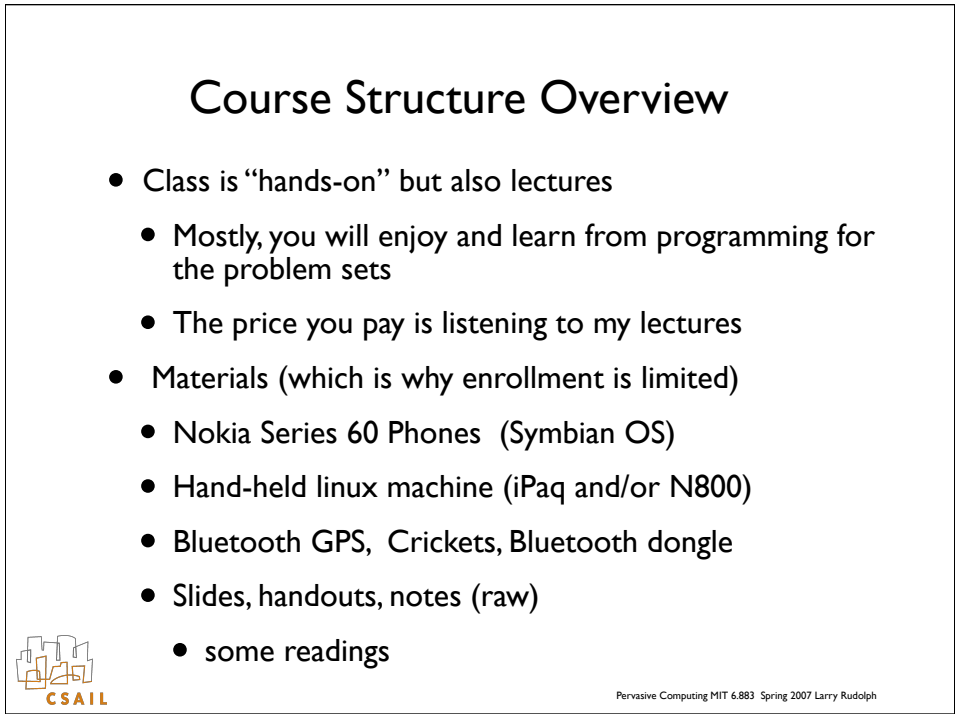
Lecture 1
Introduction

Pervasive & Mobile Computing
MIT 6.883

Larry Rudolph (MIT)

Massachusetts
Institute of
Technology



Here are some notes. I will try to annotate each slide. In general, slides are simply outlines and reminders for what I want to say. It would be great if there were a textbook that went along with the class, since some people remember things better by reading than by hearing. These annotations are an attempt to address this deficiency in the class.

The course is unique, timely, and continues to evolve. The field of pervasive computing is ill-defined. Even the name is ill-defined. Some people call it ubiquitous computing, others mobile computing. Still others, may look at the course content and call it a user interface course. It is a set of material that may be important as cell phones, hand-held, and embedded devices continue to evolve. The best way to understand the current state of the art is by trying to build some prototype applications. To this end, the class will provide the student with the opportunity.

This is a picture of my desk and some of the equipment I plan to use in the class. The good news is that
there is a lot of stuff, the bad news is that there is not enough. I have about 15 phones all of which are
programmable. Eight of them are identical and the best ones for the class. A few are older ones that have
less memory. Some are newer, fancier ones, but as you will find out, have more annoying built-in security
making it a bit harder to program. I also have about a dozen linux iPaq's. These are getting old but
are still useful, especially with the cricket indoor location beacons and listeners. There are a half-dozen or so
bluetooth gps receivers and a whole bunch of bluetooth dongles for use in your pc/ laptops. Finally, there are
a few N800 Linux tablet's but I am hoping to get more.

Administration

- Official Web Site
  - web.mit.edu/6.883
    - (http://people.csail.mit.edu/rudolph/Teaching/home883.html)
- Official Wiki
  - Last year's site: http://org.csail.mit.edu/mode
  - A new twiki will be setup and visible by the world and people will come to view it.
- Grade: 30% problem sets, 30% quiz, 30% project, 10% participation

Massachusetts Institute of Technology

The web site will have the schedule, problem set announcements, lecture slides and notes, and any readings.
The wiki is for the class use and is the main way that we can share information. Please note that the class will
be followed by people from all over the world. I expect students to post questions, answers, and interesting
papers, news stories, applications, etc.

The good, the bad, the ugly

- The course should be fun because
  - you get to program cell phones get a glimpse of the future
- The course should be challenging because
  - if covers a large range of topics and you may have to discover a lot by yourself
- It should be frustrating because there is not enough support (welcome to the real world)

The point is that this course is not for everyone. Unlike many other courses, you will not be spoon-fed. There will
be lots of frustration when something does not work as specified. You will have to rely on each other. Moreover,
there is no TA for the course and so many times you are on your own. On the other-hand, I will be interacting with
you more than faculty usually interact with students. On the third hand, along with the freedom comes not only
responsibility but also risk-taking. Assignments will often be under-specified. I may not always have as much time to
devote to the course as I would like.

