Einführung in die Erweiterte Realität

11. Context Toolkit

Prof. Gudrun Klinker, Ph.D.
Institut für Informatik,Technische Universität München
klinker@in.tum.de

Jan 25, 2005

Literature

- Anind K. Dey, Gregory D. Abowd, and Danieal Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", *Human-Computer Interaction Journal (HCI)* special issue on context-aware computing, Vol. 16 (2-4), 2001, pp. 97-166, http://www.cc.gatech.edu/fce/contexttoolkit
- Anind K. Dey, "Providing Architectural Support for Building Context-Aware Applications", Ph.D. Thesis, Georgia Institute of Technology, Nov. 2000.

Introduction

Introduction

- User interfaces aside from the traditional desktop environment -> *Ubiquitous Computing*
- *Context* for Human-Computer-Interaction provided by physical and electronic environment -> *Context-Aware Computing*
- Context automatically sensed in a physical environment
- Implicit input to positively affect an application

-> *Location-based services*

Introduction

- Needed: Conceptual models and tools to support the rapid development of rich context-aware applications.
- General understanding:
  - What is a context
  - What can it be used for

Introduction - Difficulties of Handling Context -

- For designers:
  - No conceptual tools and methods to account for context-awarenes.
  - Choice of context often defined by availability of hardware and software sensors (sensor-driven approach)
- For developers:
  - Distribution of context across several places
    -> Distributed networks of sensors
  - Mobile users in a highly dynamic environment
    -> Diverse platforms
- Needed: System for iterative development, easy to modify, reusable

Definition: Context

- Any information that can be used to characterize the *situation of entities* that are considered relevant to the *interaction* between a user and an application.
- Typically: Location, identity and state of people, groups and computational and physical objects.
- Both implicit and explicit user input!
- Here: Context doesn’t have to be acquired automatically

Categories of Context

- Entities
  - Places
  - People / Groups
  - Things
- Categories
  - Identity
  - Location (6 DOF)
  - Status (Activity) (Intrinsic characteristics of an entity)
  - Time
- Inference (derivation) of higher context attributes

Context-Aware Functions

- Presentation of information and services
- Automatic execution of a service
- Attaching context information for later retrieval

Requirements for Dealing with Context

Requirements for Dealing with Context

1. Separation of concerns
2. Context interpretation
3. Transparent, distributed communications
4. Constant availability of context acquisition
5. Context storage
6. Resource discovery

Req1: Separation of Concerns

- Separation between
  - Acquisition of context (low-level details of individual sensors)
  - Handling of context (application semantics)
- GUIs: user input is handled by *Widgets*.
  - Querying mechanism
  - Notifying (callback) mechanism
  - Common external interface for all widgets

Req2: Context Interpretation

- Multiple layers of context data
- Transparent recursive interpretation

Req3: Transparent, Distributed Communications

- Distributed sensors
- Distributed services
- Multiple applications

Fact that communication is distributed should be transparent to both sensors and applications.

Relieve the designer of having to build a communications framework!

Needed: Global timeclock mechanism

Req4: Constant Availability of Context Acquisition

- Context-aware applications should not instantiate individual sensors to acquire context data.
- Context acquisition modules must operate independently of specific applications.
- Available all the time!

Req5: Context Storage

- Use of historical context information to
  - establish trends
  - predict future context values
- Architecture must support storage of context.

Req6: Resource Discovery

- Information prior to establishing a communication: Application -> Sensor:
  - What kind of information does a sensor provide?
  - Where is it located? (Hostname, port)
  - How can an application communicate with it?

-> Central resource discovery mechanism

Context Abstractions

Context Abstractions

1. Context widgets
2. Interpreters
3. Aggregators
4. Services
5. Discoverers

Context Abstractions - 1. Context Widgets -

- Analog to GUI Widgets
- Benefits:
  - Hide the complexity of the acutal sensor
  - Abstract context information to suit the expected needs of an application
  - Provide reusable and customizable building blocks
- Context widgets encapsulate context information and provide methods to access it like a GUI widget (query and notify mechanisms)
- Context widgets provide abstractions that encapsulate acquisition and handling of a piece of context information
- Every attribute is represented by a context widget.

Context Abstractions - 2. Interpreters -

- Raise the level of abstraction of a piece of context!
- Transform context by taking information from one or more context sources and producing a new piece of information

- Traditionally done inside an application.
- Now: separated from applications into reusable interpreter modules -> reusable
- Common interface to all interpreters.

- Collecting pieces of context information that are logically related into a common repository
- E.g: gather information from distributed sensors
- “One-stop-shop”
- Every entity is represented by an aggregator

