Ubicomp and Physical Interaction

Ubicomp?

- Computation embedded in the physical spaces around us
- "Ambient intelligence"
- Take advantage of naturally-occurring actions and activities to support people
  - Input in the real world
  - Output in the real world also
- Culmination of our discussion of natural data types
- "Context-aware computing" -- making computers more aware of the context of the people who are using them

What is Context?

- Any information that can be used to characterize the situation of an entity
    - Who, what, where, when
- Why is it important?
    - information, usually implicit, that applications do not have access to
    - It's input that you don't get in a GUI

How to Use Context

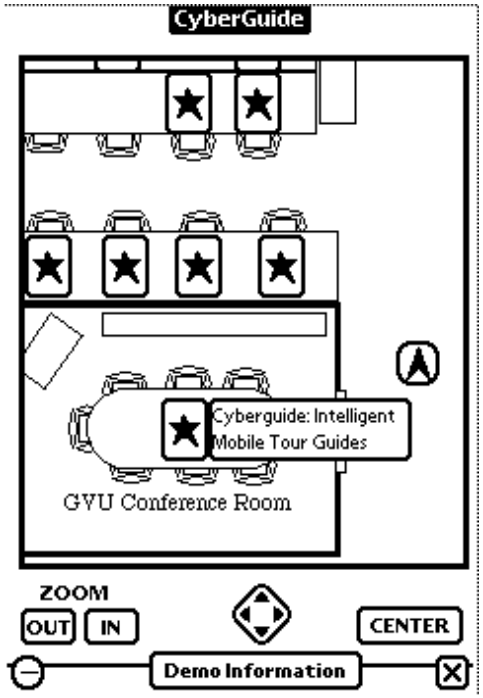
- To present relevant information to someone
  - Mobile tour guide
- To perform an action automatically
  - Print to nearest printer
- To show an action that use can choose
  - Want to phone the number in this email?

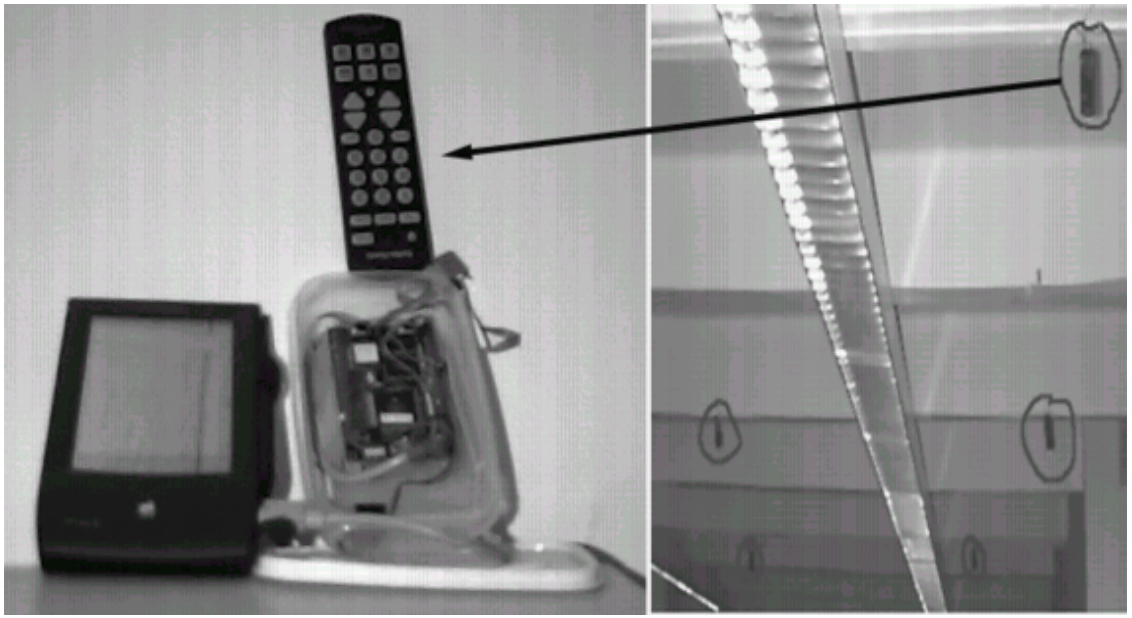
Case Study: tour guides

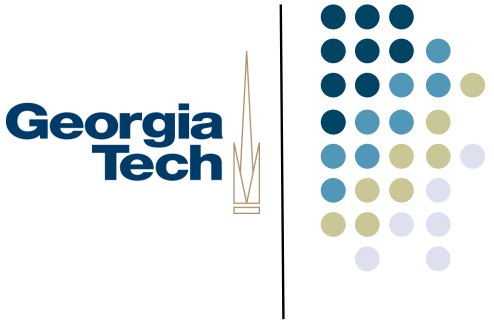
- Very popular theme
  - Location is an easy piece of context

- G. Abowd *et al.* Cyberguide: A mobile context-aware tour guide. ACM Wireless Networks, 3:5, 1997.