Some student's excel in this environment. Think of it more as a research group rather than a standard course that has
been well honed over the years.

Problem Sets

- Preliminary Ideas:
  - Analyze a data set that contains cell towers and the gps coordinates where my phone "heard" the tower
  - geographically distributed "race"
  - 2-d boggle
  - location-aware lying
  - parallel search over the brains of your friends
  - guided tour of campus
  - conference kiosk support

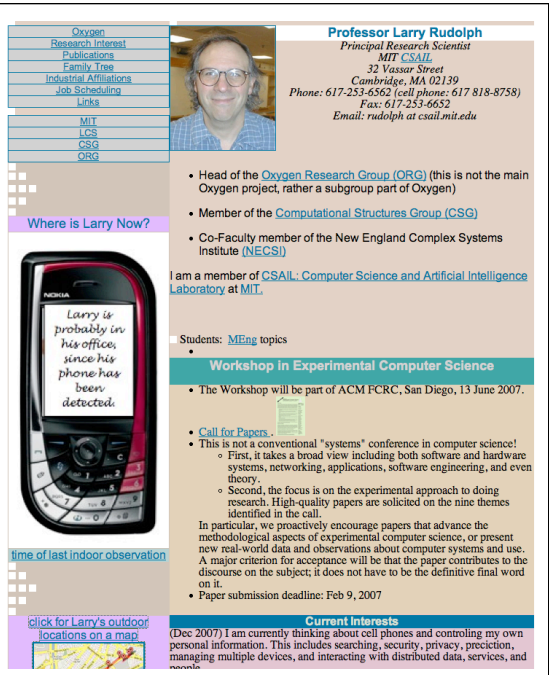
Massachusetts Institute of Technology

The problem sets are not yet set in stone. Moreover, I am trying to get students from other universities to participate in
some of the problem sets. Will let you know if and when this will happen. Do you have opinions? We will also be working with the MIT museum. They are building a campus tour guide so that when one visits various locations on campus, one can get pictures, video's, audio, and other archival information about what happened at that location itself. I believe that students will appreciate developing code that may actually be used.

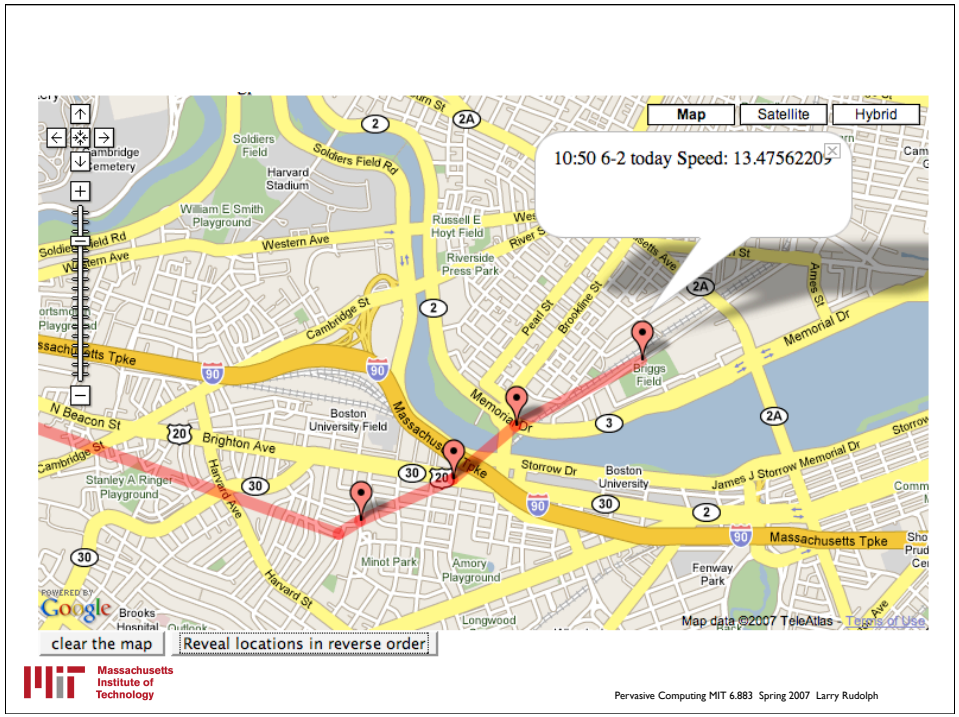
Where to find me?

- I track my indoor and outdoor locations.
- My website lists some of these

http://people.csail.mit.edu/rudolph



As we say in the business: "It is important to eat your own dog food." Now, I have a dog and I have been known to
munch on his treats every now and then. They are not so bad, kinda dry. I do not know the origin of this saying, but, I have been tracking my indoor and outdoor whereabouts for the past year. I carry around a bluetooth gps receiver and a spare phone. My phone talks to my server periodically (that is, when it does not crash). My server then displays this information in various ways.
The data I have collected can be used in some of the assignments.



Just to give you a taste of what I have been doing. This used to be novel, but now the technology is easy.
However, we will try to do some new interesting things with location.

