

The Context Toolkit: Supporting Context- Aware Computing

Anind K. Dey
Future Computing Environments
College of Computing & GVU Center
Georgia Tech

Contrast to Traditional GUI

- Not only direct interaction for input and output
- Need to use explicit AND implicit input
- Different computing paradigm

2/1/01

3

Ubiquitous Computing

- Off the desktop, invisible, pervasive



2/1/01

2

Context and Context-Awareness

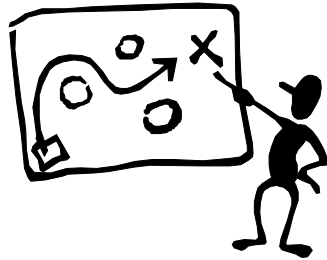
- Context: *any information that can be used to characterize the situation of an entity*
- Context-awareness
 - General model of interactive computing
 - Addresses subset of ubicomp problems

2/1/01

4

Talk Overview

- Difficult in using context and building applications
- Context Toolkit solution
- Applications
- Research platform



2/1/01

5

Building Applications

- M. Weiser: The whole point of ubiquitous computing, of course, is the applications.
- But ... what if the applications are hard to build?

2/1/01

7

Context-Aware Applications

- Tour guides (museums)
- Travel assistants (Hertz NeverLost)
- Personalization systems (myYahoo)
- And even ...

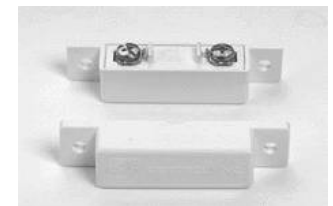


2/1/01

6

Why Context is Hard to Use

- Acquired from sensors
 - Not just keyboards and mice – lots of heterogeneous devices
- Need to abstract data
- Distributed
- Dynamic



2/1/01

8

Results of Difficulties

- *Ad hoc* application building
 - Difficult to build, reuse and evolve
- Small variety of sensors
- Small variety of context
- Few applications, mostly simple
- Practical: difficult to prototype, test and evaluate

2/1/01

9

Context Toolkit Solution

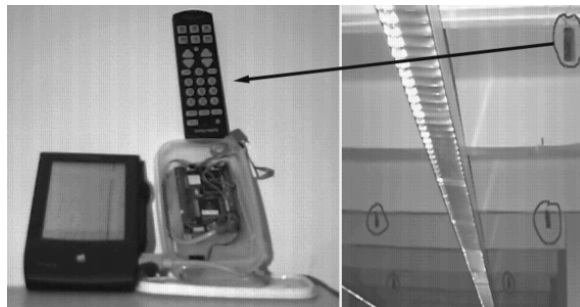
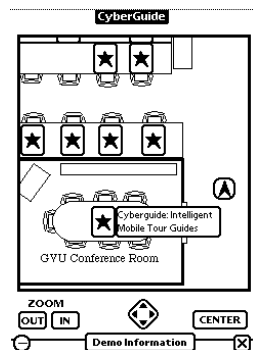
- Need to address problems of dealing with context
- Toolkit provides:
 - Methodology for dealing with context
 - Framework for acquiring and handling context
 - Library of customizable components

2/1/01

11

Why Applications are Hard to Build: A Case Study

- Cyberguide case study: no separation of concerns

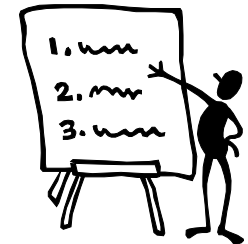


2/1/01

10

Toolkit Requirements

- Context specification

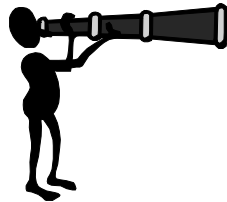


2/1/01

12

Toolkit Requirements

- Context specification
- Discovery

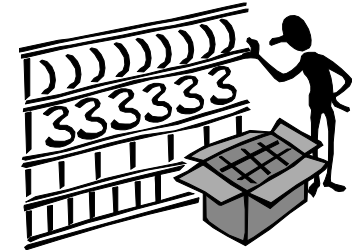


2/1/01

13

Toolkit Requirements

- Context specification
- Discovery
- Separation of concerns
- Storage

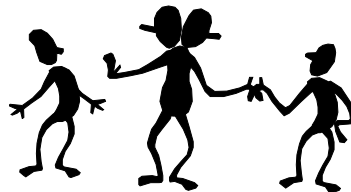


2/1/01

15

Toolkit Requirements

- Context specification
- Discovery
- Separation of concerns



2/1/01

14

Toolkit Requirements

- Context specification
- Discovery
- Separation of concerns
- Storage
- Interpretation

