Building Context-Aware Applications

Anind K. Dey
Senior Researcher, Intel Research Berkeley
Assistant Adjunct Professor, UC Berkeley
anind@intel-research.net

Transition from Hans' Talk

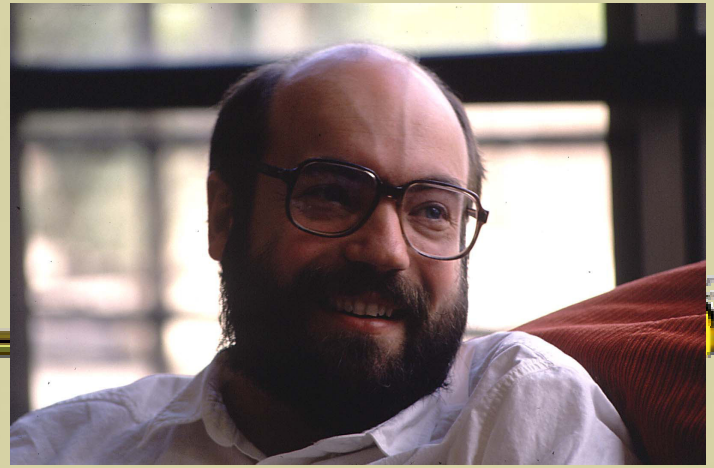
- Value of sensors in ubiquitous computing
- Good example of context-aware systems

➔ Fundamentals of building systems

Presentation Overview

- Motivation and What is Context
- Design Process – no support
- Design Process – with support
- Survey of Support
- Research Challenges

The Big Picture



- Mark Weiser: Chief Technologist at PARC
- Began Ubiquitous Computing Project in 1988
- Paradigm Shift

The Big Picture

- Ubiquitous computing
  - dynamic environments and new technologies
  - transparent/invisible/calm

The Big Picture

- "The most profound technologies are those that disappear."
- "The whole purpose for ubicomp, of course, is the applications."

One Slice of Ubiquitous Computing

- Automated Capture and Access
- Natural Interfaces
- Context-Aware Computing
  - Active Badges
  - PARCTabs

Motivating Example

- Context is fundamental:
  - Effective use of context is key for a an effective ubicomp environment
- Most common application?

Motivating Example

- Effective use of context is key for a an effective ubicomp environment

- Reminder to buy milk
- When to deliver: not time/location specific
- How to deliver: appropriate modality

What is Context?

What is Context?

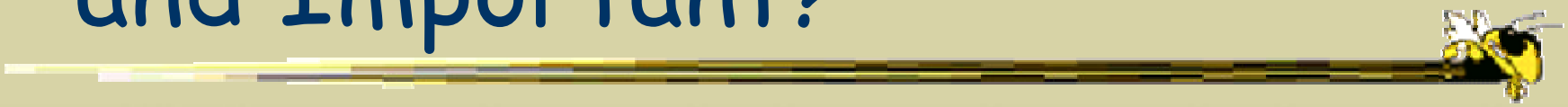
You, as end user or designer, decide:
Application dependent and situation specific

What is Context?

Application dependent and situation specific

- Typically: identity, location, time
  - Rest is implied
  - Not much else used: e.g. activity, mood

Why is Context Interesting and Important?



- Interesting
- Important
  - Explicit information already available to applications
  - Context is information, usually implicit, that applications do not have access to
  - Makes applications "smarter", increases communication bandwidth

What Do Applications Do With Context?

- What is a context-aware application?
  - App that uses context to perform some behavior/service for its user(s)
- 3 types of behaviors:
  - Display context
  - Automatically execute/adapt services
  - Tag captured information for easier retrieval
- Schilit94, Pascoe98

Displaying Context

- Directly display sensed context
- *e.g.* In/Out Board (Salber98), Location Maps (Want92), Status Displays (e.g. weather, activity)

Execute/Adapt Services

- Select and perform a relevant service from multiple services
- Change how a service behaves or is executed
- *e.g.* Print to nearest printer, Information Displays

Tag Information

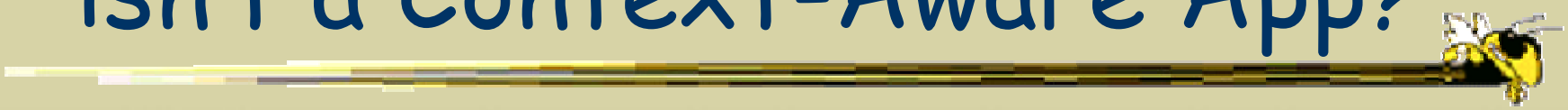
- Use context to aid in context-based retrieval
- May be easier to remember context than specific information

- *e.g.* Forget-me-not (Lamming94), Mobile Computing Fieldwork (Pascoe96), Conference Assistant (Dey99)

Canonical Applications

- Location Maps
- Information Displays
- Mobile Tour Guides
  - Display location
  - Present related information, create trip report when over
  - Retrieve selected information after tour

What isn't Context or What isn't a Context-Aware App?



- Explicit vs. implicit information
- Implicit tends to be more interesting

- ...

Presentation Overview

- Motivation and What is Context
- Design Process – no support
- Design Process – with support
- Survey of Support
- Research Challenges

Design Process – No Support

- Geared towards re-use and multiple applications

1. Specification
2. Acquisition and Representation
3. Delivery/Distribution
4. Reception and Storage
5. Action (the application)

Design Process - Specification

- Context to use
- Context behaviors to perform

- Key step in the design process: problem specification

Design Process - Acquisition

- Install relevant sensors
- Create or learn to use API
- Support ability to query or be notified of change in sensor data
- Store context
- Interpret/abstract context

Design Process – Delivery/Distribution

- Hans gave good overview
  - Issues of location/time
  - App/network-level delivery/routing models
  - Where to sense
- Transport mechanism
- Context typically captured remotely from application at different time
- Context captured in sensor-rich environment/device serving multiple applications
- Need to deliver/distribute context to multiple, remote apps

Design Process - Reception

- Application locates relevant sensors
- Requests context via queries, polls, notifications
- Additional interpretation/abstraction

Design Process - Action

- Combine received context with previous context for further analysis
- Perform an action(s) based on the analysis results