How Cyberguide worked







Why is this hard?

- Steps
  - Acquisition
  - Representation
  - Interpretation
  - Storage
  - Delivery
  - Reaction
- Most of these steps repeated in all development.

Early Work on Context Support

- Bill Schilit, Xerox PARC
  - Main software architect of PARCTab
  - Location-aware rules for app behavior

The Context Toolkit

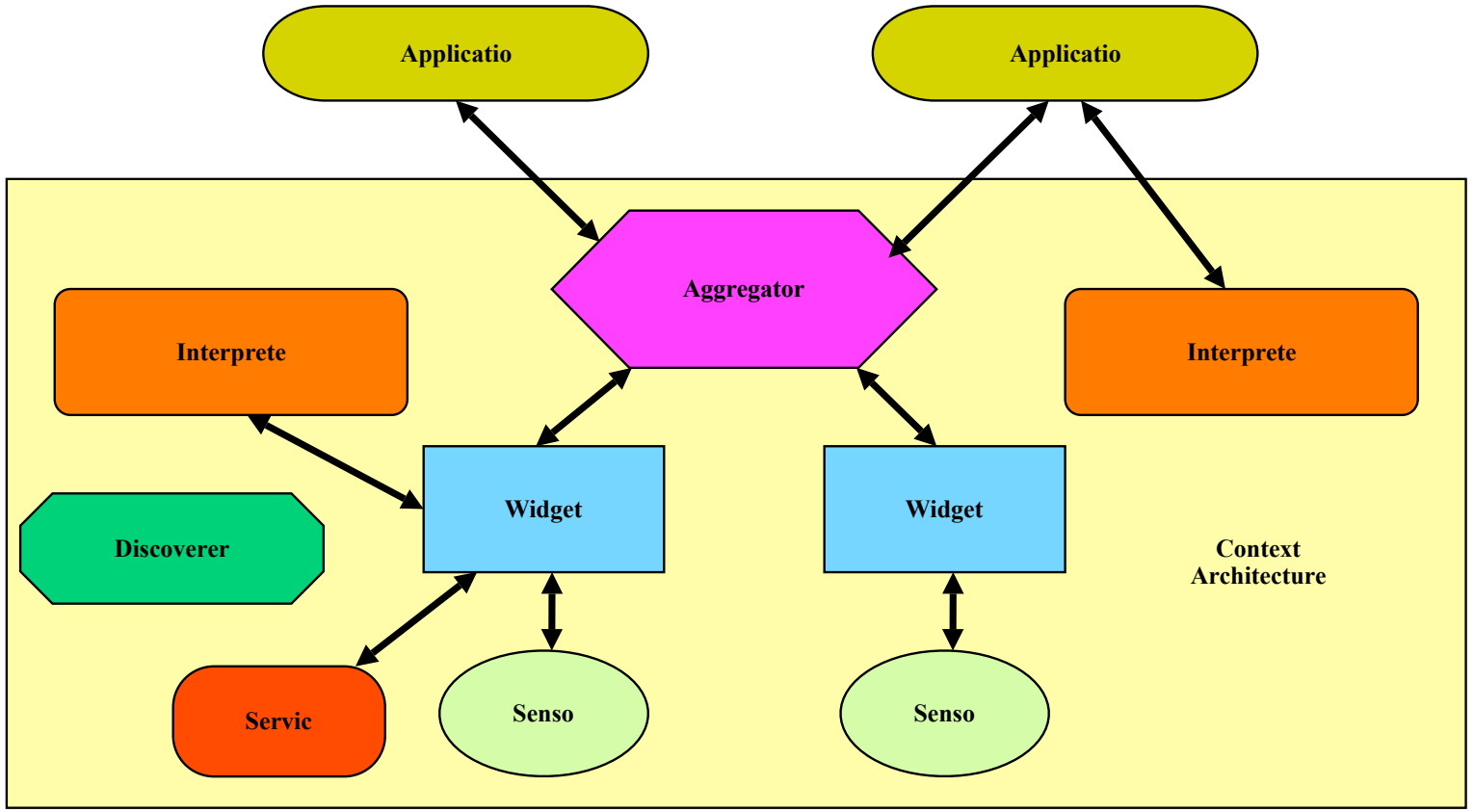
D. Salber, A. Dey & G. Abowd. The Context Toolkit: Aiding the development of context-enabled applications. *CHI '99*, pp. 434-441.

