Looking Forward to Ubiquitous Computing that Looks Ahead

Gaetano Borriello

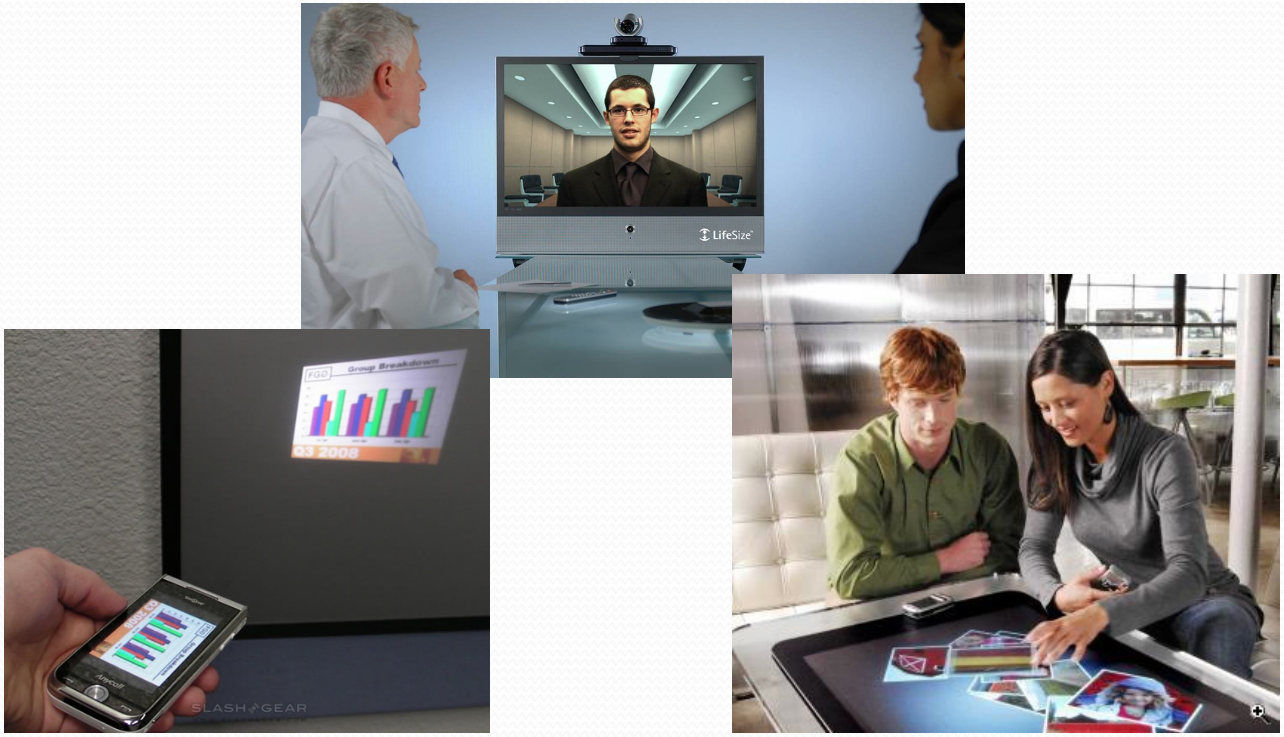
University of Washington
Department of Computer Science & Engineering

Outline

- Survey of the landscape
- The gadget trap
- What is missing?
- A blast from the past
- A manifesto