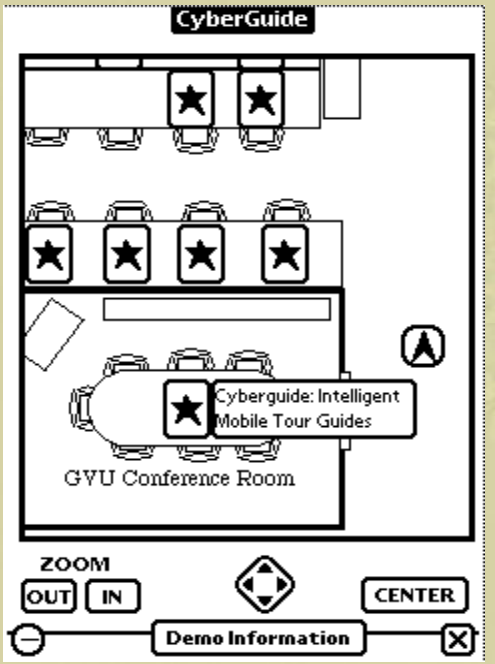
Using the Design Process: Simple Mobile Tour Guide

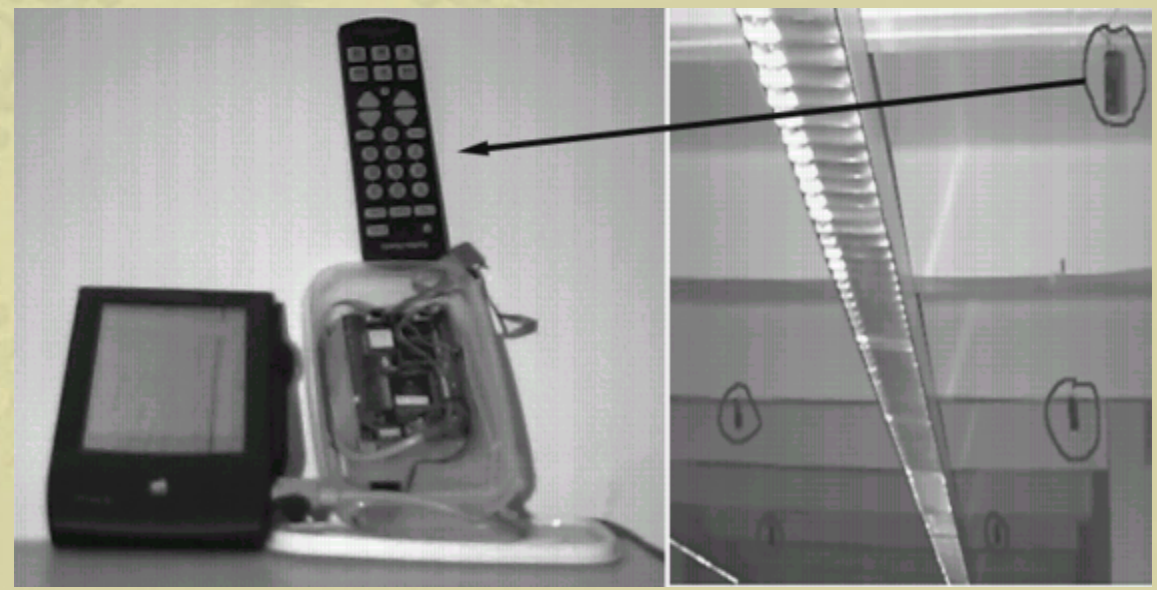
- Application:
  - Display list of all unseen demos
  - Highlight relevant demos
  - Show map centered on user's location
  - Display information about closest demo
  - Directions to demos

  - Single-user application

Simple Tour Guide

- Cyberguide





Simple Mobile Tour Guide

- Specification
  - Context:
    - List of demos visited
    - User location
    - User orientation
  - Behaviors:
    - 4 actions from previous slide

Simple Mobile Tour Guide

- Acquisition
  - Sensors:
    - Orientation: electronic compass
    - Location: Active Badge
    - List of demos: Active Badge and software
  - API: build for orientation and list, but learn Active Badge
  - Store data in local file
  - Interpretation of location to determine if at a demo

Simple Mobile Tour Guide

- Delivery
  - Only remote sensor is Active Badge
  - Write transport mechanism to support form of RPC to acquire data

Simple Mobile Tour Guide

- Reception
  - Application knows what sensors to use, but what if sensor-rich environment
  - Requests notification of changes in location and orientation
  - Additional interpretation to get list of demos

Simple Mobile Tour Guide

- Action
  - When application gets update about orientation, update map
  - When application gets update about location
    - Update map
    - Update lists of unseen demos
    - Highlight relevant demos

Group Exercise #1

- Choose a context-aware application, clearly indicating context and behaviors being used
- Develop/sketch/write a description of how to build it, assuming that there is no available support
- Make sure to address each of the 5 steps in the design process

Group Exercise #1: Review

- What was hard?
- What was easy?
- What kinds of applications/domains?

Presentation Overview

- Introductions
- Motivation and What is Context
- Design Process – no support
- Design Process – with support
- Survey of Support
- Research Challenges

Design Process – with support

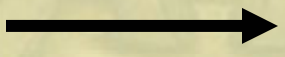
- What support is necessary?

Accidental vs. Essential Tasks

- Brooks 87 "No Silver Bullet: Essence and Accidents of Software Engineering"
- Accidental tasks
  - Problems induced by design tools
  - Common across applications
- Essential tasks
  - Inherent problems
  - Specific to application being built

Design Process Revisited

1. Specification
2. Acquisition
3. Delivery
4. Reception
5. Action



1. Specification
2. *Acquisition*
3. Action

Framework Requirements

- Context specification – support for specifying needed context
- Discovery – locating components that can acquire and act on context
- Separation of concerns – separate the acquisition of context from the use of context

Framework Requirements

- Storage – context history important
- Interpretation – abstract to higher level
- Transparent communications – simplify application development
- Constant availability – context needs to be acquired 24/7

