# Ubiquitous Computing

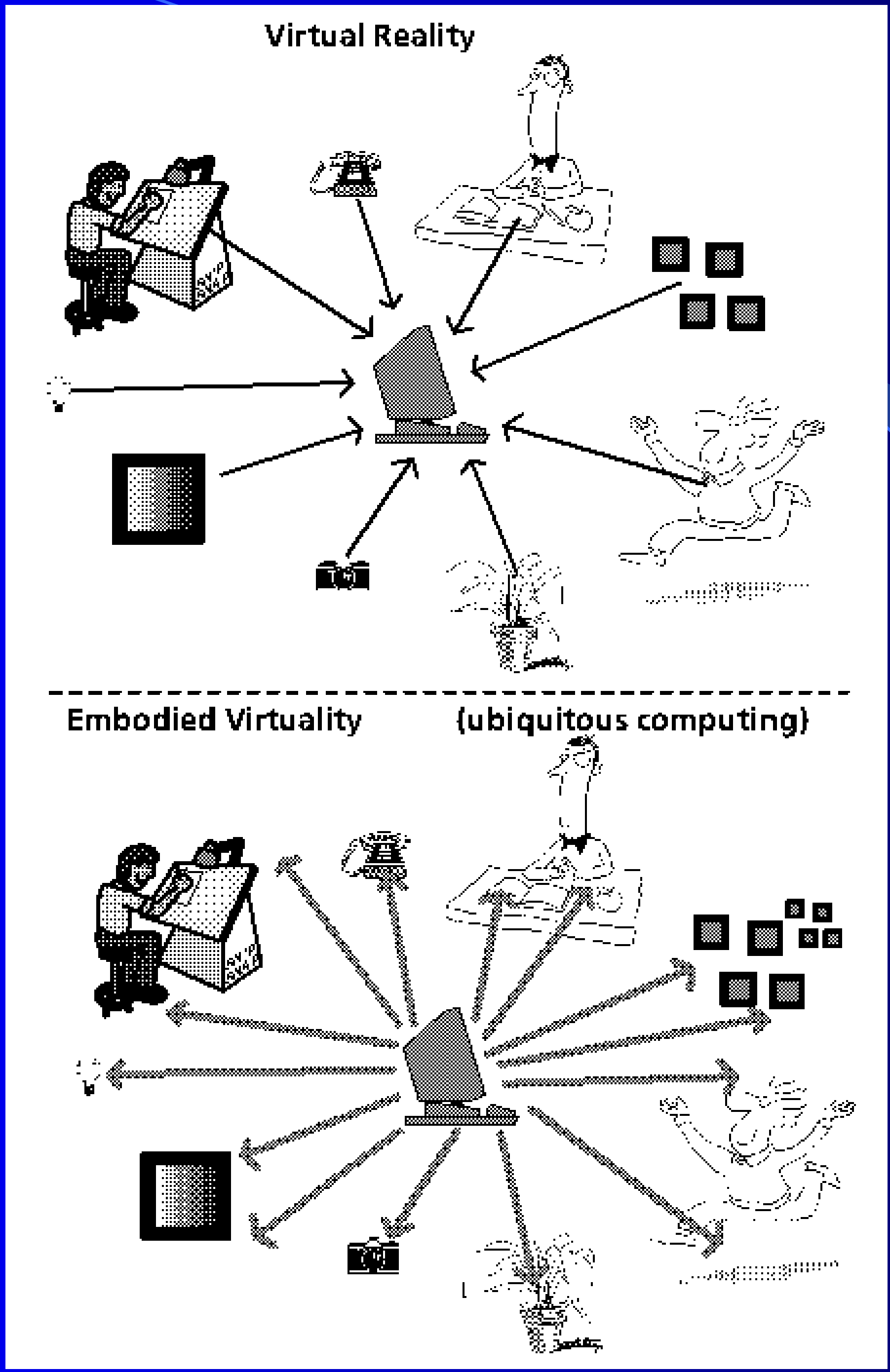
Ashley Green and Brad Rosen

Advanced Topics Informal Systems

Professor Silberschatz and Professor Yang

# Contents

- Introduction
- Design Challenges / Problems
- Discussion Points


Virtual Reality
Embodied Virtuality
(ubiquitous computing)

# Introduction

Computer today is isolated, so rather than being a tool through which we work, it becomes the main focus of attention.