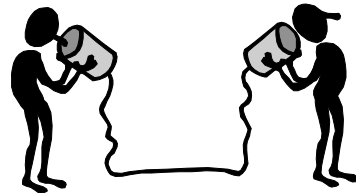


2/1/01

16

Toolkit Requirements

- Context specification
- Discovery
- Separation of concerns
- Storage
- Interpretation
- Transparent communications



2/1/01

17

Learning from GUIs

- Concept of a widget
 - Separation, callbacks, attributes, encapsulation, abstraction
- *e.g.* button



2/1/01

19

Toolkit Requirements

- Context specification
- Discovery
- Separation of concerns
- Storage
- Interpretation
- Transparent communications
- Constant availability

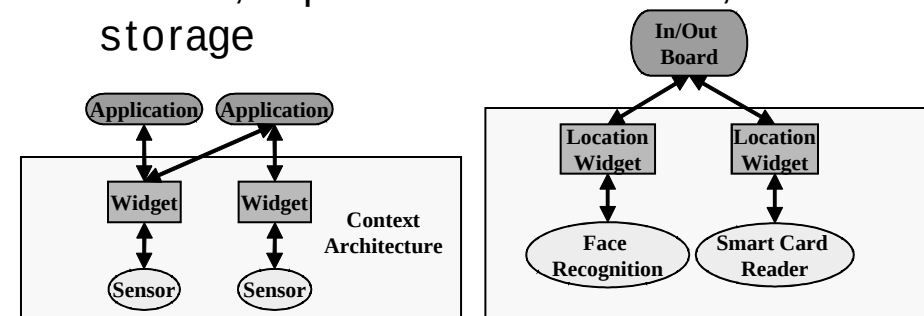


2/1/01

18

Context Widgets

- Responsible for acquiring and abstracting data from particular sensor, separation of concerns, storage

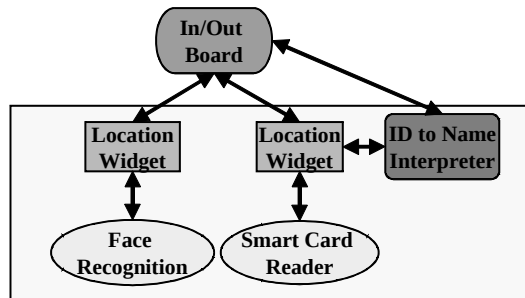


2/1/01

20

Context Interpreters

- Convert or interpret context to higher level information
- Context not available at appropriate level