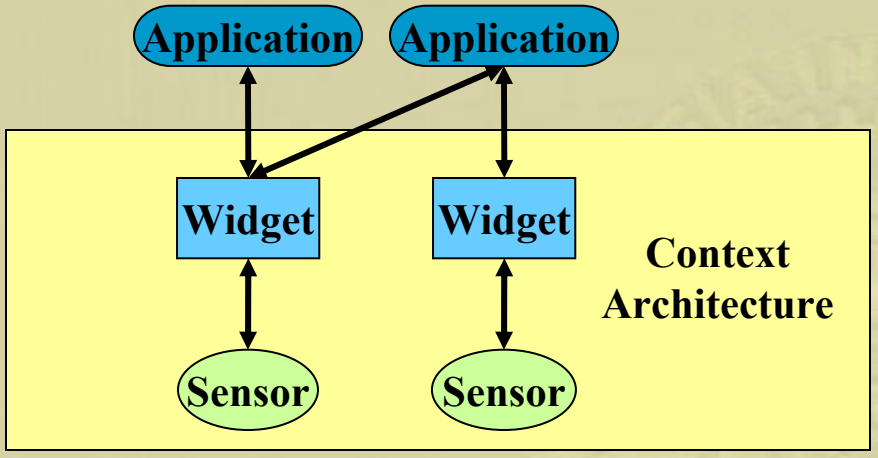
2 ways to view the world

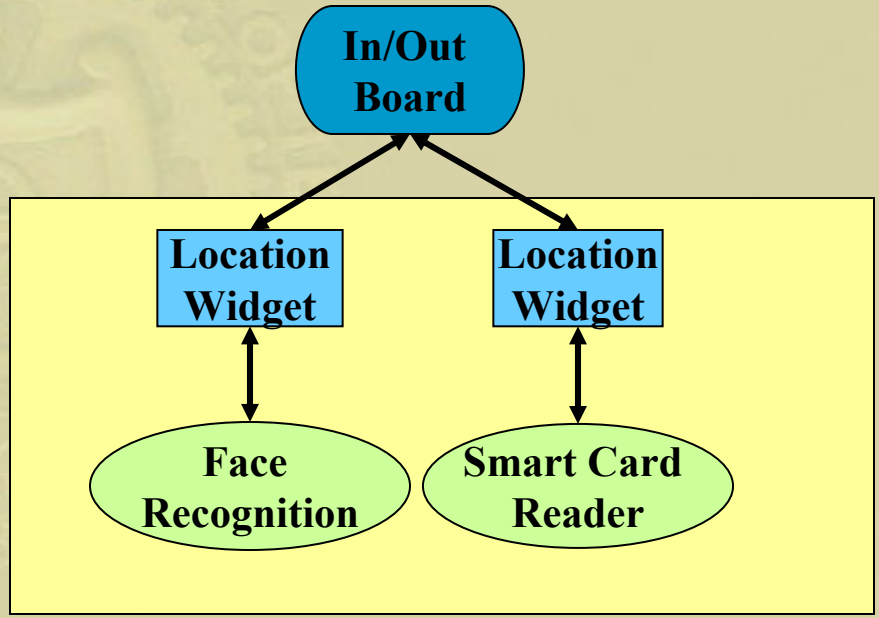
- From the field of artificial intelligence:
  - Component-based: components correspond to real-world entities and logical pieces of computation
  - Situation-based: treat the world like a collection of information and ask for information you're interested in

Building Blocks: Widgets

- Context widgets
  - Analogy to GUI widget
    - Separation, callbacks, attributes, encapsulation, abstraction
    - E.g. GUI button
  - Why: Responsible for acquiring and abstracting data from particular sensor, separation of concerns, storage