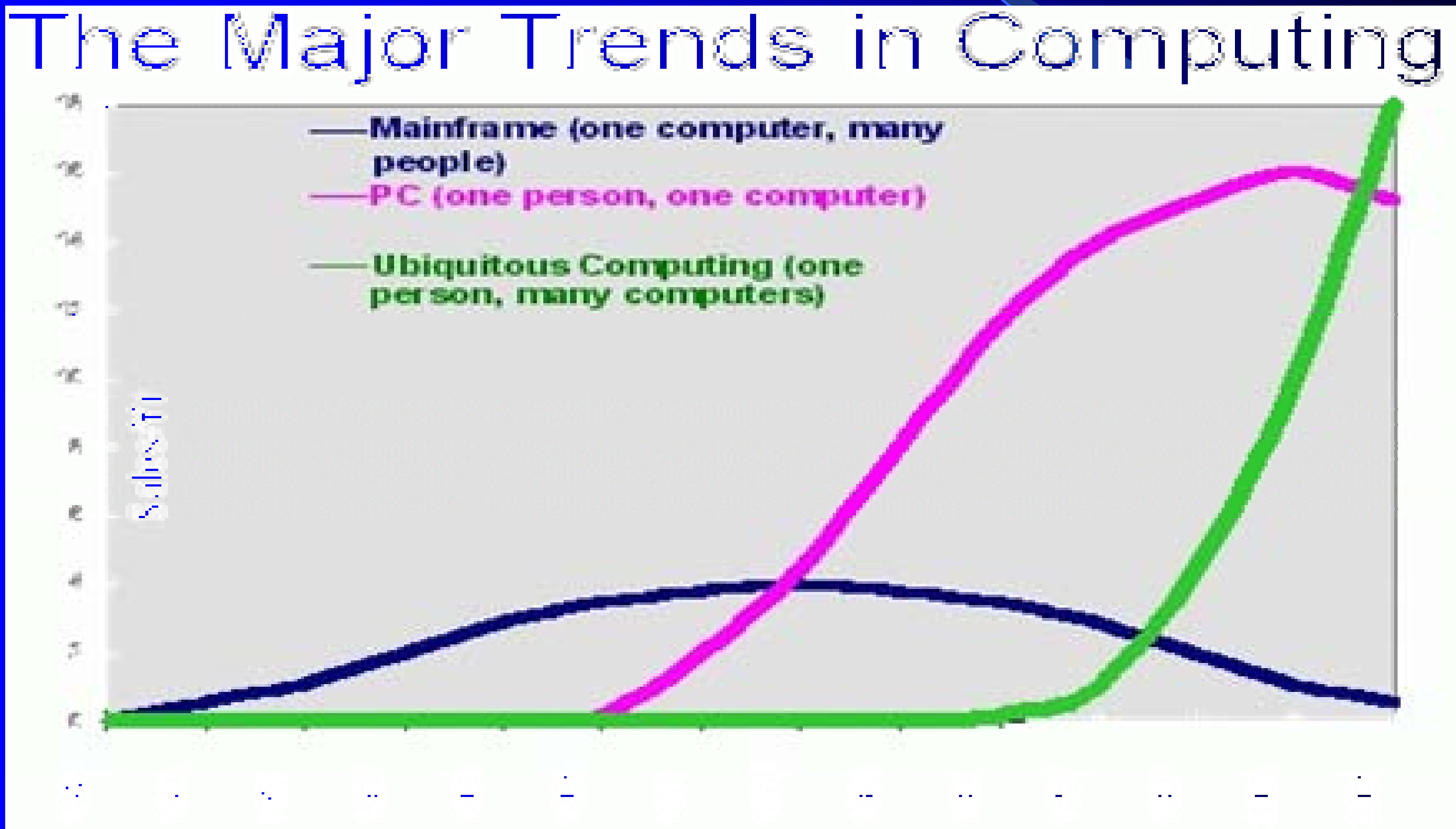
Resulting goal is to place the focus back on the user instead of on the computer: ubiquitous (invisible) computing

Definition: Method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user.