Organization of material

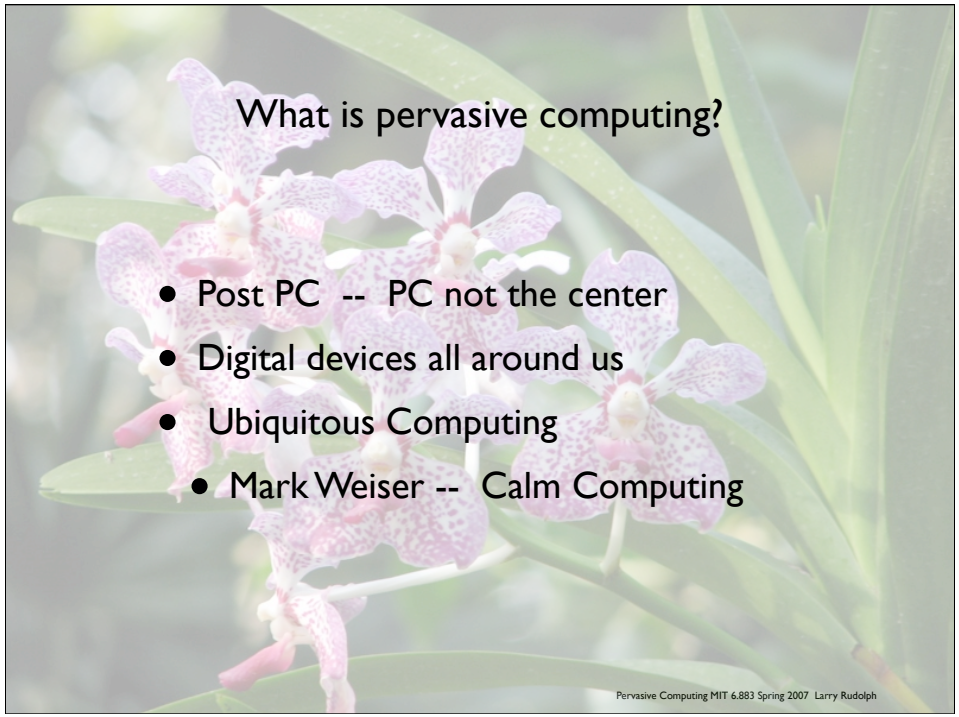
- Top-down
  - would be nice to start writing apps
  - but we are not there yet
- Bottom-up
  - Build on what is known
    - Keyboard, mouse, pen
    - Location, Speech, Multimodal
    - Integrative Technologies

Monday	Tuesday	Wednesday	Thursday	Friday
Feb 6 *(Reg Day)*	Feb 7 **L1: Overview**	Feb 7	Feb 8 **L2: Mobile Phones**	Feb 9
Feb 12	Feb 13 **L3: Python, Connectivity**	Feb 14	Feb 15 **L4: Socket Programming & Bluetooth**	Feb 16
Feb 19 President's Day	Feb 20 Monday Schedule	Feb 21	Feb 22 **L5: Location -- GPS**	Feb 23
Feb 26	Feb 27 **L6: Location -- Cell Towers**	Feb 28	March 1 **L7: Location -- Cricket**	March 2
March 5	March 6 **L8: C++ on Series 60**	March 7	March 8 **L9: User Interface, GUIs**	March 9 (add date)
March 12	March 13 **L10: Flash Programming**	March 14	March 15 **L11: GUI's Keyboards**	March 16
March 17	March 18 **XWand & Arrays**	March 19	March 20 **QUIZ**	March 21
March 26 **SPRING BREAK**	March 27 **SPRING BREAK**	March 28 **SPRING BREAK**	March 29 **SPRING BREAK**	March 30 **SPRING BREAK**

The course outline is preliminary but should give a feel for the subjects and topics to be covered. Many of these topics require an entire semester to do justice. The point of this course is to give an overview of some of the capabilities of mobile computing. The goal is that you, as students, will integrate this material in some new and interesting fashion. Too often,
when one goes deeply into a subject, one cannot integrate. You cannot see the forest for the trees. In this course, we will be flying over the forest in the hopes of seeing the whole landscape. Let us hope we have enough fuel to get us to the other side rather than crash landing in the middle somewhere. Unlike many of the world's forests, ours is healthy and rapidly expanding.

March 26 **SPRING BREAK**	March 27 **SPRING BREAK**	March 28 **SPRING BREAK**	March 29 **SPRING BREAK**	March 30 **SPRING BREAK**
April 2	April 3 **L12**	April 4	April 5 **L13: Speech**	April 6
April 9	April 10 **L14: Speech II**	April 11	April 12 **L15: Vision**	April 13
April 16 (no class -- patriot's day)	April 17 (no class -- patriot's day)	April 18	April 19 **L16: Kiosk**	April 20
April 23	April 24 **L17: Sketching**	April 25)	April 26 (drop dat*)* **L18: Security** PUFs	April 27
Apr 30	May 1 **L19: Debugging**	May 2	May 3 **L20: Publish-Subscribe & Middleware**	May 4
May 7	May 8 **Project Presentations**	May 10	May 11 **Project Presentations**	May 12
May 14	May 15 **Fun**	May 16	May 17*(Last day of classes)****Fun***	

The goal is to finish up early before the crunch of other courses.