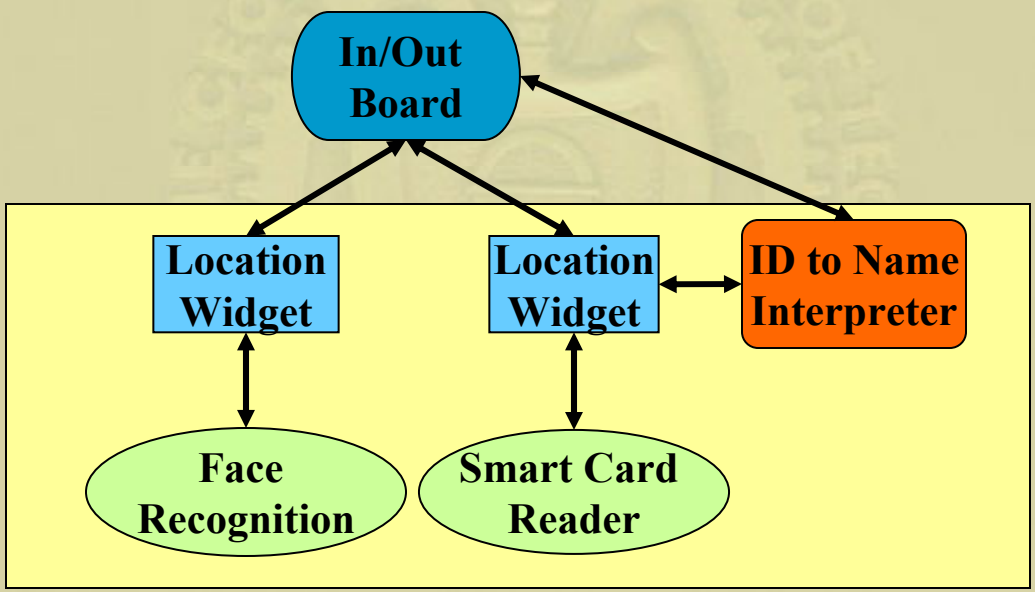
Building Blocks: Widgets





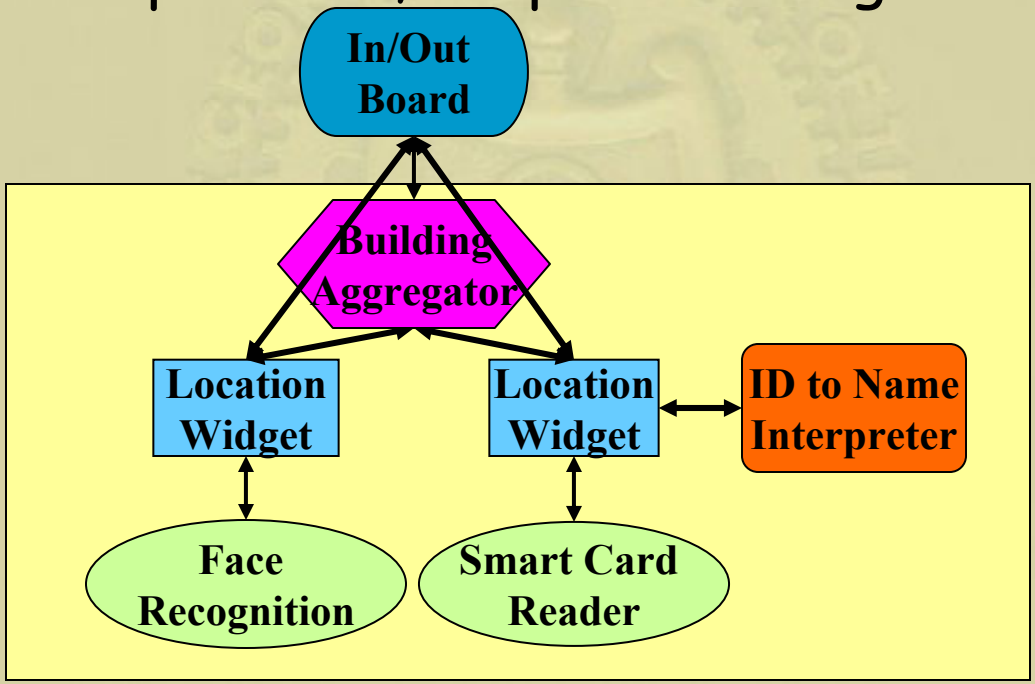
Building Blocks: Interpreters

- Context interpreters – convert or interpret context to higher level information
- Context not available at appropriate level



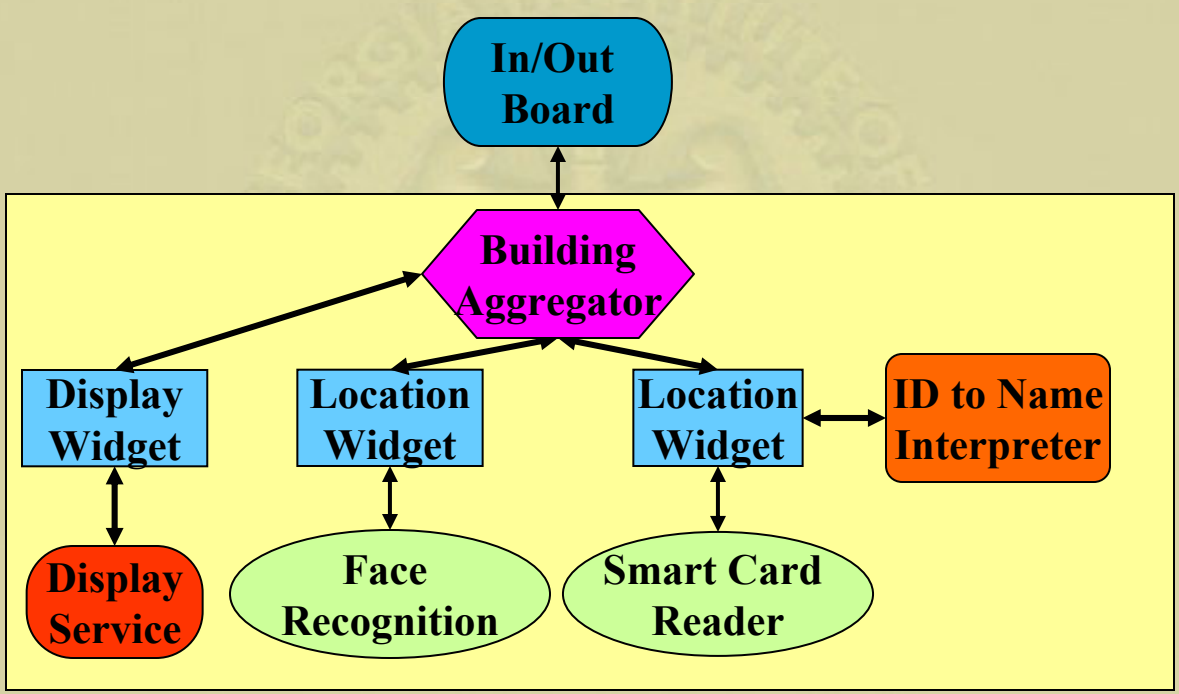
Building Blocks: Aggregators

- Context aggregators – collect context relevant to particular entities
- Further separation, simplifies design