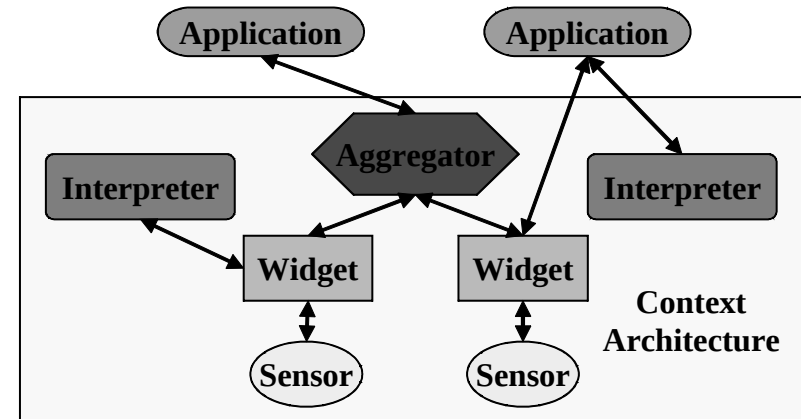


2/1/01

21

Context Toolkit Framework

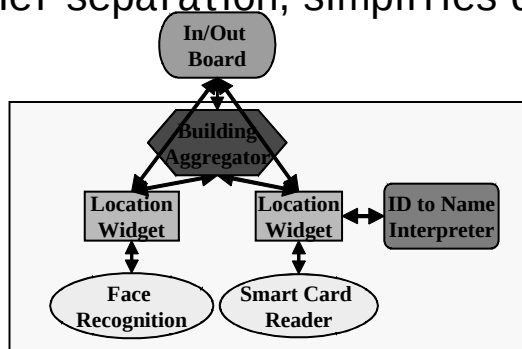
- Supports real-world model/methodology and provides library



23

Context Aggregators

- Collect context relevant to particular entities (recall definition)
- Further separation, simplifies design



22

Applications

- In/Out Board
- Serendipitous Meetings
- Aware Home
- Conference Assistant

2/1/01

24

In/Out Board and Context-Aware Mailing List (CHI '99)

- Simple app demonstrates support for reusability and evolving applications

Name	Status	Time
Gregory Abowd	Out	10:50am
Jason Brotherton	In	9:28am
Anind Dey	In	12:08pm
M. Futakawa	In	12:00pm
Y. Ishiguro	Out	10:52am
Rob Kooper	Out	5:26pm
Kent Lyons	Out	12:27pm
Jen Mankoff	In	12:08pm
David Nguyen	In	11:19am
Rob Orr	Out	1:25pm
Maria Pimentel	Out	5:54pm
Daniel Salber	In	10:14am
Brad Singletary	Out	2:50pm
Khai Truong	Out	1:25pm

2/1/01

Name	Status
Gregory Abowd	in
Jason Brotherton	out
Anind Dey	in
Tanisha Hall	out
Cory Kidd	out
Kent Lyons	in
Jen Mankoff	in
Todd Miller	out
Kris Nagel	in
David Nguyen	out
Rob Orr	in
Daniel Salber	out
Chris Shaw	out
Brad Singletary	in
Khai Truong	out

25

Serendipitous Meetings

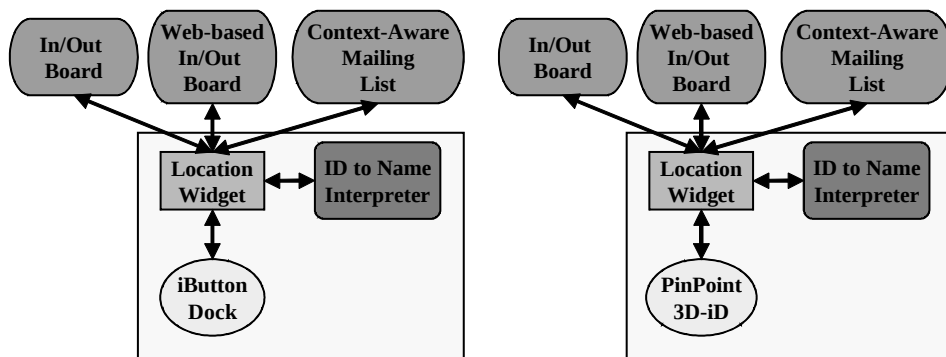
Annotations for the Serendipitous Meetings application:

- Ink written before current time is in original color**: Points to handwritten notes on the whiteboard.
- Ink written after current time is in original color**: Points to handwritten notes on the whiteboard.
- Current time within session**: Points to the timeline.
- Selected session**: Points to the timeline.
- Selected day**: Points to the timeline.
- Day containing whiteboard activity**: Points to the timeline.
- Filters**: Points to the timeline.
- Playback controls**: Points to the timeline.