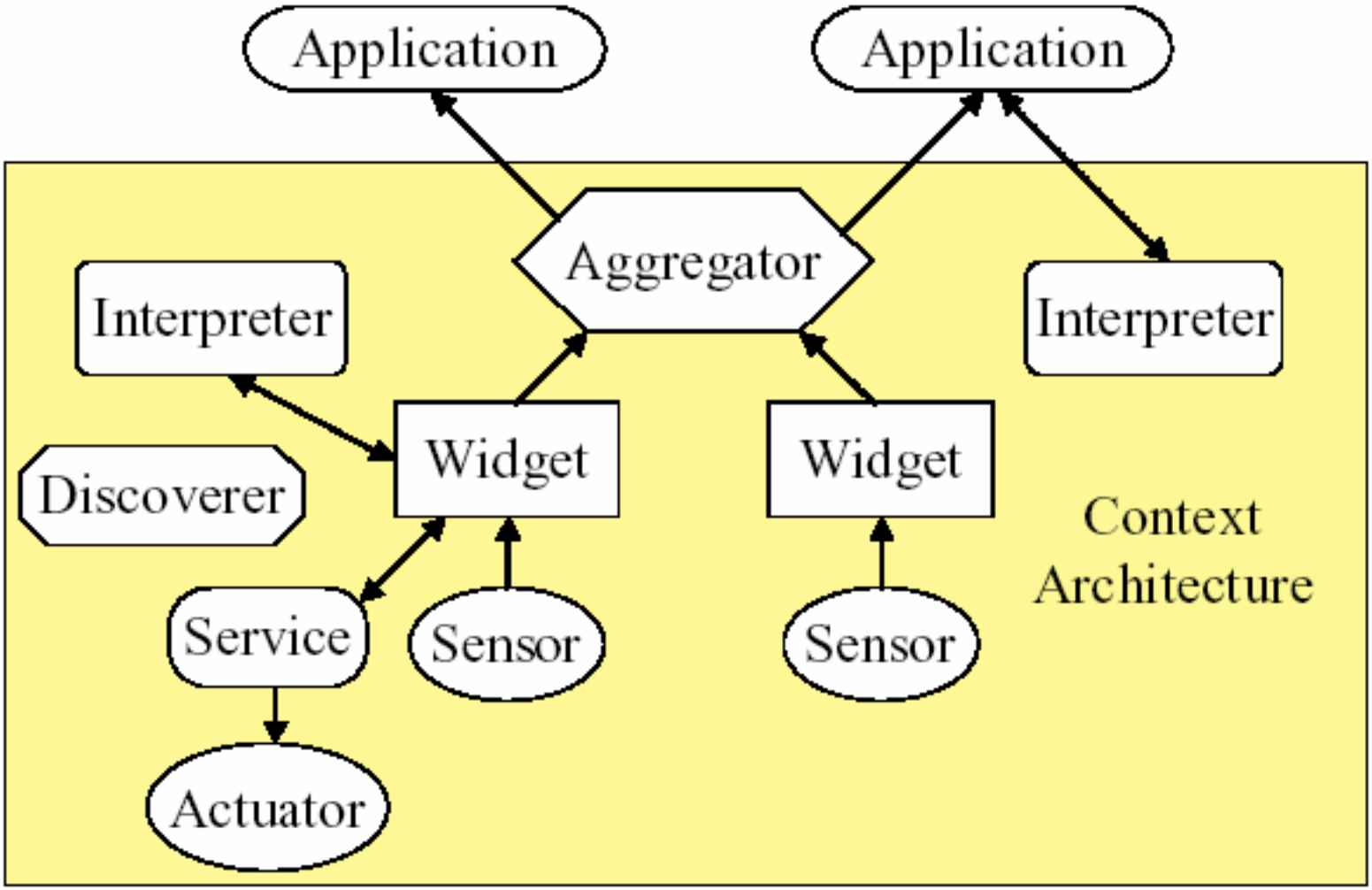
- Components in the framework that execute actions on behalf of applications
- Context service: responsible for controlling or changing state information in the environment using an actuator (output).
- Synchronous or asynchronous.

Context Abstractions - 5. Discoverers -

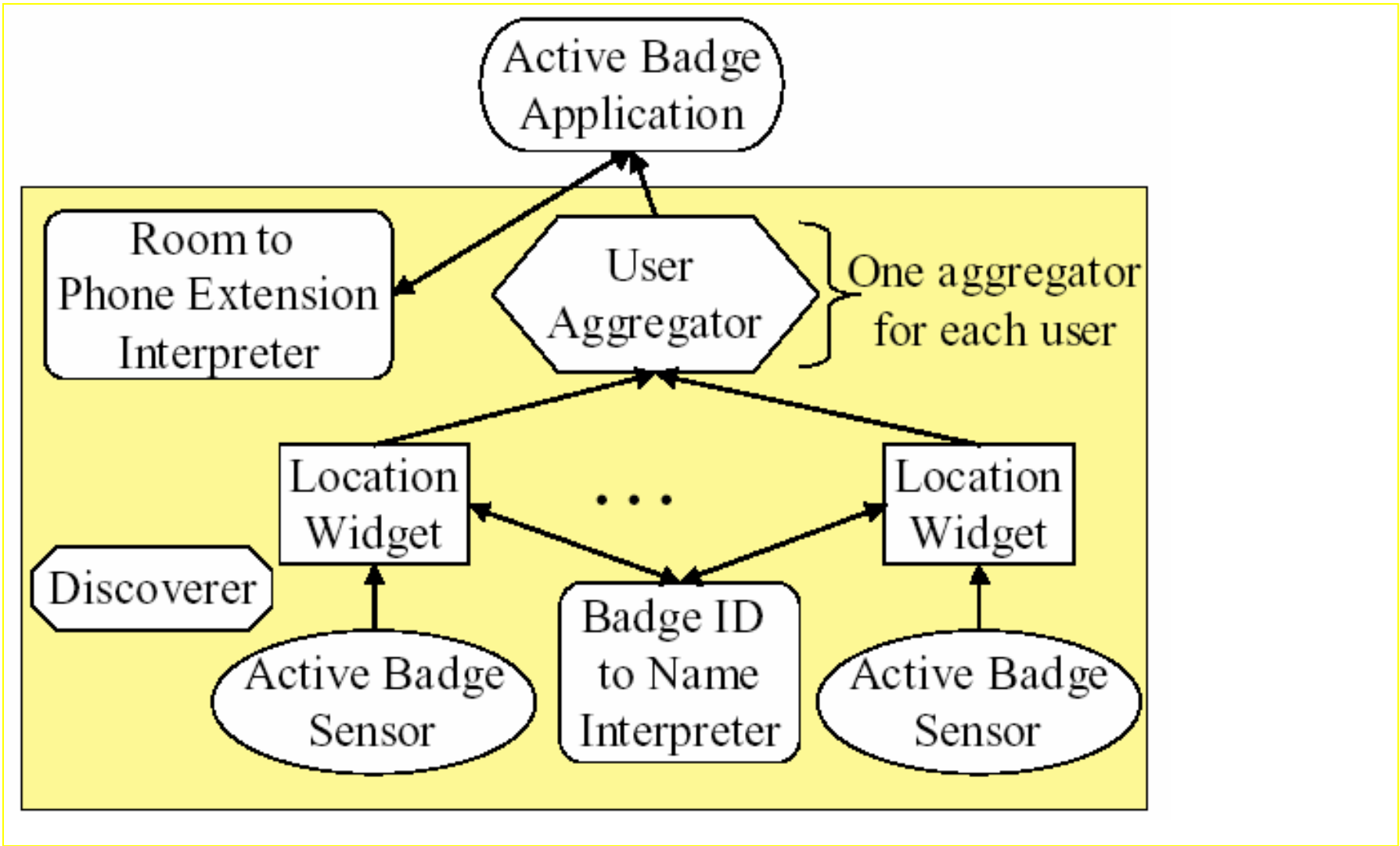
- Maintain a registry of what capabilities exist in a framework.
- When a component is started, it notifies a discoverer of its presence, capabilities, and contact information.
- Discoverer removes components from the list, when they do not respond.
- Component lookups:
  - White pages
  - Yellow pages
- Status information provided via
  - notification (publisher/subscriber)
  - querying (polling)