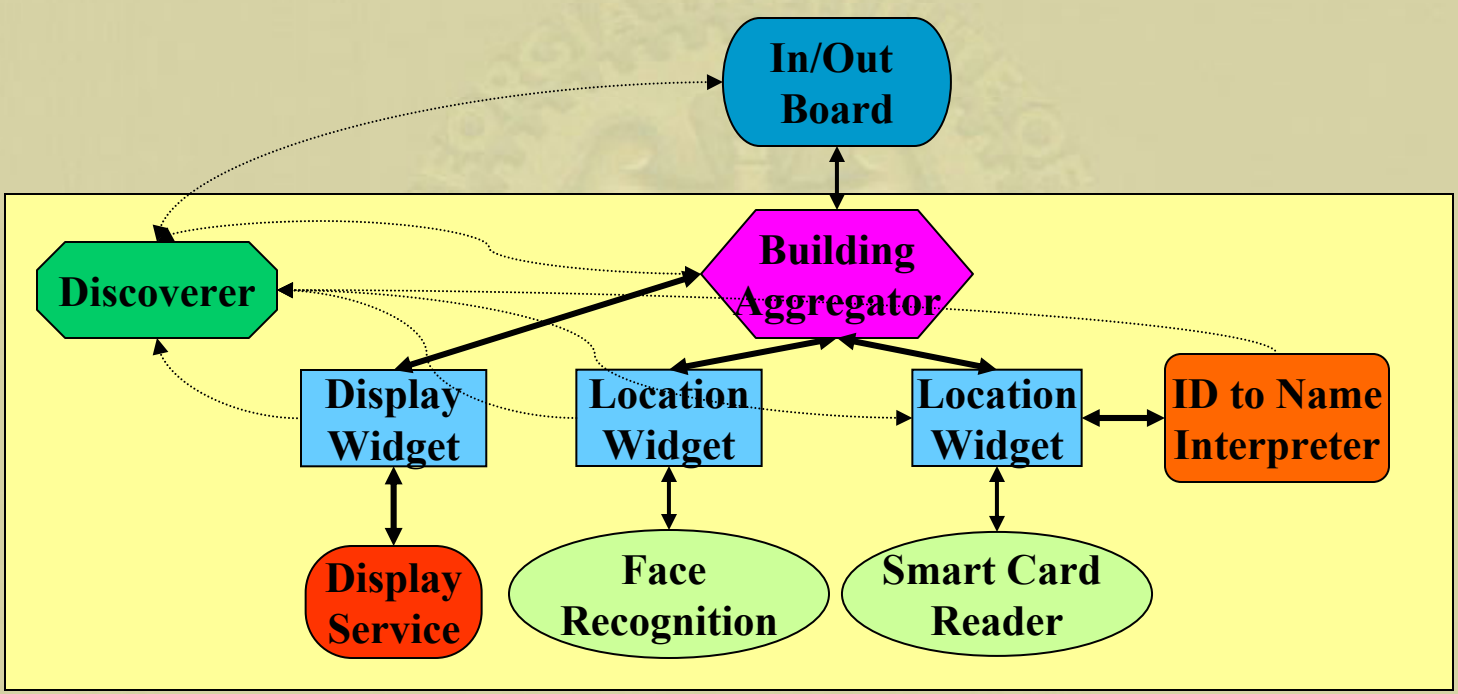
Building Blocks: Services

- Context services – perform behaviors that act on the environment



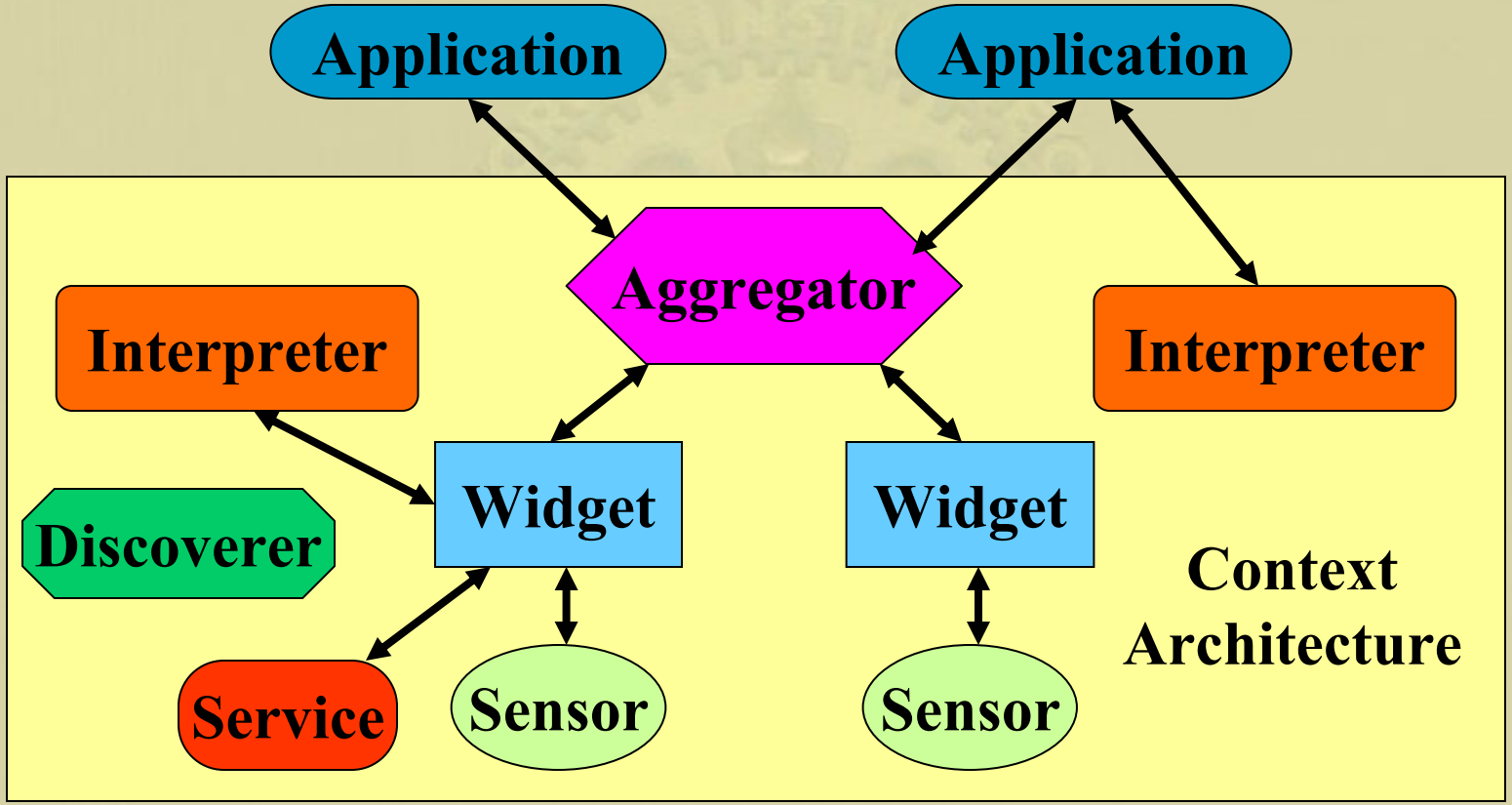
Building Blocks: Discoverer

- Discoverer – registry for context components



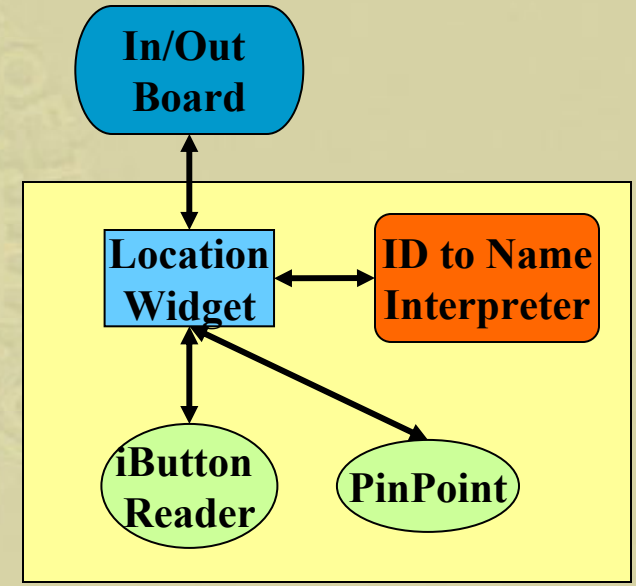
Component-Based Design

- Context component abstraction



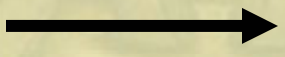
Applications

- In/Out Board and Context-Aware Mailing List (Salber98) – simple, reusable, evolving