Toolkit available at: http://www.cc.gatech.edu/fce/ctk

- Three main abstractions:
  - Context widget
  - Interpreter
  - Aggregator

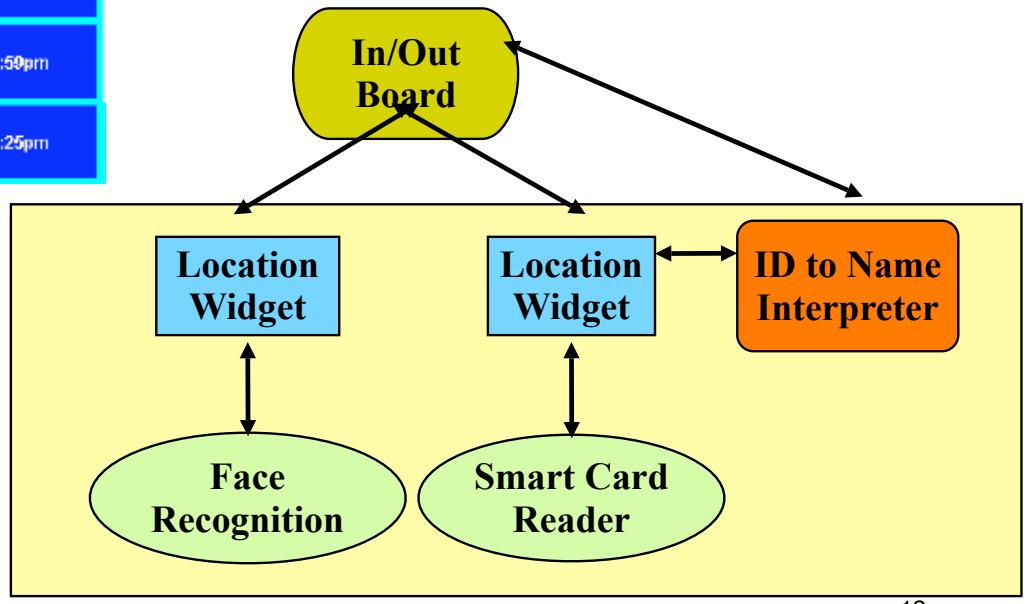
The Context Toolkit

- Context component abstraction



Simple Example: In/Out Board

Simple Example: In/Out Board



What remains hard?

- Sensing…
- Actuation…
- We'll get back to how to address these (Phidgets)

Example: Intelligent Spaces

- Stanford Interactive Workspaces Project: iRoom
- Since 1999
- http://iwork.stanford.edu

- Focus:
  - Single room
  - Collection of large/small displays
  - Synchronous, collocated, small workgroups

Guiding Principles

- Rely on social conventions
  - User control vs. automatic "smart" behavior
  - The Semantic Rubicon
- Wide applicability
  - Think about variety of interactive spaces
- Simplicity
  - From user and developer perspective

Displays

- Tiled SmartBoards
- Interactive Mural
- Table top
- Laptops

Interaction Techniques

- Point Right
  - Brad Johanson, M. Stone and T. Winograd, PointRight: Experience with Flexible Input Redirection in Interactive Workspaces, UIST 2002.

- Simplified control of mouse/keyboard input focus across multiple displays

Interaction Techniques

- Flow Menu
  - F. Guimbrètiere and T. Winograd. Flow Menu: Combining Command, Text and Data Entry. UIST 2000.

- Smooth integration of command selection and parameter input for pen-based interaction.

Interaction Techniques

- Multibrowsing
  - B. Johanson, S. Ponnekanti, C. Sengupta, A. Fox. Multibrowsing: Moving web content across multiple displays. Ubicomp 2001.

- Technique for integrating Web content with multiple displays.

Interaction Techniques

- Scaling behavior in interactive mural
  - F. Guimbrètiere, M. Stone and T. Winograd, Fluid Interaction with High-resolution wall-size displays. UIST 2001.

Infrastructure

- Services for
  - Data
  - Control
  - Coordination
- iROS
  - Interactive Room Operating System

Infrastructure

- Event Heap
  - B. Johanson and A. Fox. The Event Heap: A Coordination Infrastructure for Interactive Workspaces
  - Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002), June 2002.

