Designing a context-aware system is difficult
Suddenly, your app knows nothing with 100% certainty
What is it going to do now?
Context Aware Systems

- The world is fractured
- Context-Aware systems try to make sense of the chaos
Context Aware Systems

- Heterogeneity in hardware
  - Invisible embedded systems in cars, walls, objects
  - Feature phones, smart phones, tablets, ultra-books, netbooks, laptops, desktops
  - e-readers, mp3 players, personal health systems
  - Keyboard, Mouse, touch, gesture, tilt, eye-tracking
Context Aware Systems

- Heterogeneity in software and standards
  - Windows, iOS, Linux, Symbian, Android
  - Wifi, Bluetooth, Zigbee, WiMax, 4G, Ethernet, IrDA
Context Aware Systems

- Heterogeneity in sensing
  - Location
    - GPS
    - Cell-tower
    - Wifi
    - IP lookup
Context Aware Systems

- Heterogeneity in use-cases
  - Home
  - Office
  - Hospital
  - Car
  - Outdoors
  - Indoors
  - Crowds
  - Retail
  - Agriculture
  - Wilderness
Context Aware Systems

- Heterogeneity in use-cases
  - 1 : 1
    - Device = Owner
  - fallacy
  - 1 : many
    - Family Plan
  - many : many
    - Zipcar model
Many of the challenges only occur because of an application focus.

Many devices remain resource-constrained:
- CPU
- Memory
- Bandwidth
- Power (wireless comms)

- Resource-aware computing
- power foraging
- cyber foraging
Context Aware Systems

- Volatility is the rule not the exception
- Service discovery
  - Jini, UPnP, Bonjour, Bluetooth
- The system is distributed
  - "The set of users, devices, hardware, software and operating systems in ubicomp systems is highly dynamic and change frequently"
- Connections are volatile
- Network is volatile
Context Aware Systems

- Volatility of usage environment
  - Location of the users
    - Location-based services
  - Changing context of the computers
    - Context-aware computing
  - Multiple activities of the users
    - Activity-based computing (ABC)
Context Aware Systems

- Systems don’t get the attention of the user for free
  - Sending notifications
    - may never be seen
  - Asking what to do with a failure
    - user won’t respond
  - Asking the user to upgrade or install something else
- Autonomic computing
- Multi-agent systems
- Contingency Management
- Graceful degradation
Context Aware Systems

- Security and Privacy
- Trust
  - devices are not going to be under administrative control
- Resource assumptions are wrong
  - no access to security servers
  - no resources to compute crypto
- device is mobile
- Data is collected without users knowledge
- Short connections don’t lend themselves to passwords
  - refrigerator, HVAC, etc.
Design a system that adjusts your cell-phone ring tone. How does it work? What is a scenario that captures it’s ideal use case?
You live in a world where you depend on technology. If it fails hard you are in deep trouble.