# Introduction

- Today:  You realize a block is too heavy to lift.  You whistle/call/motion for your super-heavy helper to assist you.
- Pervasive Computing:  You go to lift the block, and your invisible-computer-agent detects you are not strong enough to do so, and automatically assists you without you even *asking* for it.  [Perhaps not even **realizing**]
- ***Pervasive/Ubiquitous computing requires extreme AI.***

# Introduction



# Introduction

- Physical space with embedded computing/sensing power creates an heretofore unseen fusion.
- Example:  A corridor or room automatically adjusts heating, cooling and lighting levels based on the occupant's profile.
- "Smartness" may extend to individual objects [e.g. moldable handles that reshape themselves, cars that automatically adjust steering wheel and seat placement] regardless of the space they are in.
- "Smart" software.  [Simple example: "vacation/out of office" messages]

# Introduction

- INSERT GRAPHIC HERE: PG 11 of Satyanarayanan [pervasive computing vision/challenges]

# Introduction

- Another aspect of “ubicomp” is non-VR-immersion: the idea that with ubiquitous computing present in the environment itself, the way we utilize and interact with our surroundings will better reflect that capability.
- [Xerox PARC implementation…]

# Introduction

Example Implementation: Boards, Pads and Tabs

- Board: wall-sized interactive surface
- Pads: Interactive surface with pen emphasis (ie. ScratchPad)
- Tabs: pressure sensitive screen, three buttons and ability to sense position in an environment

Idea: have one or two boards, many pads and hundreds of tabs in an environment (home, office, classroom, etc.) that interact with each other to adapt to and serve the user.

# Introduction

As seen with the board/pads/tabs, ubiquitous computing should eradicate the relationship between a computer and a user.

Ideally, intelligent devices will become increasingly pervasive to form “smart environments”, wherein personalized devices interact with users, sense their environment and communicate with each other.

In order to do this nodes must:

1. self-organize themselves into ad hoc networks
2. divide the task of monitoring among themselves
3. perform tasks in an energy-efficient manner
4. adapt sensing quality only to the available resources
5. reorganize upon failure or addition of nodes.

# Design Challenges / Problems

Before looking at the many design challenges, let’s look at a smaller view of these problems just the context of boards, pads and tabs.

1. Board
2. Pad
   a. have to balance communication, ram, multimedia and expansion ports
   b. pen emphasis
3. Tab
   a. size and power consumption (don’t want to change batteries every week b/c takes away the idea of being invisible)
   b. have to balance size, bandwidth, processing and memory.

# Design Challenges / Problems

1. **Tracking user intent**
2. Cyber Foraging / High-Level energy management
3. Networking Protocols
4. Spectrum
5. Scalability
6. Rules of Coexistance
7. Adaptation Strategy
8. Privacy (location and trust)
9. Masking Uneven Conditioning
10. Context Awareness / Proactivity v. Transparency

# Design Challenges / Problems

## Tracking User Intent

- must track user intent in order to determine which system actions will help rather than hinder the user.
- Question: can user intent be inferred or does it have to be explicitly provided? (implications for the idea of invisible computing)

# Design Challenges / Problems

1. Tracking user intent
2. **Cyber Foraging / High-Level energy management**
3. Networking Protocols
4. Spectrum
5. Scalability
6. Rules of Coexistance
7. Adaptation Strategy
8. Privacy (location and trust)
9. Masking Uneven Conditioning
10. Context Awareness / Proactivity v. Transparency

# Design Challenges / Problems
## Cyber Foraging & High Level Energy Management

Need to make mobile devices smaller, lighter and have longer battery life, but unfortunately computing capabilities will be compromised.

A consensus exists that advances in battery technology and low-power circuit design cannot, by themselves, reconcile these opposing constraints.

# Design Challenges / Problems
## Cyber Foraging & High Level Energy Management

Proposed solution: "cyber foraging"

dynamically augment the computing resources of a wireless mobile computer by exploiting wired hardware infrastructure ("temporary assistance")

Scenerio: *when a mobile computer enters a neighborhood, it first detects the presence of potential surrogates and negotiates their use. Communication with a surrogate is via short-range wireless peer-to-peer technology, with the surrogate serving as the mobile computer's networking gateway to the Internet.*

# Design Challenges / Problems
## Cyber Foraging & High Level Energy Management

Important Research Questions:

1. How does one discover the presence of surrogates?
2. How does one establish an appropriate level of trust in a surrogate?
3. How much advance notice does a surrogate need to act as an effective staging server with minimal delay?
4. What are the implications for scalability?
5. What is the system support needed to make surrogate use seamless and minimally intrusive for a user?