Using the Conceptual Framework

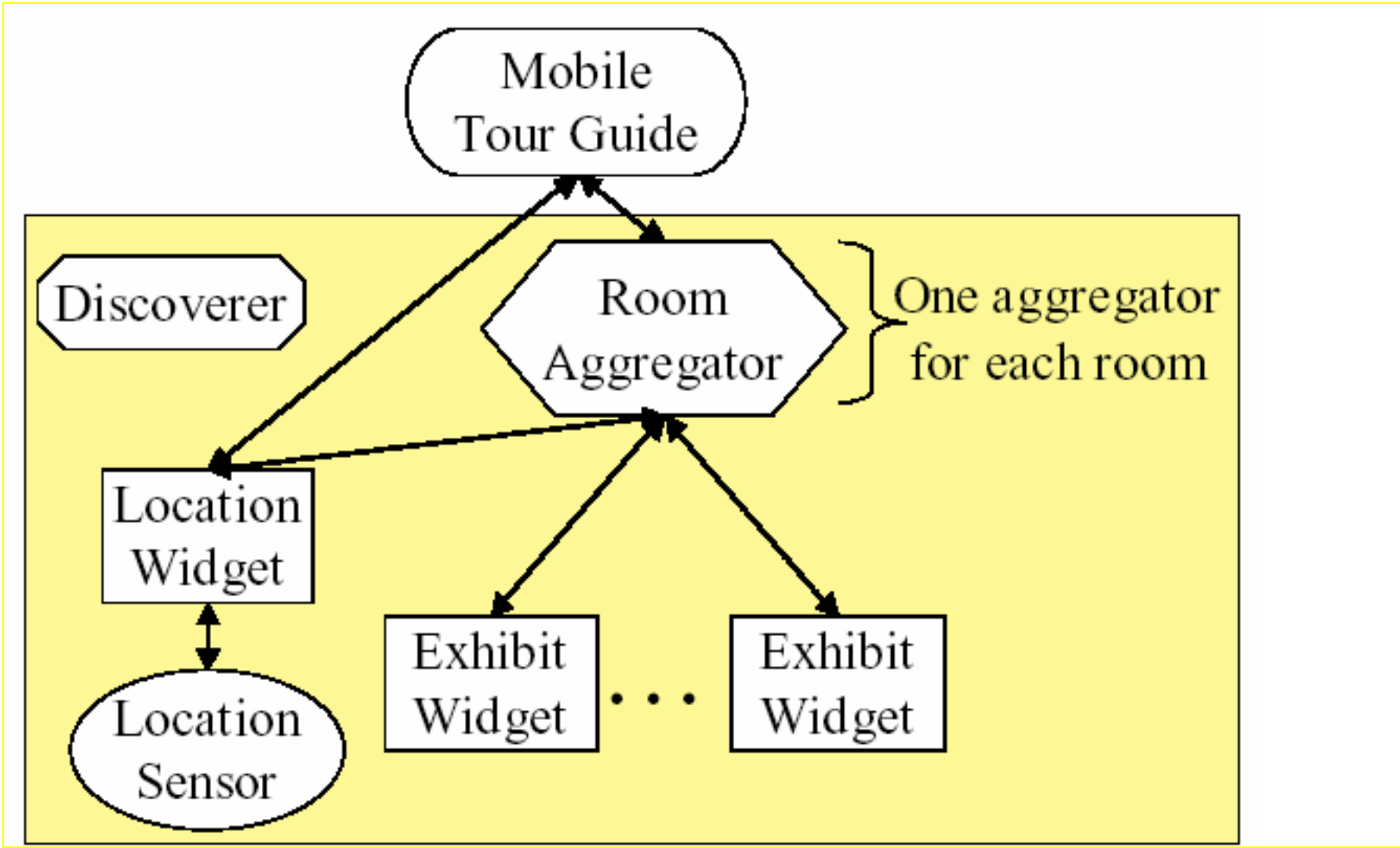
Using the Conceptual Framework



Active Badge Call-Forwarding



Mobile Tour Guide



The Context Toolkit

The Context Toolkit

Each component implemented as a single process, distributed across different processors, using peer-to-peer communication.

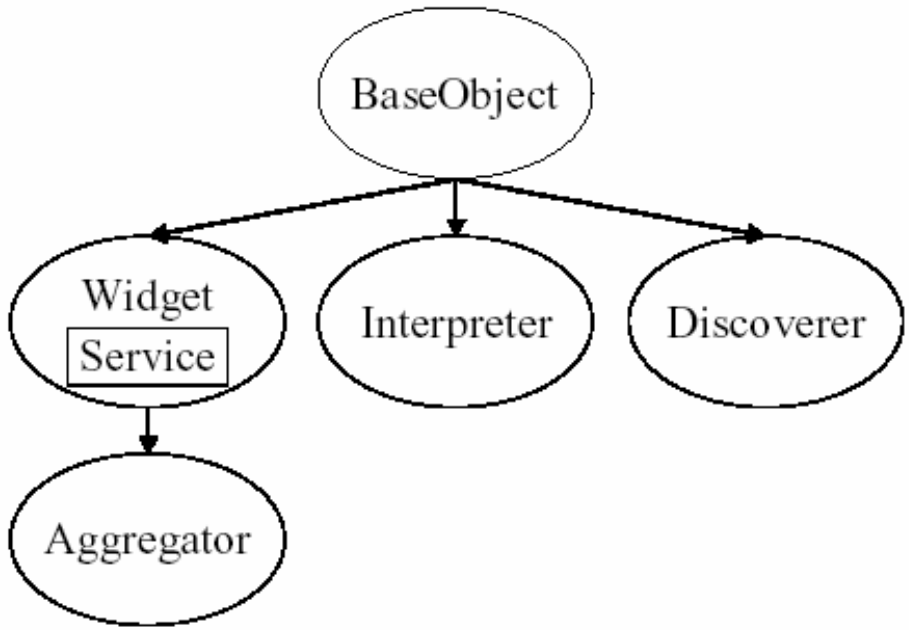
Java, C++, Frontier, Visual Basic, Python