There are two terms, “ubiquitous” and “pervasive” computing that are both used to describe roughly the same thing. According to dictionary.com, ubiquitous means Being or seeming to be everywhere at the same time; omnipresent: “plodded through the shadows fruitlessly like an ubiquitous spook” (Joseph Heller). The term pervasive adj 1: spread throughout; "a pervasive anxiety overshadows the triumphs of individuals" [syn: pervading] 2: spreading throughout; "armed with permeative irony...he punctures affectations"; "the pervasive odor of garlic"; "an error is pervasive if it is material to more than one conclusion" [syn: permeant, permeating, permeative]

So neither term carries much meaning. We are talking about the what will the world be like when we graduate from the PC on the desk.

The origin of the course: Project Oxygen

To bring an abundance of computation & communication
within easy reach of humans
through natural perceptual interfaces of speech and vision
so computation blends into peoples' lives
enabling them to easily do tasks they want to do:
- ***collaborate,***
- ***access knowledge,***
- ***automate routine tasks***

CSAIL

Pervasive, Human-Centric Computing

What do these words mean?

- Computers are already pervasive
  - even in Boston
- Computers are already human-centric
  - are they for the birds?
- It's not really about computing
  - we already know how to do that

Computers are pervasive. One cannot go mile without coming in contact with one. They have been pervasive for sometime. It is interesting that the household penetration of computers has not increased very much recently. Households with a computer may now have several of them, but not that many households that never had a computer get one. But something has changed in computing. The internet is now firmly established, but it does not end there. Computers are encroaching into the very fabric of our lives.

So, what do we mean?

- Pervasive
  - Should be where we need them
    - not have to go to them or set them up
- Human-centric
  - Computers should adapt to humans
    - computation enters our world/environment
- Computing
  - Computer-mediated function
    - digital media

Look back to see ahead

- Monolithic Programs & Hardware
- Decompose into interactive pieces
  - Compose to build large thing
- Continue decomposing into autonomous, interacting components

Finding and naming stuff

- Few items
  - Use list
- Many items
  - Use hierarchy
- Very many items
  - Use multi-index

Linux on Handheld

- Why Linux?
  - Linux allows full access to all software
  - Common development with desktop
  - Can use open source code from many sources
- Porting Linux to a handheld device
  - More difficult than standard PC or Laptop
    - Non-standard interfaces (screen, control FPGAs, touch screen, …)
    - Requires rewritable Flash ROMs

Linux Handheld Devices

- Linux phones are coming (here already)
  - we only care if linux is exposed to user
  - OpenMoko -- open source linux phone
    - why is this important?
- Lots of other devices. Some alive some gone.
  - Add any that you find to our wiki

HP iPAQ 3870

3870 iPAQ

– 206 MHz Strong Arm
– 64 Mbytes SDRAM
– 32 Mbytes flash storage
– Bluetooth
– SD/MMC card slot
– 16 bit color display

5500 iPAQ

– 400 MHz Xscale
– 128 Mbytes SDRAM
–48 Mbytes flash storage
– Bluetooth & WiFi
– SD/MMC card slot
– 16 bit color display

Nokia N800 Internet Tablet

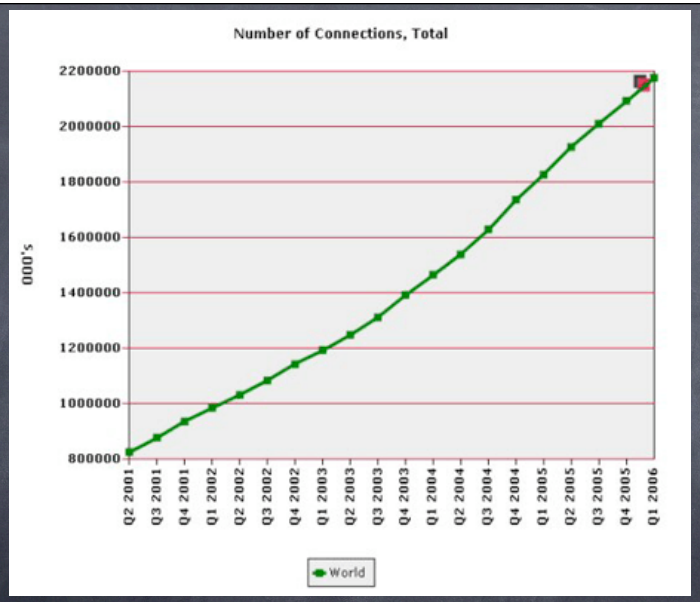
CPU: 330 MHz TI OMAP 2420
OS: Linux (Maemo 3.0)
Connectivity: WiFi/Bluetooth (including Bluetooth DUN)
ROM: 256M Flash
RAM: 128M RAM
Hard Disk: None (internal SD up to 4GB)
Display: 800x480 LCD touchscreen, 4.1" diag.
Interface: Dual SD cards, USB, Earphones, microphone, power socket, retractable webcam
Keys: Power, 5D navigation, Home, Escape, Menu, zoom in, zoom out, fullscreen
Battery: 1500 mAh rechargable
GPS: None (Optional Bluetooth with Navicore software due Spring 2007)
Size: 144x75x13 mm
Weight: 206 grams