# Design Challenges / Problems

1. Tracking user intent
2. Cyber Foraging / High-Level energy management
3. **Networking Protocols**
4. Spectrum
5. Scalability
6. Rules of Coexistance
7. Adaptation Strategy
8. Privacy (location and trust)
9. Masking Uneven Conditioning
10. Context Awareness / Proactivity v. Transparency

# Design Challenges / Problems
## Networking Protocols

- In common computing, media methods are collision detection and token-passing (distributed systems ).
- These methods will not work in a wireless domain because not every device is assured of being able to hear every other device.
- MACA: two stations desiring to communicate first send a request-to-send followed by a clear-to-send. Requires stations whose packets collide to backoff at a random time and try again. This way one packet can dominate bandwidth in order to create fairer allocation of bandwidth.

# Design Challenges / Problems

1. Tracking user intent
2. Cyber Foraging / High-Level energy management
3. Networking Protocols
4. **Spectrum**
5. Scalability
6. Rules of Coexistance
7. Adaptation Strategy
8. Privacy (location and trust)
9. Masking Uneven Conditioning
10. Context Awareness / Proactivity v. Transparency

# Design Challenges / Problems
## Spectrum

Must be possible to begin transmission in a particular location without prior consent or licensing procedures

1. If the number of deployed devices is large, the overhead of a licensing process will be excessive
2. Since some devices will be mobile, it's not efficient to give this device exclusive rights to spectrum at every location it might reside.

Consequently, additional allocations with appropriate rules will be needed to support wide-scale deployment.

# Design Challenges / Problems
## Spectrum

Can you think of a possible solution?

Solution: impose constraints on how spectrum will be used with spectrum policies

1. all devices should have adequate quality of service
2. no devise starvation
3. policies should not inhibit innovation in this field.
4. Policies should not sig. increase device costs

# Design Challenges / Problems

1. Tracking user intent
2. Cyber Foraging / High-Level energy management
3. Networking Protocols
4. Spectrum
5. **Scalability**
6. Rules of Coexistance
7. Adaptation Strategy
8. Privacy (location and trust)
9. Masking Uneven Conditioning
10. Context Awareness / Proactivity v. Transparency

# Design Challenges / Problems
## Scalability

- Current analysis of scalability has ignored physical distance between devices.

  ie. a web server or file server should handle as many clients as possible, regardless of whether they are located next door or across the country.

- How is it different in ubiquitous computing?

  the density of interactions has to fall off as one moves away; otherwise both the user and his computing system will be overwhelmed by distant interactions that are of little importance.

Discussion: Bandwidth interactions, /. Effect, "Smart Slashdotting"
Implications of automatic throttling [nimda, code red, etc]

# Design Challenges / Problems

1. Tracking user intent
2. Cyber Foraging / High-Level energy management
3. Networking Protocols
4. Spectrum
5. Scalability
6. **Rules of Co-Existance**
7. Adaptation Strategy
8. Privacy (location and trust)
9. Masking Uneven Conditioning
10. Context Awareness / Proactivity v. Transparency

# Design Challenges / Problems
## Rules of Co-Existance

- Devices must coexist by sharing spectrum and possibly interoperate
- Two methods to support open access in smart environment:

  1. Create an unlicensed spectrum band where government allows any device to transmit without permission.
  2. The government licenses the spectrum to a band manager (a commercial band manager will have a financial interest in promoting both efficient use and innovation)

Methods to determine "rent" paid by devices to band managers should charge rent in proportion to the frequency at which devices transmit (ie. high-power devices that transmit often should be charged more)

# Design Challenges / Problems

1. Tracking user intent
2. Cyber Foraging / High-Level energy management
3. Networking Protocols
4. Spectrum
5. Scalability
6. Rules of Coexistance
7. **Adaptation Strategy**
8. Privacy (location and trust)
9. Masking Uneven Conditioning
10. Context Awareness / Proactivity v. Transparency

# Design Challenges / Problems
## Adaptation Strategy

- In a ubiquitous system, adaptation is necessary especially in allocation and use of resources (network bandwidth, energy, computing cycles, memory, etc.)

- Three strategies:

  1. Client guides applications to use less of a scarce resource
  2. Client can ask the environment to guarantee certain level of resource
  3. Client can suggest a corrective action to user.

# Design Challenges / Problems
## Adaptation Strategy

Provoking Questions:

1. Does solution three compromise the idea of invisible computing?
2. How does a client choose between adaptation strategies?
3. How will the implementation of a smart space honor resource reservations?
4. Is adaptation using corrective actions practically feasible?
5. What are the different ways in which fidelity can be lowered for a broad range of applications?