1. Distributed communications
2. Subscriptions
3. Event handling
4. Discovery
5. Context Services

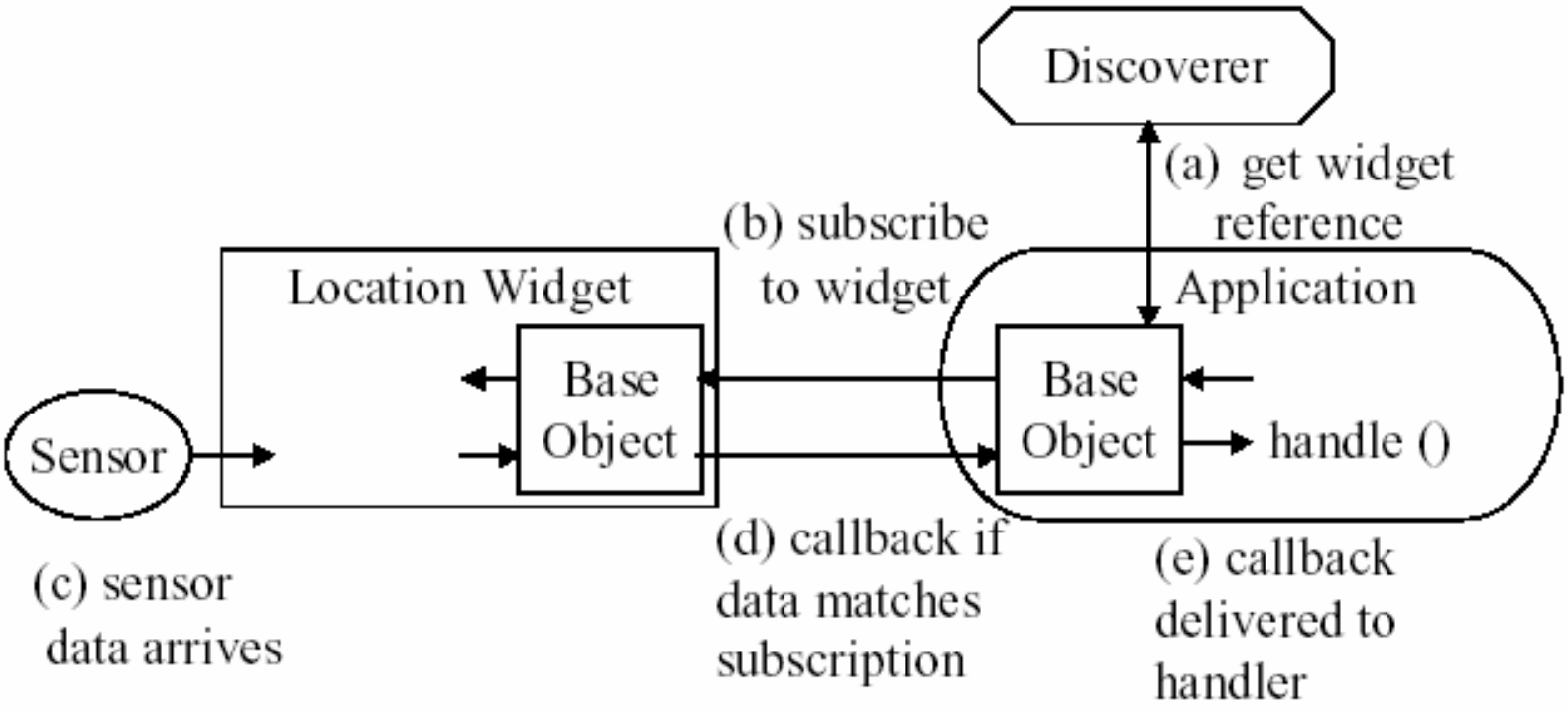
Context Toolkit - 1. Distributed Communications -

- HTTP, XML, TCP/IP (no CORBA, Java RMI !)
- Portable across unconventional, custom-built platforms (wearable computers, pagers, mobile phones, off-the-shelf custom sensors)
- Limited scalability

- Encapsulated in component ‘BaseObject’
  - handle
  - communication

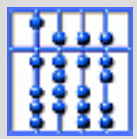


- Lists of subscribers
- Filters



- Dealing with new context data (context modification)
  - by widget:
    - detection by widget
    - timestamp
    - subscriber notification
  - by subsequent components (interpreters, aggregators):
    - context modifications
  - propagated through a distributed network
    - no unique sequence (timing), parallelism
  - all components must always be ready to process event
  - new thread per event

- Centralized component
  - simple
  - but: single point of failure
  - pluggable (easily replaceable)
- At start time all components register with the discoverer at a known network address and port
- Discoverer ‘pings’ components to verify that they are still active
- Applications don’t register with the discoverer but directly with widgets, aggregators and interpreters



- Incorporated into widgets for simpler treatment by application designer
- Widget responsible for both input (sensors) and output (actuators, services)

- The toolkit was used in a number of applications to
  - test that the abstractions were suitable to application designers