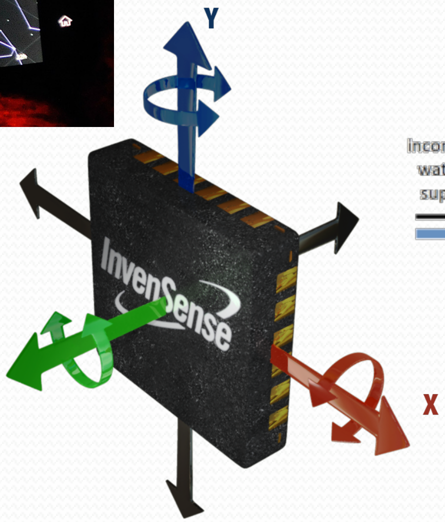
Tabs, Pads, and Boards

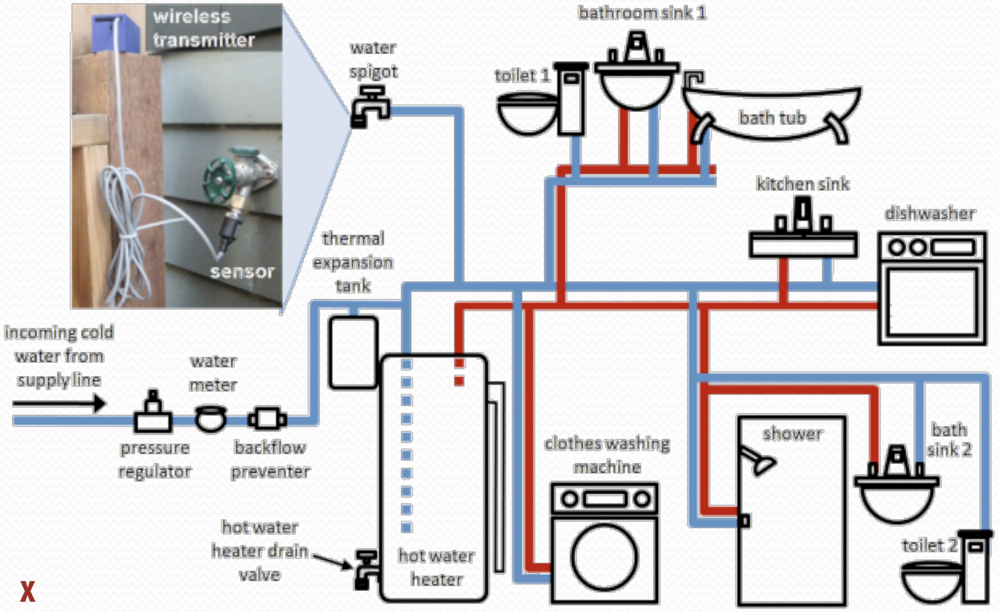
- Displays of all sizes
  - palm, notebook, wall, anywhere



Wireless Sensing

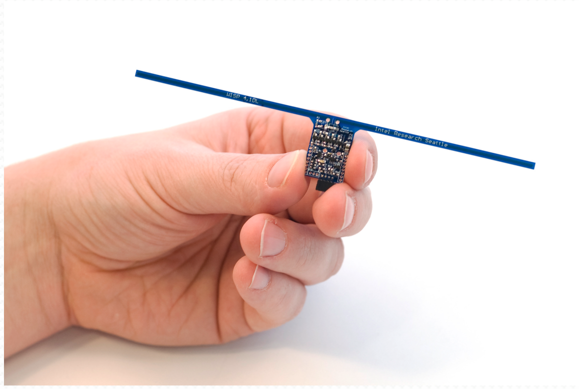
- Context-awareness from embedded sensors
  - location, movement, use of appliances





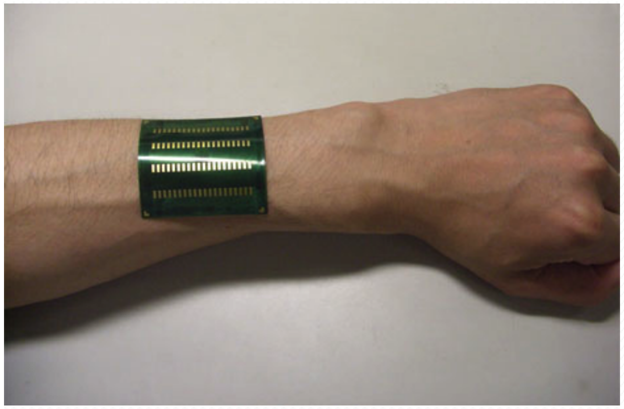
Communication

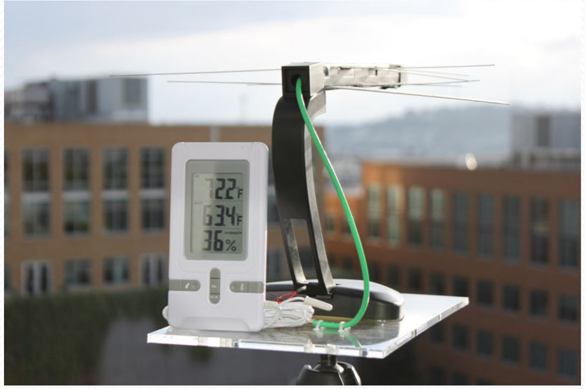
- High-bandwidth and low-power wireless
  - Cellular data, Wi-Fi, Wi-max, RFID