Massachusetts Institute of Technology

A favorite logic puzzle of mine is a story about the 40 unfaithful wives. Let's update the story to be a bit more sensitive to others. Our new version is called the 40 infected cows. There was a poor village where lucky families may own a cow, but no one had more than one cow. Now if an own discovered that his or her cow was infected, the cow must be slaughtered in the town center at dawn the next morning so that anyone could see. But only the owner could slaughter the cow and no one would tell the own if their cow was infected. It turns out, that owners never see that their own cow is infected but can see others. Everyone in the town knew there were infected cows. They may see all 40 infected cows or 39 if their own cow was infected.

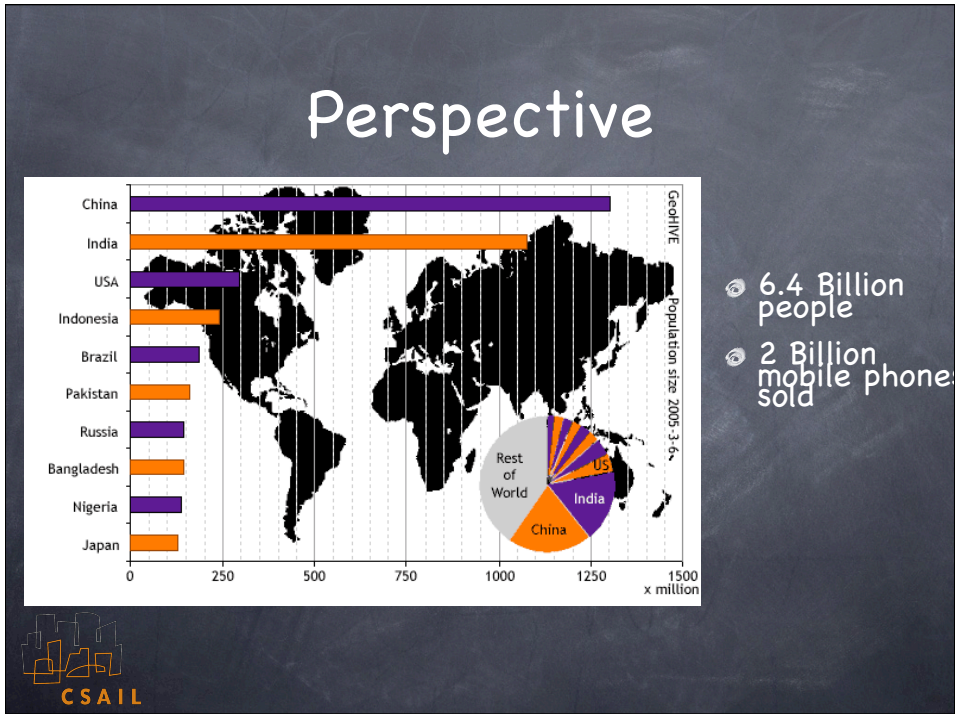
The situation stayed that way until one day, a person from the world health organization, WHO, came to the village, looked around and then announced to the whole village "There are infected cow's here, please abide by the rules" and then leaves. Nothing happens for sometime. But 40 days later, at dawn, all 40 cows are slaughtered. Why is that?
It is not difficult to tell to figure out. What is difficult, is to understand what information did the WHO provide to the village? The proclamation did not tell anyone anything that they did not already know.

I feel like I am in a similar situation, standing here telling you something that you already know. "There are a lot of mobile or cell phones" or jokingly, "The is an wide-spread infection." The number of



CSAIL

- September 18, 2005 -- 2 Billion connections.