END OF PRESENTATION IN CLASS

End of Presentation in Class

Context-Aware Applications

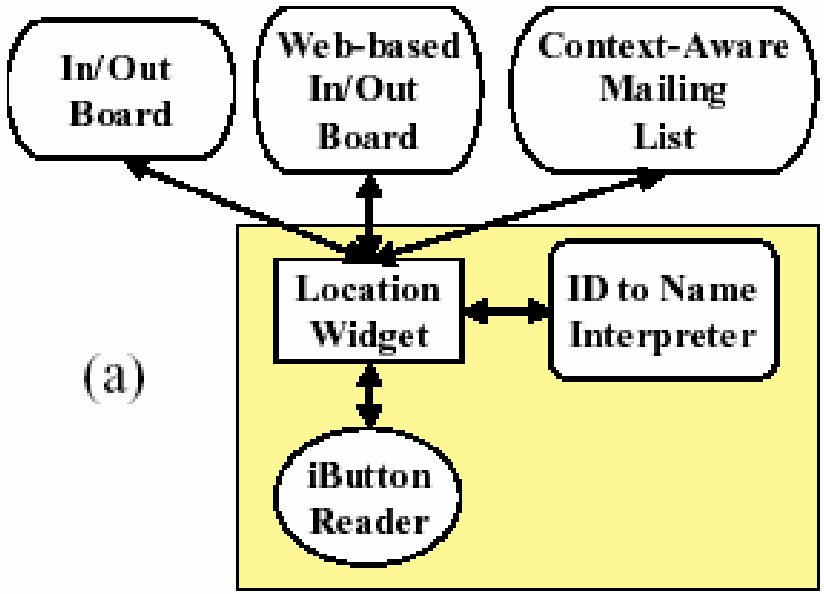
Context-Aware Applications

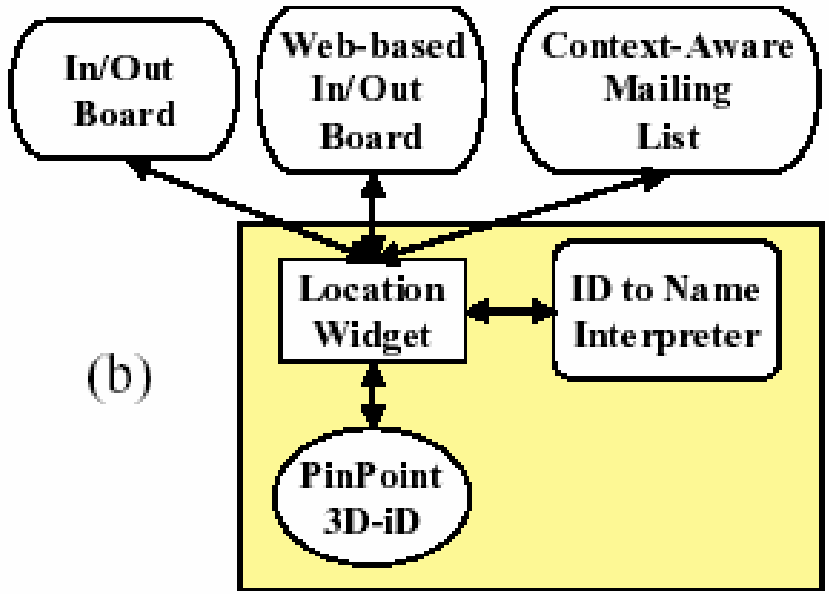
1. In/out board
2. Context-aware mailing list
3. DUMMBO: evolution of non-context-aware applications
4. Intercom: use of complex context and services
5. Conference assistant: use of complex context and systematic design of a context-aware application

Application 1: In/Out Board - Application Description -

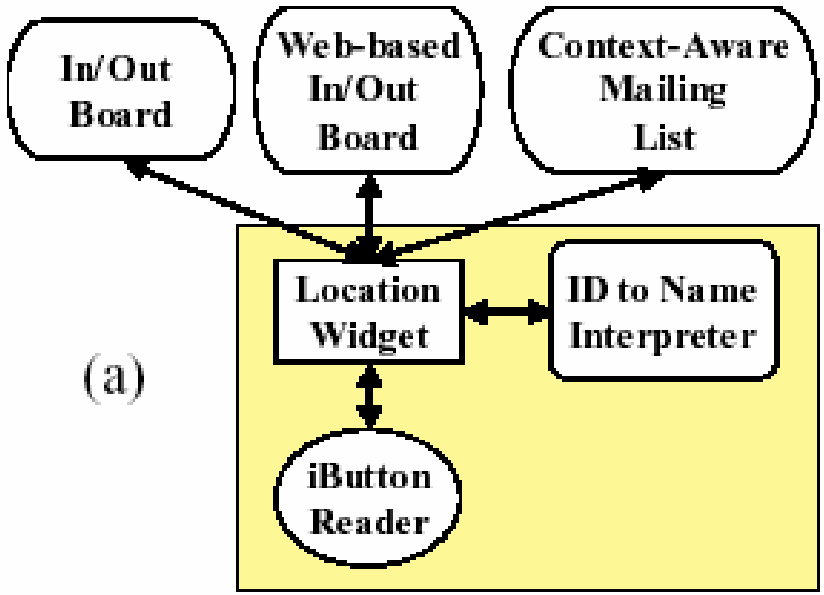
- Lab-based viewer (stationary on site)
- Web-based viewer (remote)
- Wearable viewer (mobile)
- iButton-based tracker
- RF-based tracker

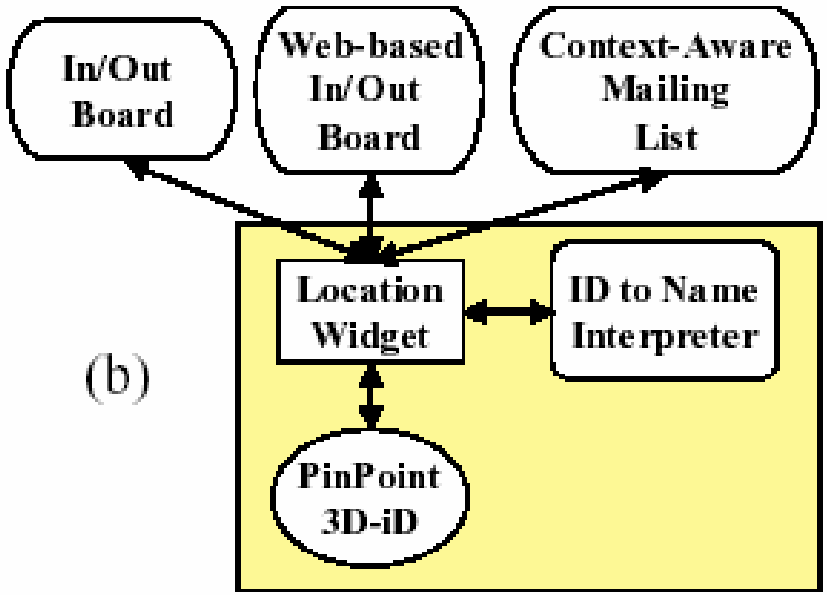
Application 1: In/Out Board - Building the Application -





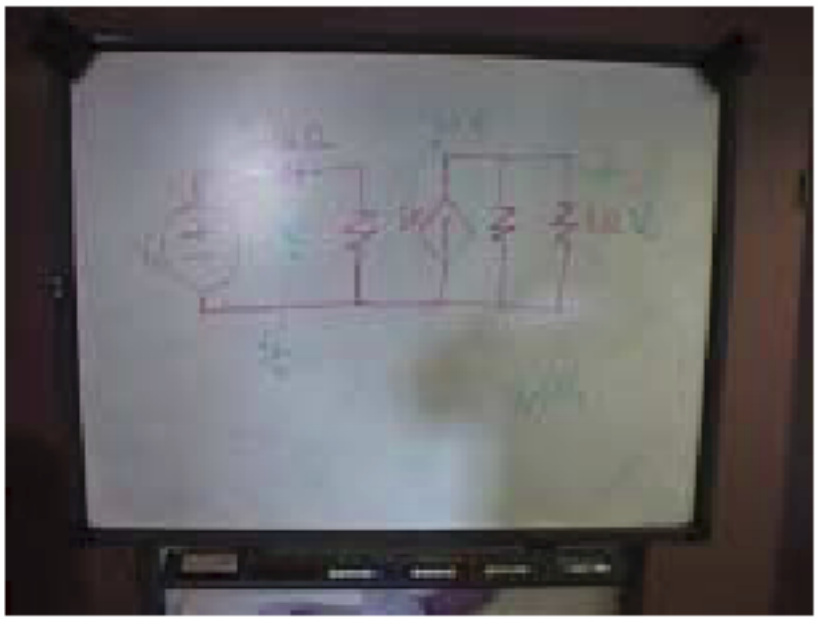
Application 2: Context-Aware Mailing List