2/1/01

27

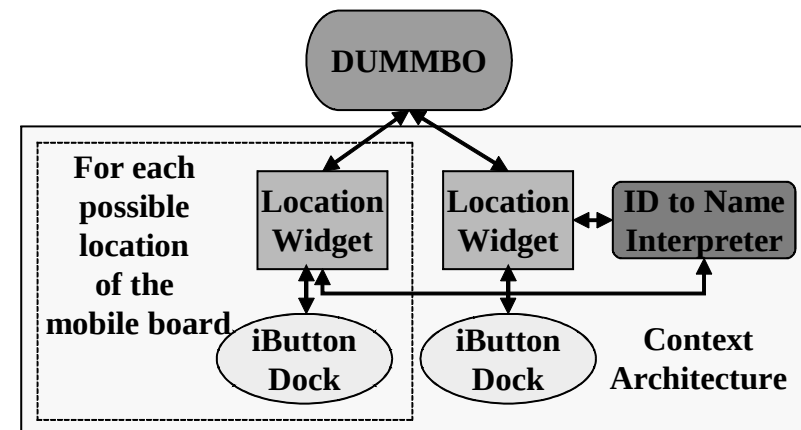
In/Out Board Architecture



2/1/01

26

Meeting Architecture



2/1/01

28

Aware Home (MANSE '99)

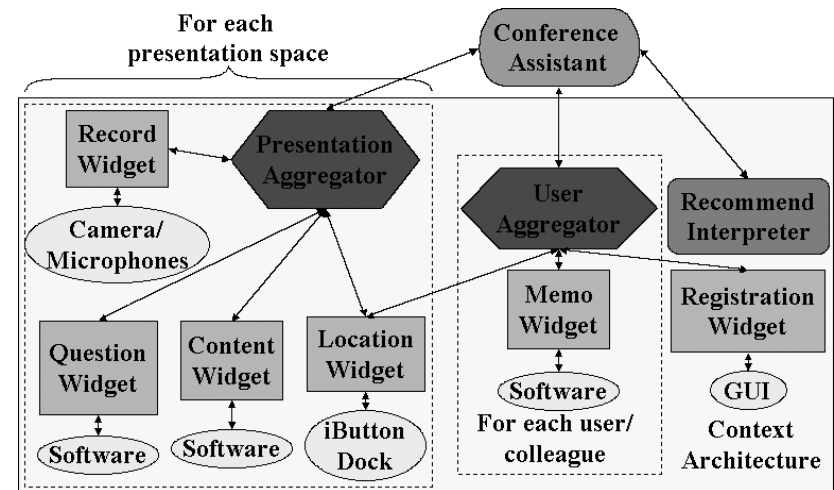
- Gre
- 3 g
- VI



2/1/01

29

Conference Assistant Arch.



2/1/01

31

2/1/01

30

Additional Apps (if time)

- Privacy: Dynamic Door Displays
- End-user programming: CybreMinder

2/1/01

32

Access Control

- Dyn



2/1/01

33

What Have We Provided?

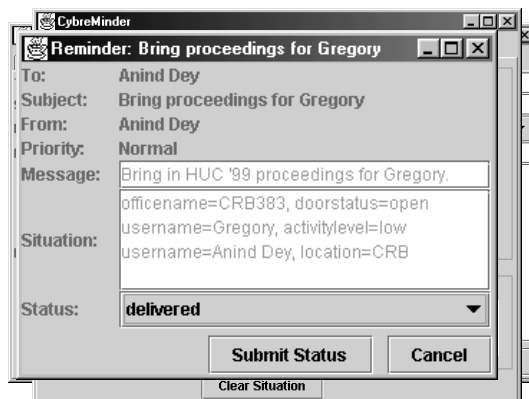
- Conceptual framework requirements
 - Lower threshold, raise ceiling
- Context Toolkit
 - Implementation and exploration
 - Easy to create and evolve applications
- Programming abstractions
 - Reusable widgets, aggregators, interpreters
 - Support simplified design process for building apps



2/1/01

35

CybreMinder (HUC '00)



2/1/01

34

Research Agenda



- What to do, and when and why
- Broaden scope of framework to be a general model of interactive and ubiquitous computing
- Complex interpretation

2/1/01

36

Conclusions

- Context-aware applications are hard to build
- Provide design process and support for building
- Implementation allows investigation of other problems in context-aware computing

2/1/01

37

Typical Design Process

- **Specification** – context and behaviors
- **Acquisition** – install, API, query/notify, store, interpret
- **Delivery** – deliver context to multiple, remote applications
- **Reception** – locate relevant sensors, request context, interpret
- **Action** – analysis and action

2/1/01

39

Acknowledgements

- Motorola & NSF & FCE
- Contact info:
 - anind@cc.gatech.edu
 - <http://www.cc.gatech.edu/fce/ctk>
 - Available for public download
- Questions?