Situations: Design Process Revisited

1. Specification
2. Acquisition
3. Delivery
4. Reception
5. Action

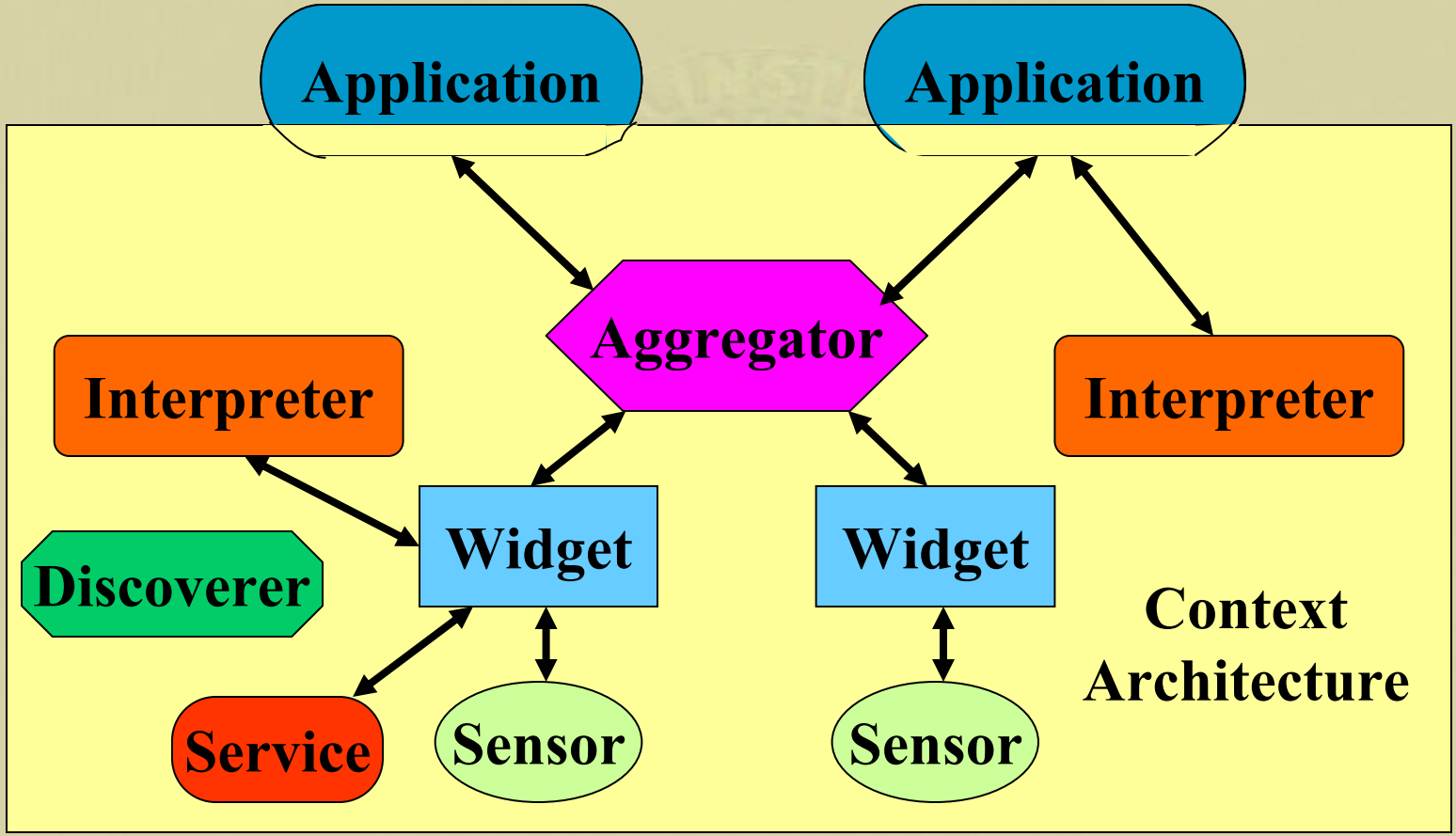


1. Specification
2. *Acquisition*
3. Action

Situation Abstraction

- Revisit context definition
- Allow programmer to define a situation (real-world callbacks)
- Architecture's responsibility to deliver it (if possible)
- Makes *specification* in design process simpler

Situation-based Design



Group Exercise #2

- Re-develop the same application
- This time, make use of the support we've discussed
- Should be much easier

Group Exercise #2 Review

- What was hard?
- What was easy?
- What would you do differently/what additional support is required?

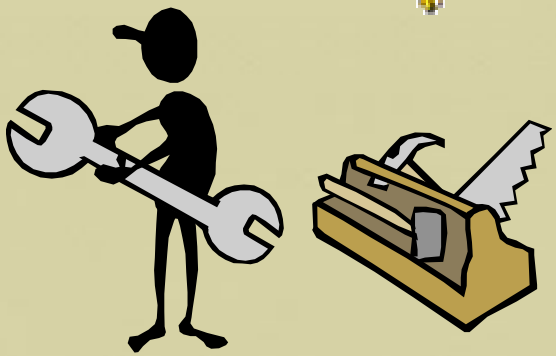
Presentation Overview

- Motivation and What is Context
- Design Process – no support
- Design Process – with support
- Survey of Support
- Research Challenges

Survey of Support

- Bootstrap building of applications – the interesting part!
- Comparison points for those building their own support

Survey of Support – Existing



- Out of date

- Schilit (Columbia, 1995)
- Stick-e notes (Kent, 1996)
- CyberDesk (Georgia Tech, 1997)
- Spirit/Anatomy (ATT UK, 1997-99)
- CALAIS (Cambridge, 1998)
- Context Toolkit (Georgia Tech, 1998)
- TEA (Teco/Nokia, 1999)

Survey of Support - Proposed

- Situated Computing Service (HP, 1997)
- Contextual Information Service (Kent, 1998)
- Ektara (MIT, 2000)

Survey of Support - Other

- OAA (OGI, 1996)
- HIVE (MIT, 1998)
- Sulawesi (Essex, 1998)
- Interactive Workspaces (Stanford, 1998)
- EasyLiving (MSR, 1999)

Presentation Overview

- Motivation and What is Context
- Design Process – no support
- Design Process – with support
- Survey of Support
- Research Challenges

Other Challenges

- Killer applications
- Taxonomy
- "Real"-world Knowledge
- Privacy
- Quality of Service
- Evaluation
- Transparency/Control
- Balance of intelligence
- Dealing with ambiguity
- Environment model (location, people, (devices?))
- Complex Interpretation

Killer Applications?