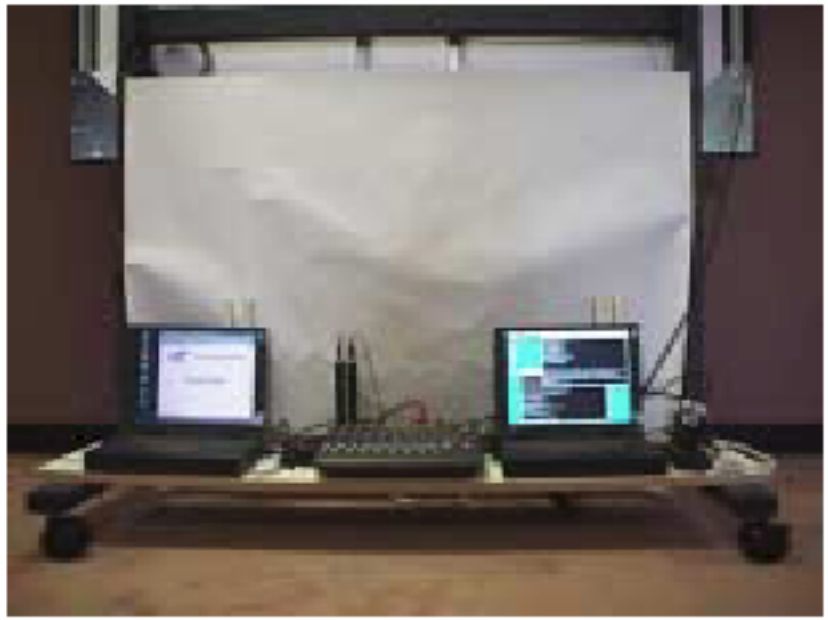




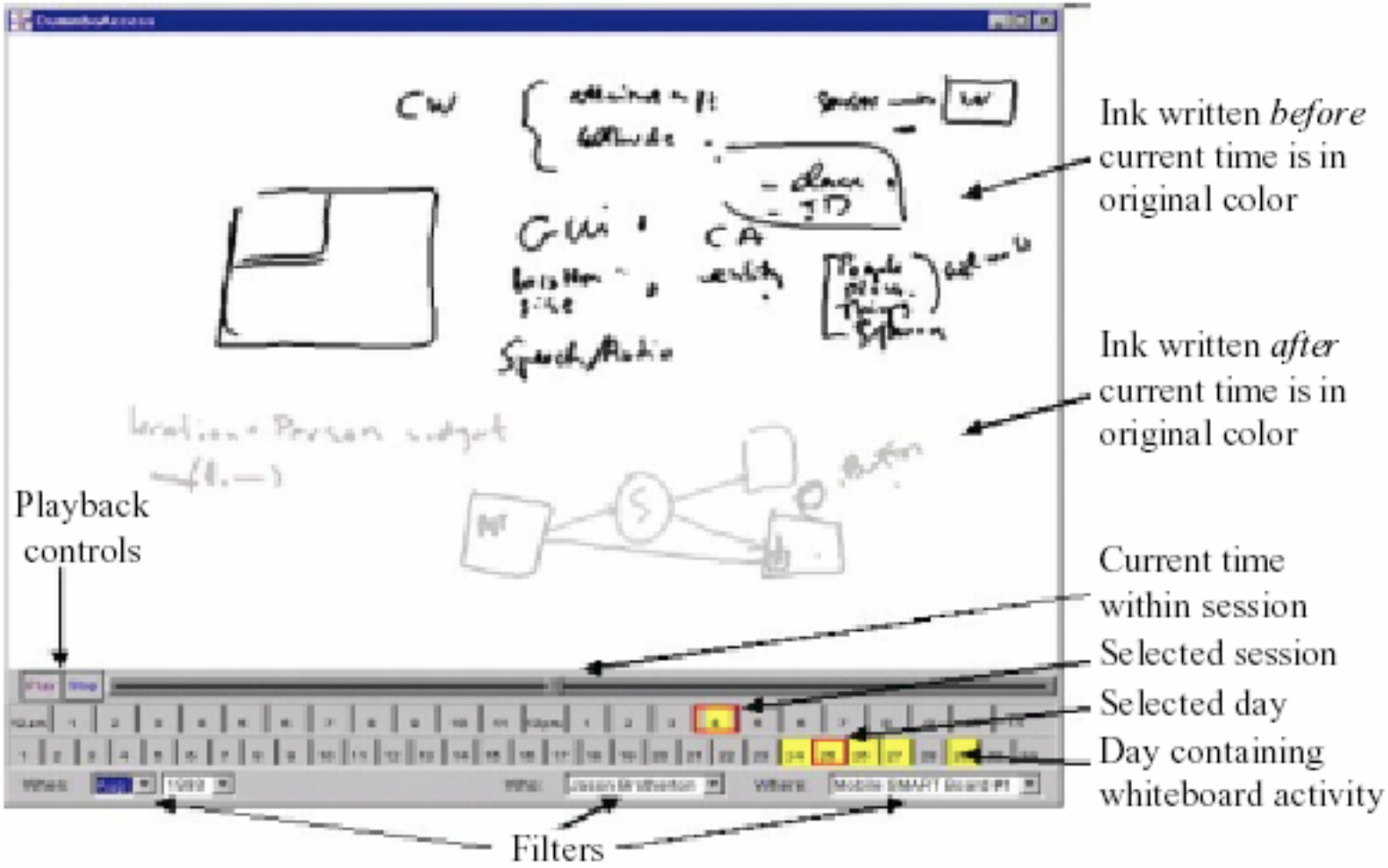
Application 3: DUMMBO - Application Description -