2/1/01

38

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

2/1/01

40

Design Process Revisited

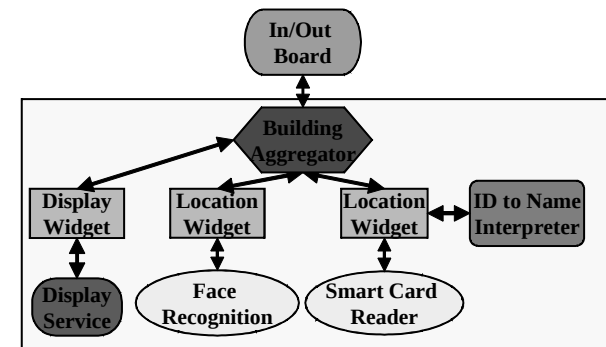
- | | |
|-----------------|----------------------|
| ■ Specification | ■ Specification |
| ■ Acquisition | ■ <i>Acquisition</i> |
| ■ Delivery | → |
| ■ Reception | |
| ■ Action | ■ Action |

2/1/01

41

Context Services

- Extend concept of widget to interactor
- Perform behaviors that act on the environment

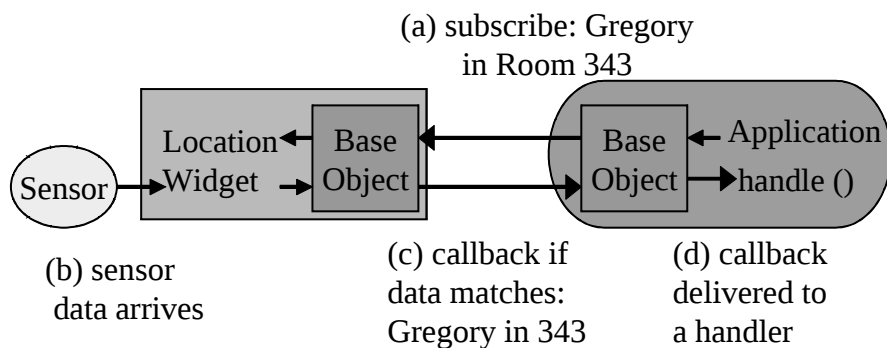


2/1/01

43

Callback Model

- Transparent communications, always available

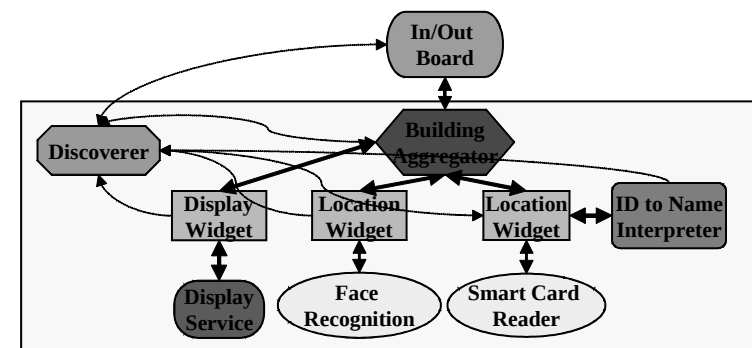


2/1/01

42

Discoverers