- Need something to focus and drive the research
- Need something to put in the hands of real people
- Nigel's big button

- Communications, support for everyday activities (finding things)

Taxonomy

- What context is important? Always and in different situations?
- How do you represent context?
- Do we need standards to allow groups to share components?

Real-World Knowledge

- Knowledge about how the world works
- Useful for advanced reasoning
- Multiple efforts:
  - ThoughtTreasure (www.signiform.com)
  - OpenMind (commonsense.media.mit.edu)
  - MindPixel (www.mindpixel.com)

Privacy

- Capturing/collecting lots of information about people, places and devices
- People uncomfortable when don't know what is being collected and how it's used
- Social vs. technological solutions

Quality of Service: Metadata

- Coverage
- Resolution
- Accuracy, confidence
- Reliability
- Frequency
- Timeliness

Intelligence

- Who is smart? User or system or both
- Who makes the decisions on what actions to take?
- Weiser: "If a computer merely knows what room it is in, it can adapt its behavior … without even a hint of AI"
- Tradeoff between user cognitive load and effort to make system "smart"

Ambiguity

- Deborah Estrin et al: partially observable dynamic system; sensors are limited
- Generally and *incorrectly* assume sensed data or inferences is accurate: ignore the confidence value
- Different ways to deal
  - Improve inferencing
  - Bring the user into the loop

Environmental Model

- Relationship between locations: hierarchical, containment, distance
- Relationship between people: friends, family, colleagues, hierarchical
- Relationship between devices?

Complex Interpretation

- Lots of interesting work here
- Most context-awareness deals with simple forms of context
- Fusion
- Sophisticated applications require higher level forms of context
- Numerical machine learning, HMMs, etc.

Summary

- What is context, context-awareness?
- Why is it interesting/valuable?
- Support for building applications: useful abstractions
- What are the interesting challenges?

Annotated Bibliography – Presentation References

- Bill Schilit, Norman Adams, and Roy Want. *Context-aware computing applications*. IEEE Workshop on Mobile Computing, Dec. 1994
  - Taxonomy of context-aware behaviors
  - http://www.fxpal.xerox.com/people/schilit/wmc-94-schilit.pdf
- Jason Pascoe. *Adding generic contextual capabilities to wearable computers*. 2nd IEEE International Symposium on Wearable Computers, October 1998, pp. 92-99.
  - Taxonomy of context-aware behaviors
  - Proposed support: Contextual Information Service
  - http://www.cs.ukc.ac.uk/pubs/1998/676/content.zip

Annotated Bibliography – Presentation References

- Daniel Salber, Anind K. Dey and Gregory D. Abowd. *The Context Toolkit: Aiding the development of context-enabled applications*. 1999 Conference on Human Factors in Computing Systems (CHI'99), Pittsburgh, PA, May 15-20, 1999. pp. 434-441.
  - In/Out Board
  - http://www.cc.gatech.edu/fce/contexttoolkit/pubs/chi99.pdf
- Roy Want *et al. The Active Badge location system*. ACM Transactions on Information Systems, vol. 10, pp. 91-102, Jan. 1992.
  - ftp://ftp.orl.co.uk/pub/docs/ORL/tr.92.1.ps.Z

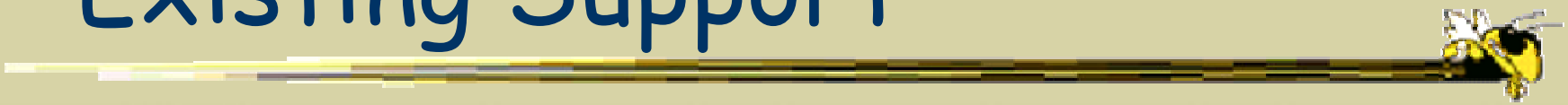
Annotated Bibliography – Presentation References

- M. Lamming and M. Flynn, *"Forget-me-not" Intimate Computing in Support of Human Memory*. Friend21 '94 International Symposium on Next Generation Human Interface.
  - http://www.xrce.xerox.com/publis/cam-trs/pdf/1994/epc-1994-103.pdf
- Jason Pascoe, Nick Ryan and David Morse, *Mobile computing in a fieldwork environment*.
  - MCFE
  - http://www.cs.ukc.ac.uk/research/infosys/mobicomp/Fieldwork/Reports/fw97.html

Annotated Bibliography – Presentation References

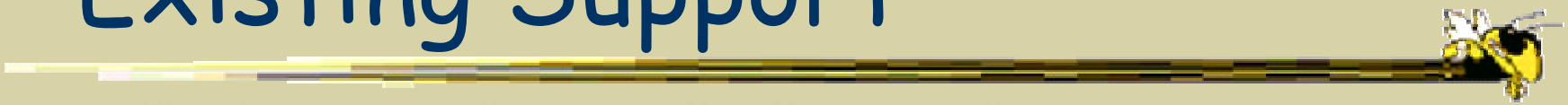
- Anind K. Dey *et al. The Conference Assistant: Combining Context-Awareness with Wearable Computing*, 3rd International Symposium on Wearable Computers (ISWC '99), October 1999. pp. 21-28.
  - Conference Assistant
  - http://www.cc.gatech.edu/fce/ctk/pubs/ISWC99.pdf
- F. P. Brooks, "No Silver Bullet--Essence and Accidents of Software Engineering," IEEE Computer 20 No. 4, pp. 10-19, April 1987.
  - http://www.cis.gsu.edu/~mmoore/CIS3300/handouts/BrooksNoSilverBullet.html