- Dynamic Ubiquitous Mobile Meeting BOard
- Existing application

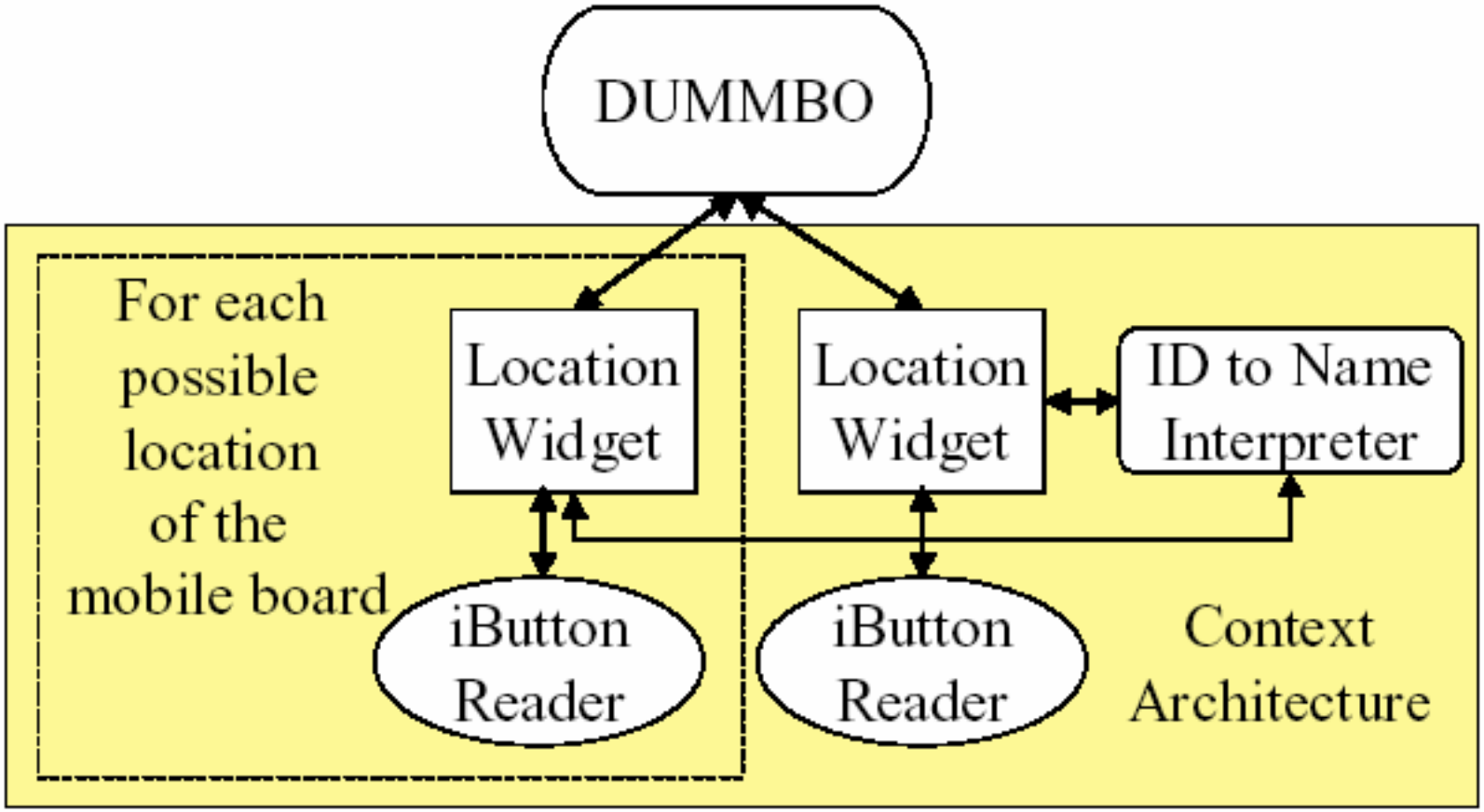




Application 3: DUMMBO - Application Description -



Application 3: DUMMBO - Building the Application -



Application 3: DUMMBO - Toolkit Support -

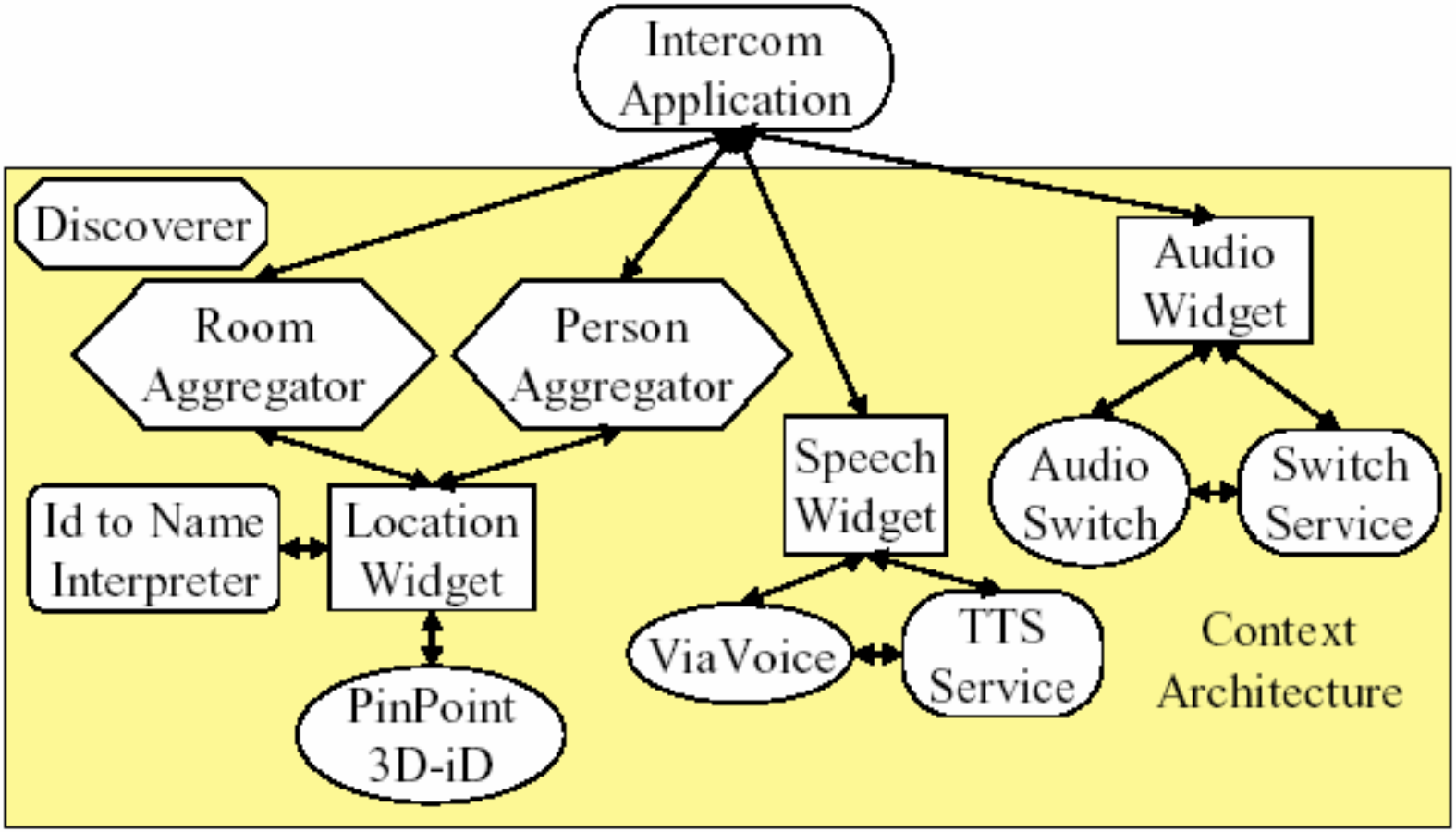
- Existing application DUMMBO, changed by original researchers (not by toolkit developers)
  - install toolkit
  - determine context widgets (multiple widgets)
  - instantiate widgets
  - handle callbacks

