Computer-Supported Cooperative Work

- No longer single user/single system
- Micro-social aspects are crucial
- E-mail as prominent success but other groupware still not widely used
  - Move to real-time and both f2f and remote
- Singular and shared interaction environments
  - Stanford iRoom
  - Multiple mice - I machine (Inkpen)
  - Remote interaction techniques
- WYSIWIS

Mark Weiser

- Introduced notion of “calm technology”
  - It’s everywhere, but recedes quietly into background
  - Ubiquitous computing
- CTO of Xerox PARC
- Sensors, actuators
- Vision and image processing

Ubiquity

- Person is no longer user of virtual device but occupant of virtual, computationally-rich environment
- Can no longer neglect macro-social aspects
- Late ‘90s - PDAs, VEs, ...
- 2000 - cell phones, RFID, tangible Uis …
- Large and small shared and partitioned devices
- Information is no longer device-singular in an application
  - Uniformity of techniques no longer applies?
  - Your machine - our information?

Immersive and manyD environments

Immersive Virtual reality (NSCA Cave™)

- Fred Brooks, Henry Fuchs
- Very specialised and hard to use
- Extremely useful in simulation
- Architecture

- Expensive and difficult to calibrate
- Perceptually challenging

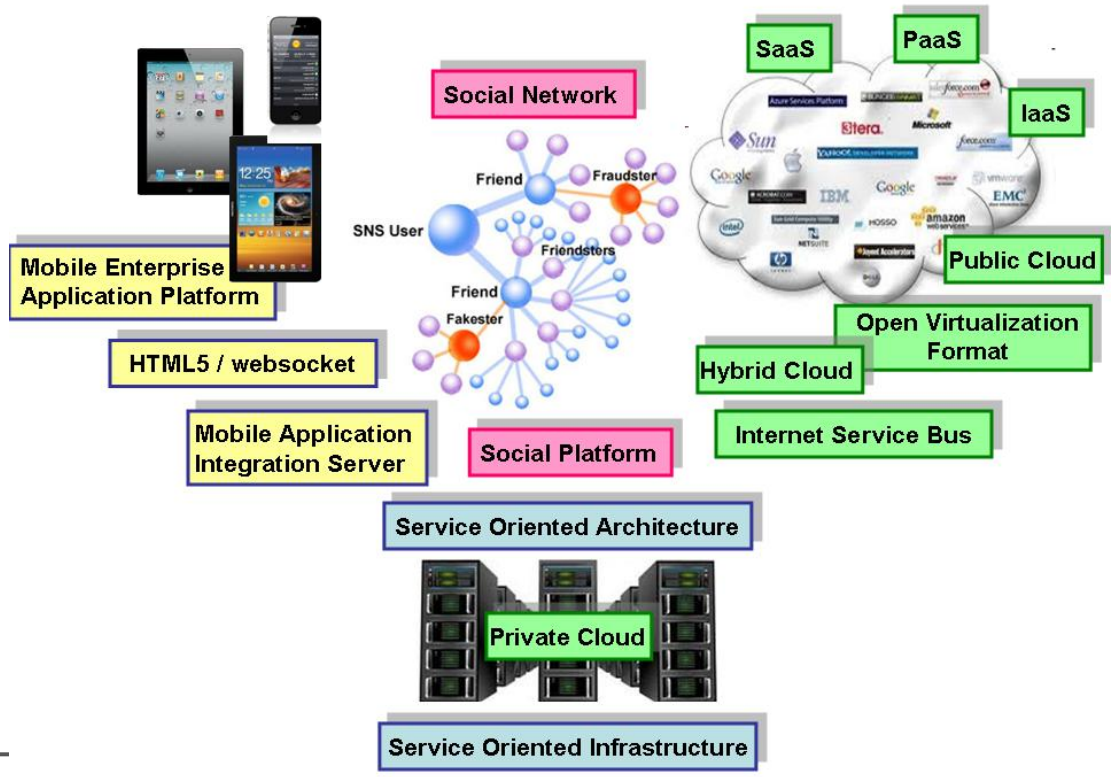
Immersive and manyD environments

- Fishtank VR and augmented reality
- Single or combined devices with many DOF (head tracker, Flock of Birds)
  - Human factors are challenging
- Stereo and large displays
  - Increasingly common usage especially in CSCW

The Cloud

New paradigm

- You are your data
- The cloud is all the available services
- Display
- interaction



Our bodies, our interaction devices …

New paradigm

Instrumenting the human

- Eye tracking/head tracking
- Motion capture
- High resolution direct input
- Less cognitive load?

Haptic and physical interfaces

- Using touch and force for direct input
- Sensors and other capture for indirect input (biomechanical signals-GSR)
- Tangible bits - Ishii
- The interactive floor