Power Harvesting

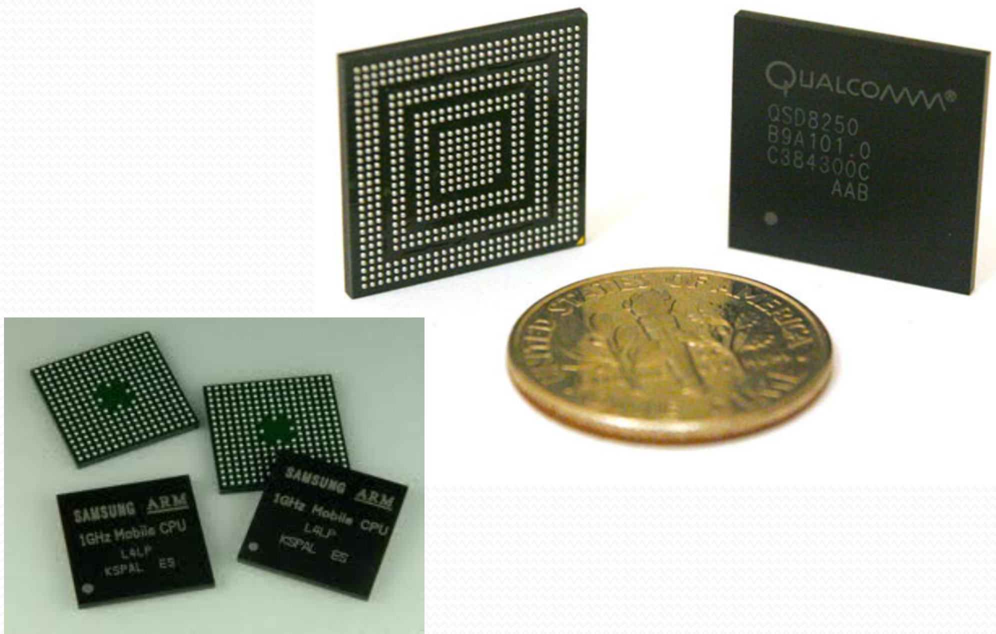
- Light, heat, mechanical, RF, inductive, vibration





Computing and Memory

- GHz multi-core processors in cell phones
- Cloud computing accessible over cell network



Let’s take stock

- We have all the pieces we said we needed to make ubiquitous computing a reality
- Yet, we have not met Mark’s vision…

> The most profound technologies are those that disappear.
> – Mark Weiser
>
> *Scientific American, Vol. 265, No. 3 (September 1991), pp. 94-104*

The Gadget Trap

- Every task needs the right tool
- But lots of tools are a pain to deal with, so combine

Universality Leads to Complexity

- That one device must have a more elaborate interface
- 100Ks of apps and even more if we count the web
- And we haven’t even scratched the surface of integration of home and car devices/apps
- We are spending way too much time learning and interacting with new tools
- This is NOT embedding computing into the fabric of everyday life – it is making IT the fabric of life

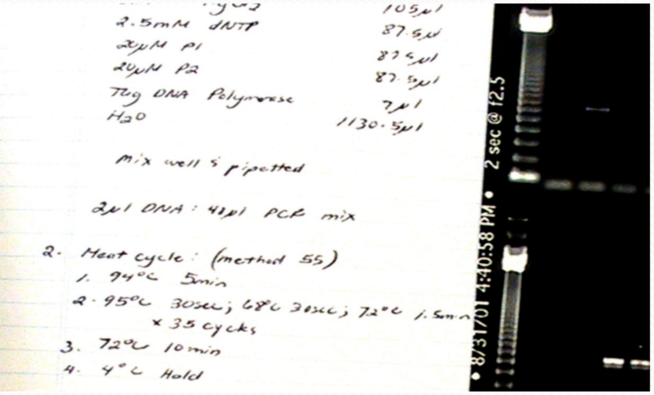
Computing Today and PeCS

- Ubiquitous and pervasive
- Immense scale and growing

- But,
  - data flows all around us (and we have to keep track of it)
  - we have to use an ever larger array of apps (lots of details)
  - interactions are complex (debugging is hard)
  - not all the mash-ups we want are there (do it manually)