# Design Challenges / Problems

1. Tracking user intent
2. Cyber Foraging / High-Level energy management
3. Networking Protocols
4. Spectrum
5. Scalability
6. Rules of Coexistance
7. Adaptation Strategy
8. **Privacy (location and trust)**
9. Masking Uneven Conditioning
10. Context Awareness / Proactivity v. Transparency

# Design Challenges / Problems
## Privacy

- Two types of privacy issues: location and trust
- Location
    1. Don’t want to store location information in a centralized location because a hack would reveal all information
    2. Instead, store information about each person at a local PC or workstation.

( Still, there are serious consequences to accumulating information about individuals over long periods of time. Implication arises that there is never a purely technological solution to privacy, but by giving more power to individual helps move society towards a more private technological world.)

# Design Challenges / Problems
## Privacy

- Trust

    1. As a user becomes more dependent on a pervasive computing system, it becomes more knowledgeable about that user's movements, behavior patterns and habits.

    2. Information must be strictly controlled in order to protect it from being used in unsavory situations (ie. targeted spam or blackmail).

# Design Challenges / Problems
## Privacy

- Trust (continued)

  3. Greater reliance on infrastructure means a user must trust that infrastructure and the infrastructure needs to be confident of the user's identity and authorization level.

  4. Difficult challenge to establish trust between both the infrastructure and the user in a manner that is minimally intrusive and preserves the goal of ubiquitous computing: invisibility.

# Design Challenges / Problems

1. Tracking user intent
2. Cyber Foraging / High-Level energy management
3. Networking Protocols
4. Spectrum
5. Scalability
6. Rules of Coexistance
7. Adaptation Strategy
8. Privacy (location and trust)
9. **Masking Uneven Conditioning**
10. Context Awareness / Proactivity v. Transparency

# Design Challenges / Problems
## Masking Uneven Conditioning

- Uniform penetration is many years or decades away, therefore currently there will be differences in “smartness” of environments.
- Smart environments in conference rooms, offices or classrooms, might be more advanced than in other venues.
- This large dynamic range of “smartness” can distract the user and detract from the goal of making pervasive computing technology invisible.
- Possible solution:
  - personal computing spaces compensate for “dumb” environments, thus the user does not have be involved.
  - Discussion of tradeoffs: Client “Fatness” vs feature set

# Design Challenges / Problems

1. Tracking user intent
2. Cyber Foraging / High-Level energy management
3. Networking Protocols
4. Spectrum
5. Scalability
6. Rules of Coexistance
7. Adaptation Strategy
8. Privacy (location and trust)
9. Masking Uneven Conditioning
10. **Context Awareness / Proactivity v. Transparency**

# Design Challenges / Problems
## Context Awareness & Proactivity v. Transparency

- A pervasive computing system that strives to be minimally intrusive has to be context aware (aware of user's state and surroundings and modify its behavior based on this information)
- Key challenge is obtaining the information needed to function in a context-aware manner.
- Questions:
  1. How is context represented internally?
  2. How frequently does context information have to be consulted?

# Design Challenges / Problems
## Context Awareness & Proactivity v. Transparency

- Since ubiquitous computing strives to be context aware in order to adapt to the current state of the user, the system can be characterized as proactive.

- But, can proactivity deter invisibility of the system?
  (ie. the paper clip on Microsoft Word)

- Questions:
  1. How are the individual user preferences and tolerance specified and then taken into account?
  2. What cues can such a system use to determine if it is veering too far from balance?
  3. Can one provide systematic design guidelines to application designers to help in this task?

# Discussion

- From what we’ve seen of “ubicomp” so far, what is one of the primary needs to enable the system?
- Hint: Not processing power, storage, or bandwidth……

# Discussion

## Spatial Awareness - Smart “Location” Services

- “Spatial Aware” applications/devices know their position [either in a relative or absolute fashion] with respect to other agents and/or the environment itself.
- Much research is centered around “location services” themselves. [e.g. GPS]

# Discussion
## Nexus

- Open, Global Infrastructure [platform]
- Designed to Enable “Spatial-aware” applications
- Meant to be the “http” of location services/spatially aware devices/applications.
- GOAL: Be “middleware”

# Discussion
## Nexus Basic Premises

- Objects are bound to spatial locality
  - Posters on walls
  - Signs at street crossings
- Most current systems are map-based
  - Mapquest, et cetera
  - Can only make a left if there **is** a left…

# Discussion
## Nexus Basic Premises

- Support spatial-aware apps by representing physical areas as "virtual areas" – perhaps augmented by the additions of "virtual objects"
- Augmented Area: Geometric space + virtual objects [not a one-to-one mapping of virtual => real objects]
- Many Augmented Areas connected.