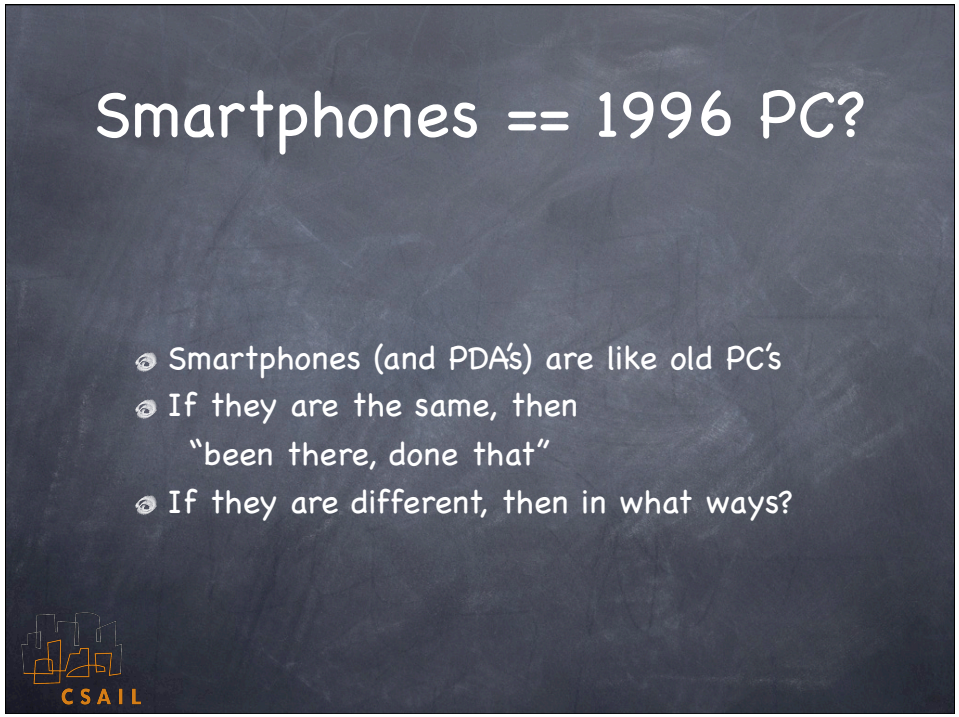


To put this number in perspective, there are 6.4 billion people in the world. That is one phone for every three people. This is even more incredible when one realizes that 2 billion people live on less than $2.00 per day. Of the remaining 4.5 billion people, if we assume a quarter are kids, then soon nearly every adult will have mobile phone. Yes, I know kids have phones but these numbers are incredible.

So what if everyone has a phone. Everyone has a watch and many of them are digital. We cannot program or even customize them. A large majority of mobile phones are simple appliances that can do little more than act as a phone. They cannot be programmed. Smartphones make up a very small fraction of the market.

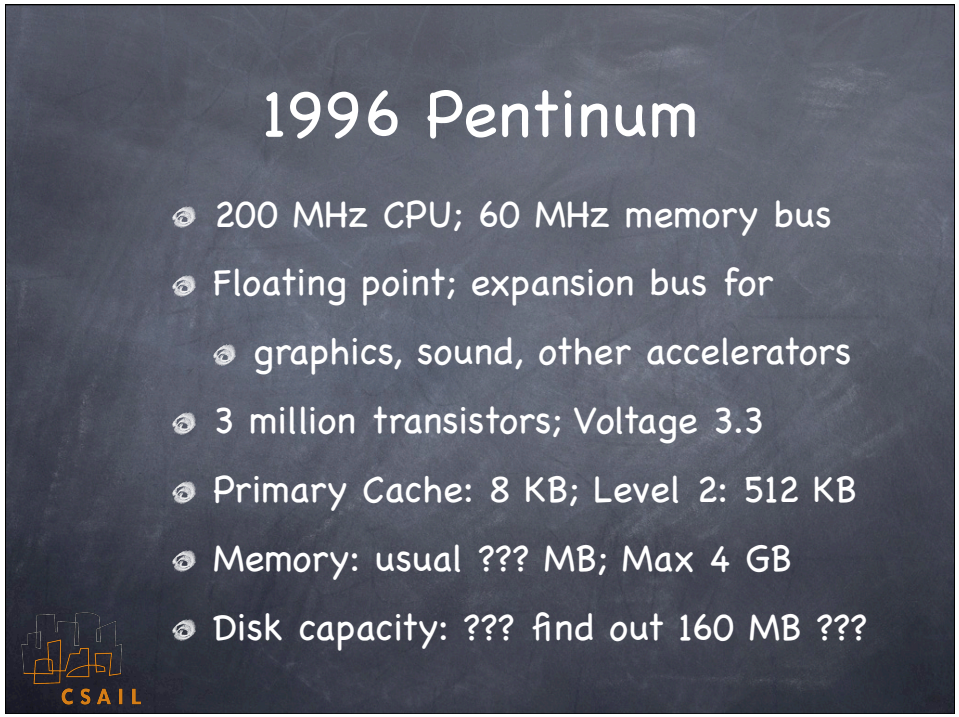
However, this will change. The economics of the semiconductor business is that the costs are all in development. The cost per chip is minor. In otherwords, soon, all phones will be smart. Of course there will be differences. There wil be basic, inexpensive phones. But I predict they will be equivalent to today's smart phones. The price of phones will be based on fashion and extra features. Perhaps a higher quality camera lens, better constructed, waterproof, or containing new sensors, such as GPS, heart monitor, or even built-in lie detectors.

We can see this trend happening already.