- Tuple space implementation
  - Minimize application coordination dependency

Infrastructure

- iCrafter
  - S. Ponnekanti, B. Lee, Armando Fox, Pat Hanrahan, and T. Winograd. ICrafter: A Service Framework for Ubiquitous Computing Environments, Ubicomp 200.
- Flexible I/O interaction with services in an interactive workspace

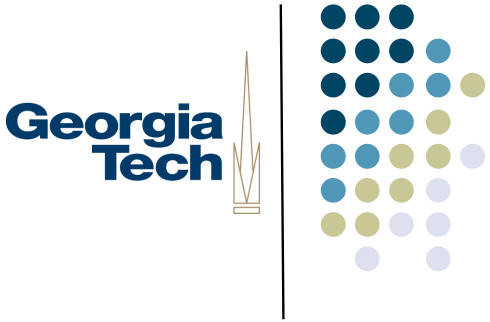
Infrastructure

- iStuff
  - http://www.stanford.edu/~borchers/istuff/
- Simplifying use of physical I/O devices
  - Similar in spirit to phidgets

Related Work

- Spaces
  - CoolTown (HP Labs)
  - eClass, Aware Home (GT)
  - Intelligent Room (MIT)
    - http://www.ai.mit.edu/projects/iroom/
  - Easy Living (Microsoft Research)
  - Ambient Workpaces (Fraunhofer/IPSI, Germany)
    - http://www.ipsi.fhg.de/ambiente/english/index.html
  - House_n (MIT)
    - http://architecture.mit.edu/house_n/

What about sensing and actuation?



- Would like to be able to sense activities in the physical world *and then present feedback/actions in the physical world also*

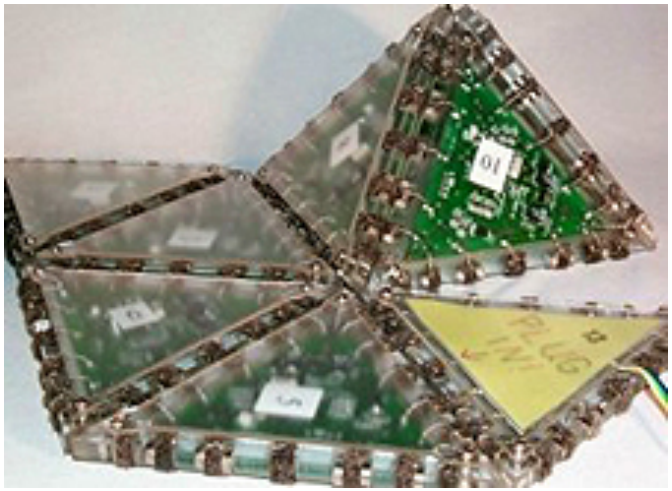
- **Tangible User Interfaces**

Tangible User Interfaces

- Hiroshi Ishii (MIT)
  - **Tangible Bits**
    - physical form to digital information
  - **Tangible User Interfaces**
    - physical objects, surfaces, and spaces that act as tangible embodiments of digital information

Triangles

- Pieces are connected together to trigger digital events
  - influence the progress of a non-linear story
  - organize media elements in order to create their own story space



LumiTouch

- Two interactive picture frames
  - User's touching of a local frame translates to a glow on remote frame
  - She's thinking of him
  - He's thinking of her

Tangible Video Browser

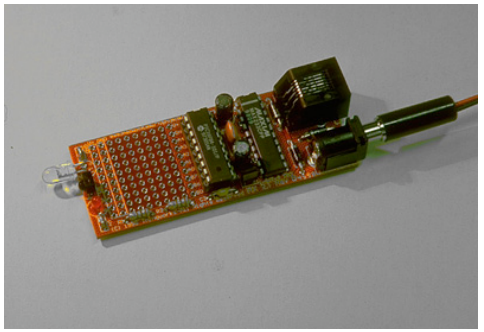
- Tokens are used to:
  - Act as container for videos
  - Select a video
  - Navigate within the video

What remains hard?

- Well…*everything* according to the paper
  - While an exciting new area, everyday programmers still face considerable hurdles if they wish to create even simple physical user interfaces. Perhaps the biggest--but we believe easily solved---obstacle is the sheer difficulty of developing and combining physical devices and interfacing them to conventional programming languages.