# Discussion
## Augmented Area Setup

- Every object represented by a data structure. [Possible weakness in scheme…]
- “Augmented Area Model(s)”
- Location-dependant [GPS or Active Badge]
- Automatic propagation of changes from the “model” to the area

# Discussion
## Augmented World

- Uniform descriptors for augmented area models
- Inter-area relationships can be ascertained
- Since all areas are Nexus-accessible, applications can switch contexts easily.
  - City Guide vs Museum guide for a tourist
- New areas just require “registration”
  - easy integration
  - “location manager” is implementation detail

# Discussion
## Examples

- Museum/City Tour guide
- Virtual Information Tower
  – Point at a person for info!
- Virtual control board
  – Control light, sound, doors, camera movements
  – Requires control signals from Model => Area

# Discussion
## Nexus Requirements

- Mobility
  - Laptops, PDA's, small mobile devices
- Heterogeneity
  - Bandwidth, storage, battery, Area sizes, object counts, new technologies
- Interoperability
  - Device, application and Area levels
- Scalability
  - Objects and users
- Privacy
  - Since information is stored/generated concerning location…

# Discussion
## How Does Nexus Fit In?

- UbiComp > Nexus
  - Provides interaction service for ubiquitous computing, information attachment/virtual objs
- Location/Context Aware Systems
  - Similar to Nexus, but only relies on localized info.
- Navigation Systems < Nexus
  - Nexus subsumes geographic modeling
- Augmented Reality
  - Superimpose more data: “see” power grids, etc

# Discussion
## Nexus: The Nitty Gritty…(isn’t pretty...)

- External Components
  - Clients: use standard interface [create/delete/link objects]
  - Sensor and control systems
- Basic Services
  - Communication/Adaptation: intra-nexus component and QoS notifications
- (Local) Data Management
  - Static vs mobile, virtual vs real. Separated by control-component and space. Areas sectioned off - differing Nexus nodes for storage.
- Distributed Data Management
  - Most complicated [next slide]
- Generic UI Support

# Discussion
## Nexus: Distributed Data Management

- Logically Centralized component set
  - Access local, distributed, and/or replicated nodes. [Somewhat like universal location]
- Location Management
  - Continuous mapping of objects and areas to the node where appropriate data is stored.
- Query and Event Service
  - Uses location management service, abstracts to provide an API for transparent access to higher level services.
- Model Management tools
  - Add/modify/delete linkages/objects, create views and layering
- Caching and Hoarding
  - Pre-fetch data, "hoarding" in advance [a priori knowledge of low or no bandwidth areas in the network]

# Discussion
## Nexus: Wrap Up & Future…

- Location Services
  - Easier with more copies, creates update problem
- Universal Hoarding
  - Know user behavior in advance
  - User Profiling
- Location-aware communications
  - Geographic multicast
  - Loose binding to model

# Discussion
## Nexus Concerns

- Unclear if appropriate data structures can be devised for **all** objects.
- Virtual mapping may not always be appropriate
  - Knobs, dials don't translate well.
  - Level of detail is vague [My alarm clock is 2" by 3"….]
- Data inundation!
- Right track, perhaps slightly **too** abstract – depends highly on the system trying to be instantiated.
- **Nexus vision strives to be independent of any other mobile/ubiquitous/pervasive vision, but in doing so, seems to sever its own justification for existence: Handoff might be better handled elsewhere.**

# Discussion
## Moving on…with some examples….

- AURA project [CMU]
- Scenario 1: Jane at Gate 23 in Pittsburgh
  - User Interaction: Dialog Box w/suggestion
  - Gate 15, distance, flight time….[culled from others]
- Scenario 2: Frantic Fred In his Office
  - Meeting cross campus
  - Voice editing en-route
  - Pre-emptive projector warming, private PDA warning
- Raises AI question of action-inaction or suggestion. [What if Fred’s aura had done nothing…]

# Discussion
## Extra Slides

Depending on time for discussion, more topics follow…

# Discussion
## Security In UbiComp

- Data warehousing [“Querying the Physical World”]
- Location-aware devices, face/voice recognition
- Forged authentication
- Trusted authentication services [raises its own privacy concern – who trusts verisign?]
- Wireless devices by default divulge some information: how to restrict
- System recovery from damaged/compromised devices: How does the floor trust the ceiling?

# Discussion
## U. Sensing/Storage…

- Ber5 todo jan19