- Registry for context components

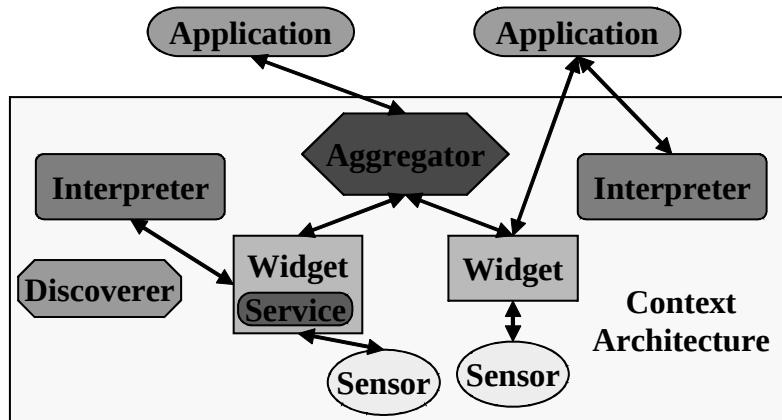


2/1/01

44

Context Toolkit Framework

- Supports real-world model/methodology and provides library



45

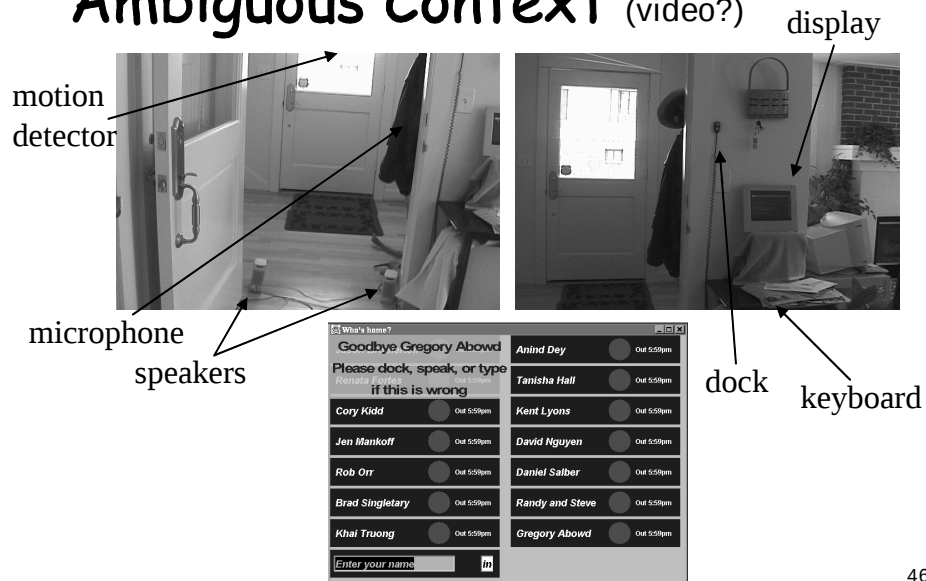
Situations: Declarative Style

- Say what you want, not how you want it done – framework figures it out
- Allow programmer to define a situation (complex real-world callbacks)
- *Specification* in design process simpler
- More robust w.r.t. component failures and easier to evolve

2/1/01

47

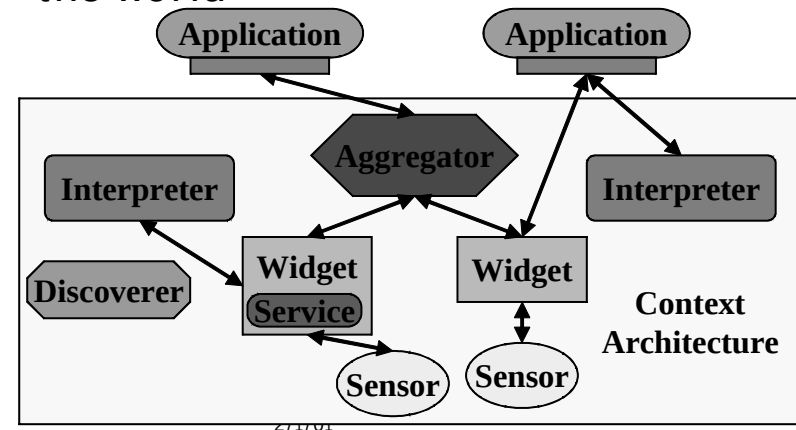
Ambiguous Context (video?)



46

Revised Framework

- Supports blackboard/box model of the world



48