Related Work

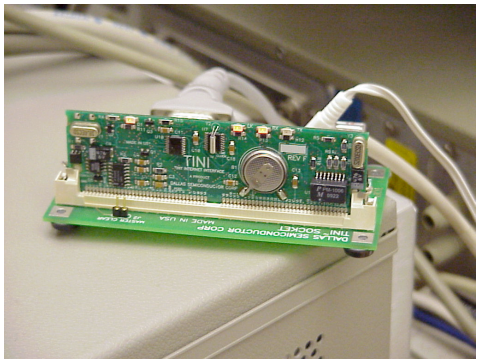
Tools for working with physical input/output devices

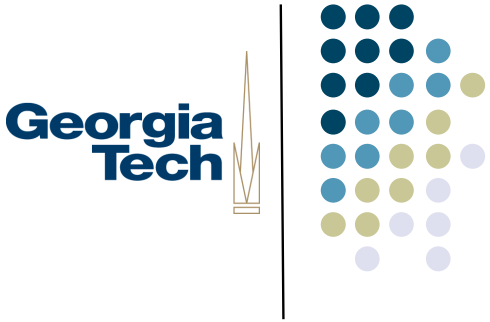


iRX Board

Digital I/O boards

Tini boards





Problems

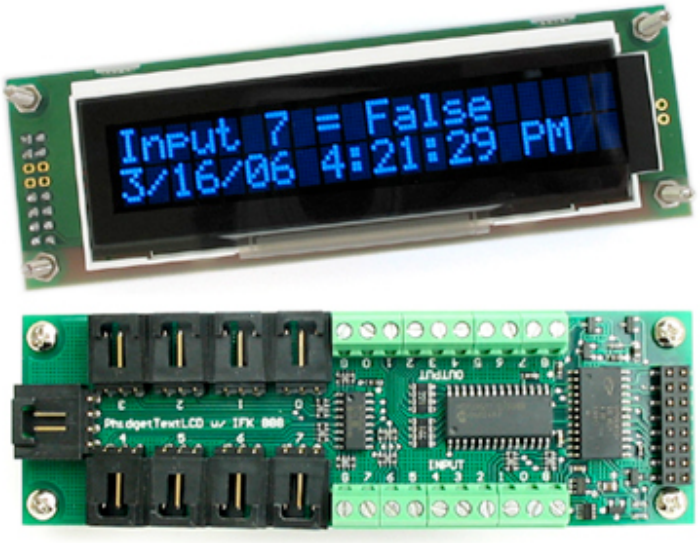
- Hard to build
- No API
- API at wrong abstraction level
- Oriented to different markets
- Difficult to write/debug w/o actual devices

- We'd like to have something that is
  - Simple so developers concentrate on overall use, modification, and recombination
  - Easy for average programmer

Phidgets!

- "Physical widgets"
  - Easily composable hardware devices
  - Provide sensing and actuation
- http://grouplab.cpsc.ucalgary.ca/phidgets/ -- research project page
- http://www.phidgets.com/ -- online store

- Basis concepts:
  - Connection manager
  - ID
  - Simulation mode



Phidget Manager

onAttach()

onDetach()

Count

Item

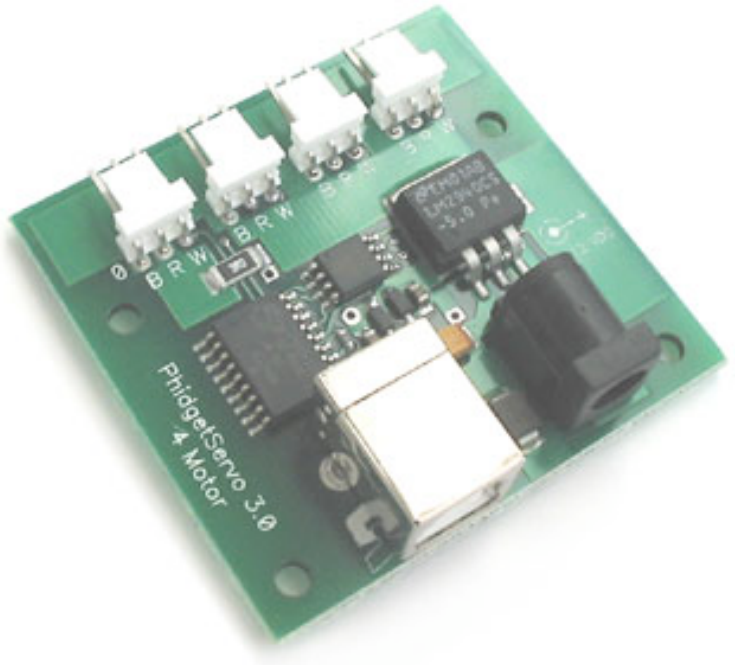
DeviceType

isAttached()

SerialNumber

Example: Phidget Servo

- MotorPosition
- NumMotors
- onPositionChanged()



Drawbacks

- Need PC
- Not mobile
- Not easy to deploy