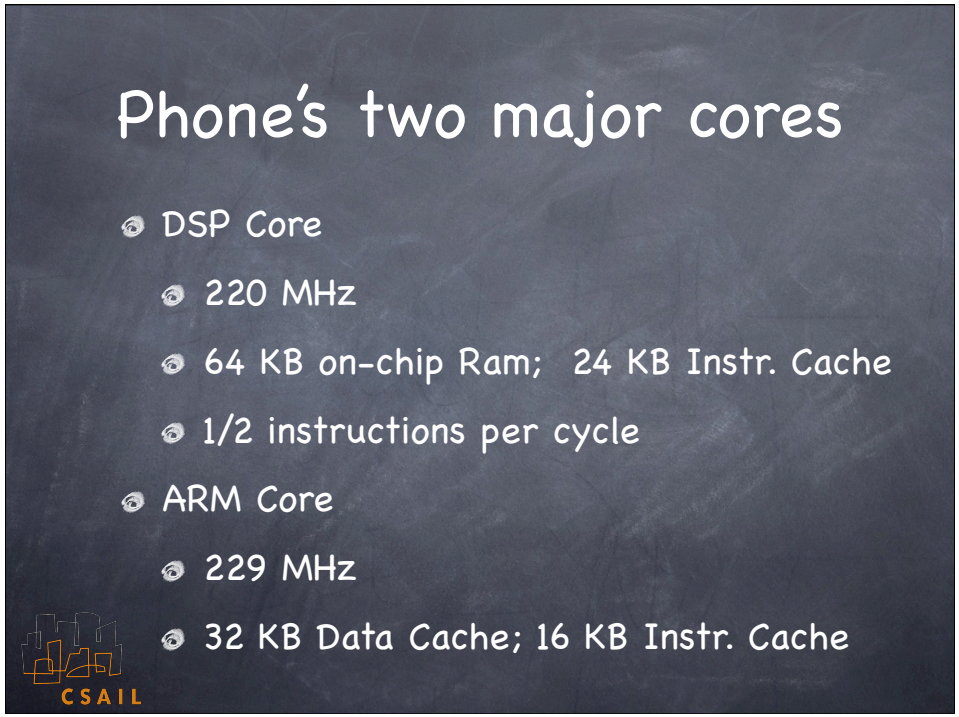


Suppose it does turn out that there are billions of smart phones, once again the question is "So what?" A smart phone looks pretty similar to a decade's old personal computer. OK, there maybe many more of them, but if they are the same then, as computer scientists, we have "Been there, done that"

On the otherhand, if they are different, then we have, potentially, lots of new challenges.

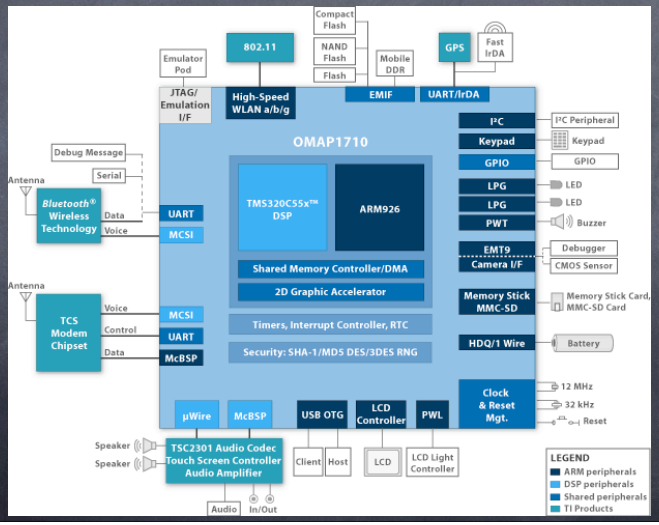


So, we have a 200 MHz processor, with a small primary cache and a 1/2 megabyte level two cache. Typically it has 64 MB but it can have upto 4 GB main memory

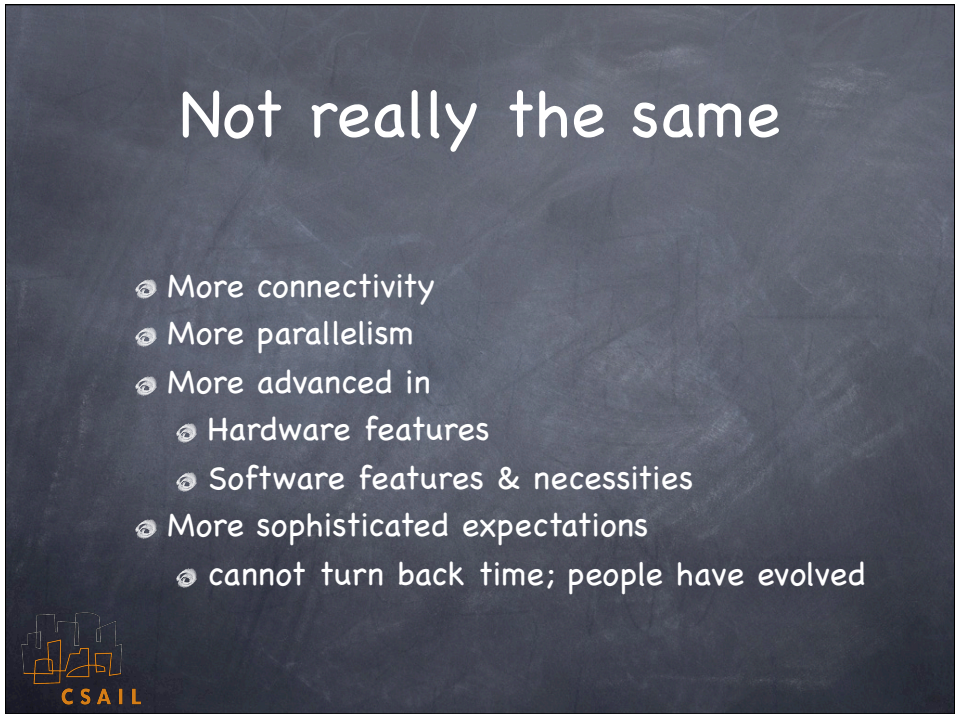


Phones come in all shapes, sizes, and configurations. Most smart phones have two major chips: a DSP and an ARM-based processor. Each are about the same speed as our Pentium, smaller memory, larger primary cache, but also about same power needs for the computation part. Obviously the radio takes up more power but the display less. What about for the disc?