Labscape (back to 2004)

- Experiment capture in a cell biology laboratory
- Cell biologists run several experiments over days
- Document their steps so there is a complete record
- Make the record shareable with other scientists
- Automatically mine set of experiments for patterns



[illegible]	105µl
2.5mM dNTP	87.5µl
20µM P1	87.5µl
20µM P2	87.5µl
Tag DNA Polymerase	7µl
H2O	1130.5µl

mix well & pipetted

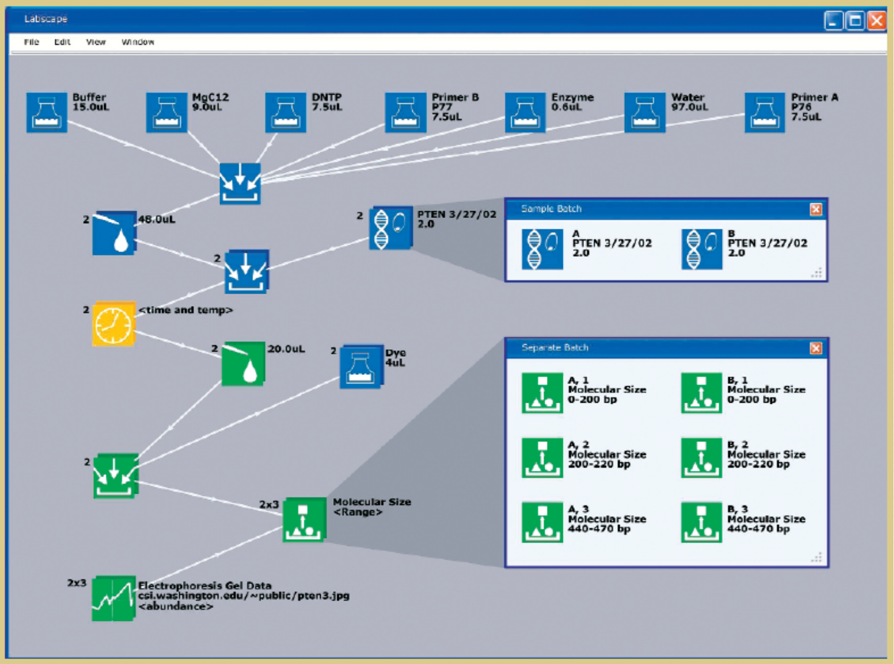
2µl DNA : 41µl PCR mix

2. Heat cycle : (method 55)
1. 94°C 5min
2. 95°C 30sec; 68°C 30sec; 72°C 1.5min
   × 35 cycles
3. 72°C 10min
4. 4°C Hold

• 8/31/01 4:40:58 PM • 2 sec @ f2.5

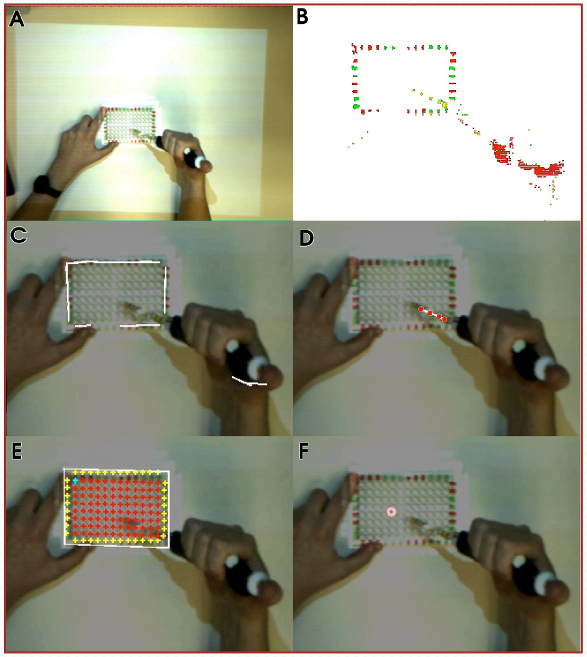
Organize experiment

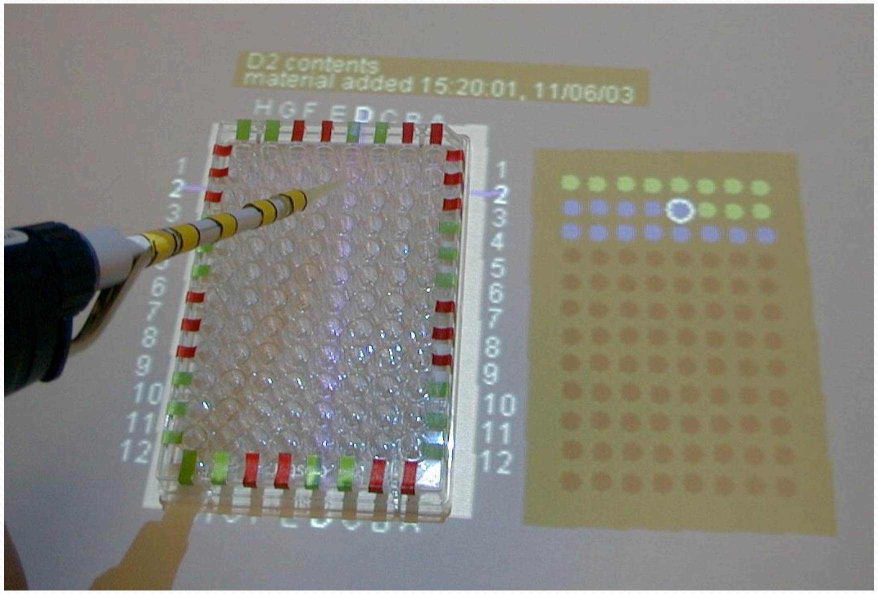
- Biologists generate plans that can guide data capture
- Serves as the interface to and template for collected data
- Automatic capture of interactions and settings
- Organization into a database
- Find data when needed at the bench
- Ability to search and correlate
- Automatic preparation of documentation



Tray and pipette interactions

- Instrumented pipette
- Bench with camera and projector
- Tangible user interface





We called it Invisible Computing

- Cognitive invisibility – not visual invisibility
- Many specialized hardware elements
  - Infrared proximity badges
  - Touch screens at benches supporting automatic migration of user interface with user
  - Bar code scanners
  - RFID readers
  - Audio annotation
  - Instrumentation interfaces

Labscape Lessons

- Redundancy to deal with noisy sensing
- Explicit representation of uncertainty
- Metadata to document data provenance and context
- Framework in which to organize data (experiment plan)
- Cognitive invisibility and high availability

Challenges for PeCS

- Our technology requires too much interaction
- Mental models of apps and data are difficult to grasp