Annotated Bibliography – Existing Support



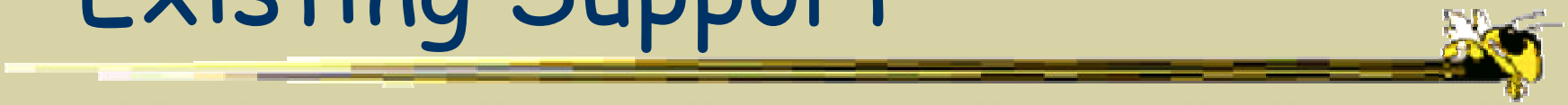
- Bill N. Schilit. *A Context-Aware System Architecture for Mobile Distributed Computing*. PhD Thesis, Columbia University, May 1995.
  - First architecture support for context-awareness
  - http://www.fxpal.xerox.com/people/schilit/schilit-thesis.pdf
- P.J. Brown, J. D. Bovey, and X. Chen, *Context-Aware Applications: From the Laboratory to the Marketplace.* IEEE Personal Communications October, 1997.
  - Architecture support for non-programmers
  - Situation support
  - Stick-e notes
  - http://www.cs.ukc.ac.uk/people/staff/pjb/papers/personal_comms.html
  - Software: http://www.cs.ukc.ac.uk/research/infosys/mobicomp/Fieldwork/Software/index.html

Annotated Bibliography – Existing Support



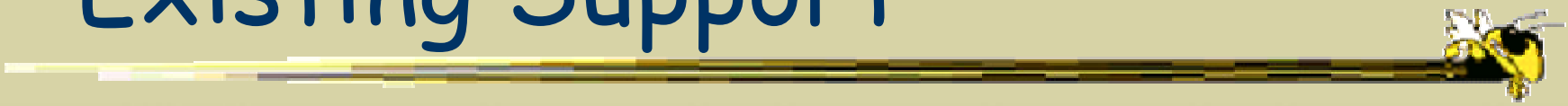
- Anind K. Dey, Gregory D. Abowd and Andy Wood. *CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services*. Knowledge-Based Systems, Vol. 11(1), September 1998, pp. 3-13.
  - Architecture support for desktop-based context
  - http://www.cc.gatech.edu/fce/cyberdesk/pubs/IUI98/IUI98.html
  - http://www.cc.gatech.edu/fce/cyberdesk/
- Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, Paul Webster. *The Anatomy of a Context-Aware Application*. Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom '99), August 1999.
  - SPIRIT
  - ftp://ftp.uk.research.att.com/pub/docs/att/tr.1999.7.ps.gz

Annotated Bibliography – Existing Support



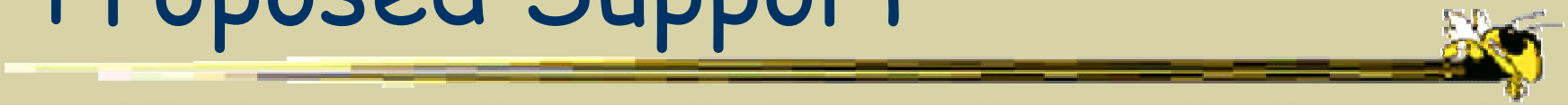
- G.J. Nelson. *Context-Aware and Location Systems*. PhD Thesis, University of Cambridge, January 1998.
  - CALAIS
  - http://www.acm.org/sigmobile/MC2R/theses/nelson.ps.gz
- Anind K. Dey, Daniel Salber and Gregory D. Abowd. *A Context-based Infrastructure for Smart Environments*. 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99), December, 1999. pp. 114-128.
  - Context Toolkit
  - Component and situation-based support
  - http://www.cc.gatech.edu/fce/contexttoolkit/pubs/MANSE99.pdf
  - Software: http://www.cc.gatech.edu/fce/contexttoolkit

Annotated Bibliography – Existing Support



- Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, Walter Van de Velde. Advanced Interaction in Context. 1st International Symposium on Handheld and Ubiquitous Computing (HUC99), 1999.
  - Situation support for non-programmers
  - TEA
  - http://www.teco.uni-karlsruhe.de/~albrecht/publication/huc99/advanced_interaction_context.pdf

Annotated Bibliography – Proposed Support



- R. Hull, P. Neaves and J. Bedrod-Roberts. *Towards Situated Computing*. 1st IEEE International Symposium on Wearable Computers, October 1997.
  - Proposed architecture
  - Situated Computing Service
  - http://www.hpl.hp.com/techreports/97/HPL-97-66.pdf
- Richard W. DeVaul, Alex Pentland. *The Ektara Architecture: The Right Framework for Context-Aware Wearable and Ubiquitous Computing Applications*. MIT Media Lab Tech Report, 2000.
  - Proposed architecture for wearable computing
  - http://www.media.mit.edu/~rich/DPiswc00.pdf

Annotated Bibliography – Other Support

- P. R. Cohen, A. J. Cheyer, M. Wang, and S. C. Baeg. *An open agent architecture*. AAAI Spring Symposium, pp. 1-8, March 1994.
  - Blackboard-based agent architecture
  - http://www.ai.sri.com/pubs/papers/Cohe9403-1:Open/document.ps.gz
  - Software: http://www.ai.sri.com/~oaa/

- Nelson Minar, Matthew Gray, Oliver Roup, Raffi Krikorian, and Pattie Maes. Hive: Distributed agents for networking things. 1st International Symposium on Agent Systems and Applications & 3rd International Symposium on Mobile Agents, 1999.
  - Mobile agent-based architecture
  - http://hive.sourceforge.net/hive-asama.html
  - Software: http://hive.sourceforge.net/

Annotated Bibliography – Other Support

- N.J. Newman and A.F. Clark. Sulawesi: A Wearable Application Integration Framework. 3rd IEEE International Symposium on Wearable Computers, 1999.
  - Architecture for wearable computing
  - http://wearables.essex.ac.uk/reports/iswc99.ps
  - Software: http://wearables.essex.ac.uk/sulawesi/

- B.L. Brumitt, S. Shafer, J. Krumm, B. Meyers. *EasyLiving and the role of Geometry in Ubiquitous Computing*. DARPA/NIST/NSF Workshop on Research Issues in Smart Computing Environments, July 1999.
  - Architecture for smart environments
  - http://www.research.microsoft.com/barry/research/ELandGeometry2.pdf