Phone == Lots of Integration



CSAIL



Despite the similarities there is really a big difference. Phones have more ways to connect. There is some internet connectivity such as GPRS, 3-G, Edge, some local connectivity such as bluetooth, and of course the phone itself. But like the computation, there much of the connectivity cannot be simultaneously used. The PC would do lotso f context switching whereas the phone turns on and off different components.

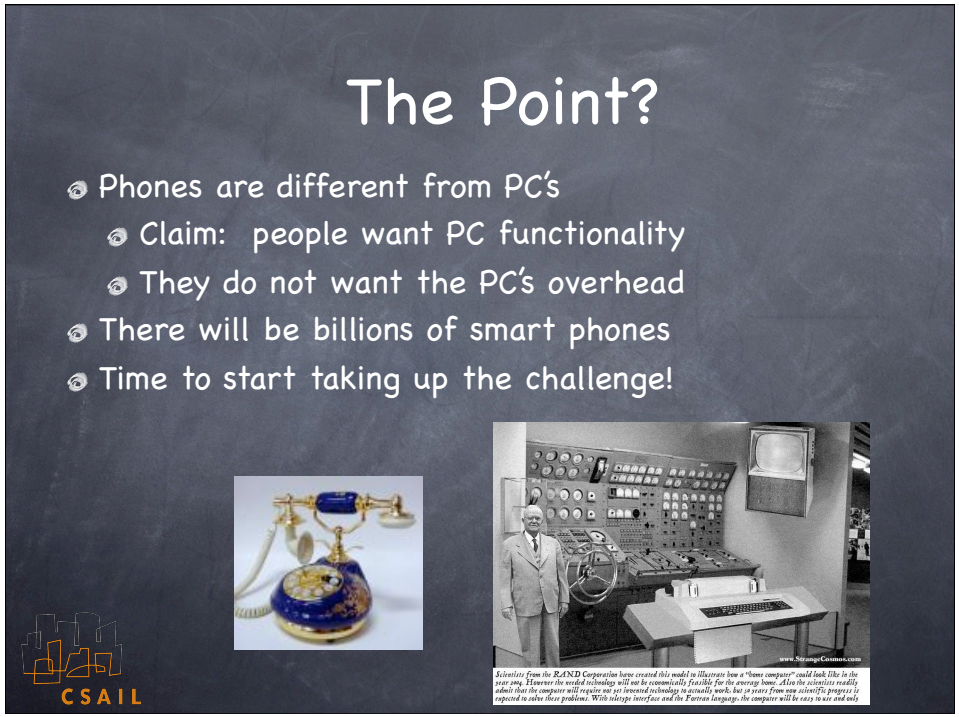
A big difference is in the level of integration. There is much more than in the PC. On the otherhand, it is not easy to add hardware. The only real way to do this is via wireless connectivity. Bluetooth works like a USB connection. The connection is easier but not as fast. There is a pci bus inside, but one cannot just plug in modules.

Another drawback is the inability to experiment. A 1996 Pentium machine was easy to add components and measure their results. More memory, sound card, graphics accelorator, larger screen, and so on. Much harder now.

Benchmarks, typical usage patterns? We are not there yet.

The main difference is that user's have changed. Even if we wanted to use the phone as a PC, users

Phones are not just little, port



We haven't been there or done that. Things are different with mobile phones. Some claim that the majority of people just want to use a phone as a phone. Afterall, if they wanted a computer they would have bought one. Computer penetration of households has not grown very much.

I claim that people simply did not want the overhead of PC's. The maintenance, decisions of buying printer, virus software, etc. is a lot of overhead. If, however, the benefits of computers come along fiarly inexpensively and more reliabily with phones, then of course they would want it.

Research Areas I

- User Interface (Huge)
  - Configuration
  - Syntax-free
  - Accessibility: physical & mental disabilities
- Security, Reliability, Fault Tolerance
  - Naive users; harsh physical world
- Synchronization & Sharing
  - Interoperability (no platform)

Research Areas II

- Architecture:
  - Phone chips as building blocks
    - wireless expansion bus (no other board)
  - Power & heat management
    - e.g. streaming video via DSP or ARM
    - local vs remote compute & store
  - No H/W upgrades

Research Areas III

- Applications
  - Services not applications; easier on user
  - Finding features (e.g. 287 menu items)
  - Platform independence (?)
    - same app for server; pc; phone
    - too many models (binary rewrite?)
- (location, user, env)-aware computing
- Phone as Sensor+Actuator Server
- Phone as (out-of-band) debugger

Conclusion

- Whatever your expertise, phones offer
  - different set of constraints
  - different levels of abstractions

If you think technology is frustrating today, just wait...