- Information design
- We get trapped by steep learning curves
- Industry designs for stickiness and brand experience
- Devices and apps don’t get better with time
- Cycles and bandwidth to spare – put them to good use

Radar O'Reilly of M*A*S*H

- The ubiquitous clerk
- Who always knows the next step
- Understands our patterns
- Is always ready and available even before we realize we need him

Simple Examples

- Apps should learn patterns of use and get things ready – e.g., traffic/bus data for the trip to/from home
- Homes that make us conscious of our energy use and schedule tasks – e.g., delay dishwashers/dryers
- Personal devices that know about each other and update their data stores to match (including a personal cloud)
- Focus on data flows and how they combine (mash-ups) rather than on a specific app

Watching video content

- Each device should know about the others in the room
  - large screen, mobile phone, pad
- Each records what the others do
  - videos watched, search terms used, reactions
- Each learns from the every session
  - usual “channels” and “programs”, continuations, etc.
- Try to anticipate
  - preload likely content onto devices, propagate links, ready likely choices

Patterns

- Our lives are more predictable than we think
- Our technology needs to learn those patterns
- And anticipate what we are likely to do
- Suggest actions and ask for reinforcement
- Eventually just does it but stays alert for exceptions

- Why should we ever do things more than once?

A Manifesto for PeCS

- Open data formats
- Focus on service composition
  - pipes, filters, forks, and mash-ups over monolithic apps
- Learn usage patterns
- Dialog with users
- Reinforcement (positive and negative) learning
- Anticipate the user and get better at it over time

Thank you

- for this opportunity
- and I look forward to your questions