  - 25 lines of Java code

Application 4: Intercom - Application Description -

- One-way and two-way conversations in instrumented homes (AwareHomeResearchInitiative).
- “House, I would like to announce”
  - “Go ahead”
  - “Dinner is ready in the kitchen”
  - “Stop the intercom”
- “House, how is the baby doing?”
- “House, I want to speak to Sam”
  - “Sam is with Barbara. Do you want to talk to both?”
- Dynamic rerouting according to room changes

Application 4: Intercom - Building the Application -



Application 4: Intercom - Toolkit Support -

- Application written by researchers not involved in the developing the toolkit.
- More sophisticated contexts
  - intepreted (dep. on previous utterances)
  - aggregated
    - room-based
    - person-based

Application 5: Conference Assistant - Application Description -

- Technical conference with parallel sessions, group of attendants (friends)
  - General: personalized schedule, location of rooms
  - Room-based: ongoing session, current presentation, URLs, miniature slides, audio, video
  - Personalized: note-taking, marked-up slides, rating of presentation
  - Group-based: location of friends, attended sessions, current interests in sessions
  - Post-conference: access to conference web-site to retrieve slides, videos etc. of attended/unattended, personal notes

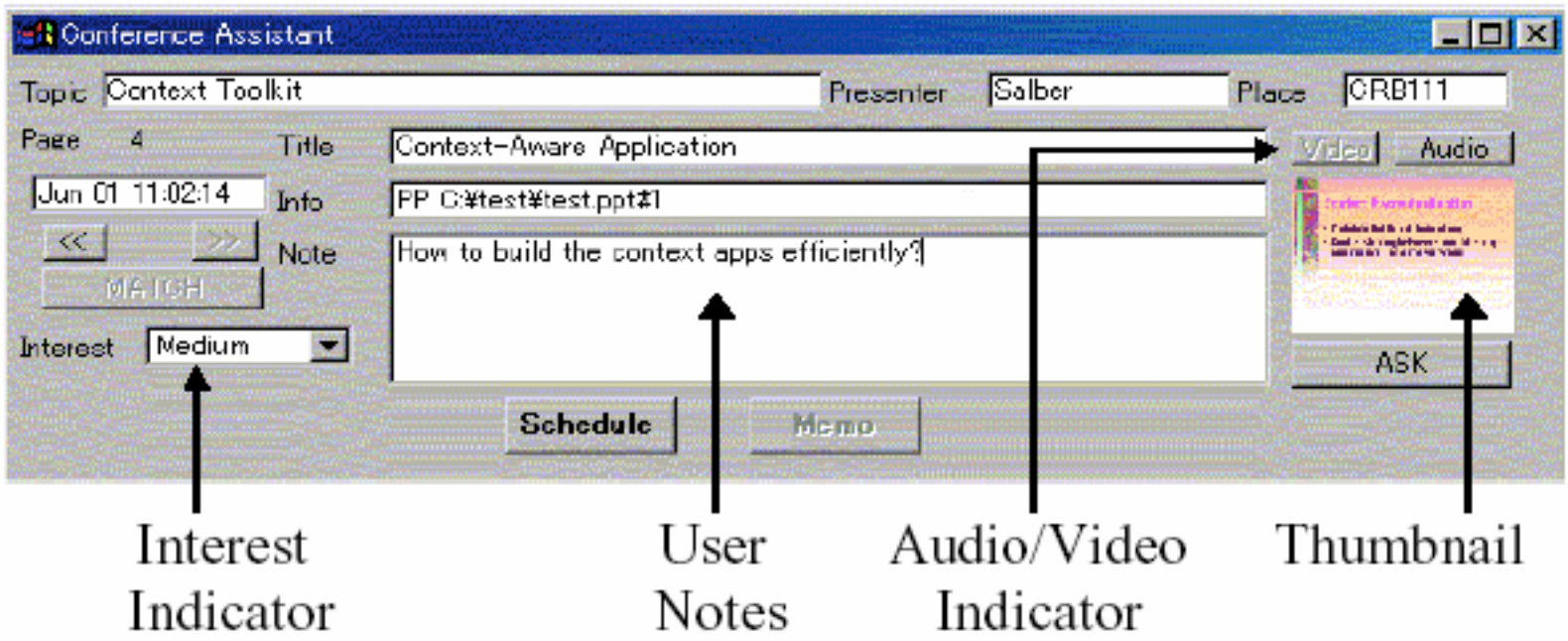
Application 5: Conference Assistant - Application Description -

- General: personalized schedule, location of rooms

9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00
Context Toolkit	Machine Learning	Human Motion	Digital Desk	Smart Floor	ERRATA	VR Gorilla	Input Devices
VR Workbench	C2000	Personal Pet	IMAGINE	Mastermind	Urban Robotics	Sound Toolkit	Head Tracking
VR Gorilla	Pepe	Ubicomp Apps	Sound Toolkit	ERRATA	C2000	Input Devices	Smart Floor

Application 5: Conference Assistant - Application Description -

- Room-based: ongoing session, current presentation, URLs, miniature slides, audio, video
- Personalized: note-taking, marked-up slides, rating of presentation

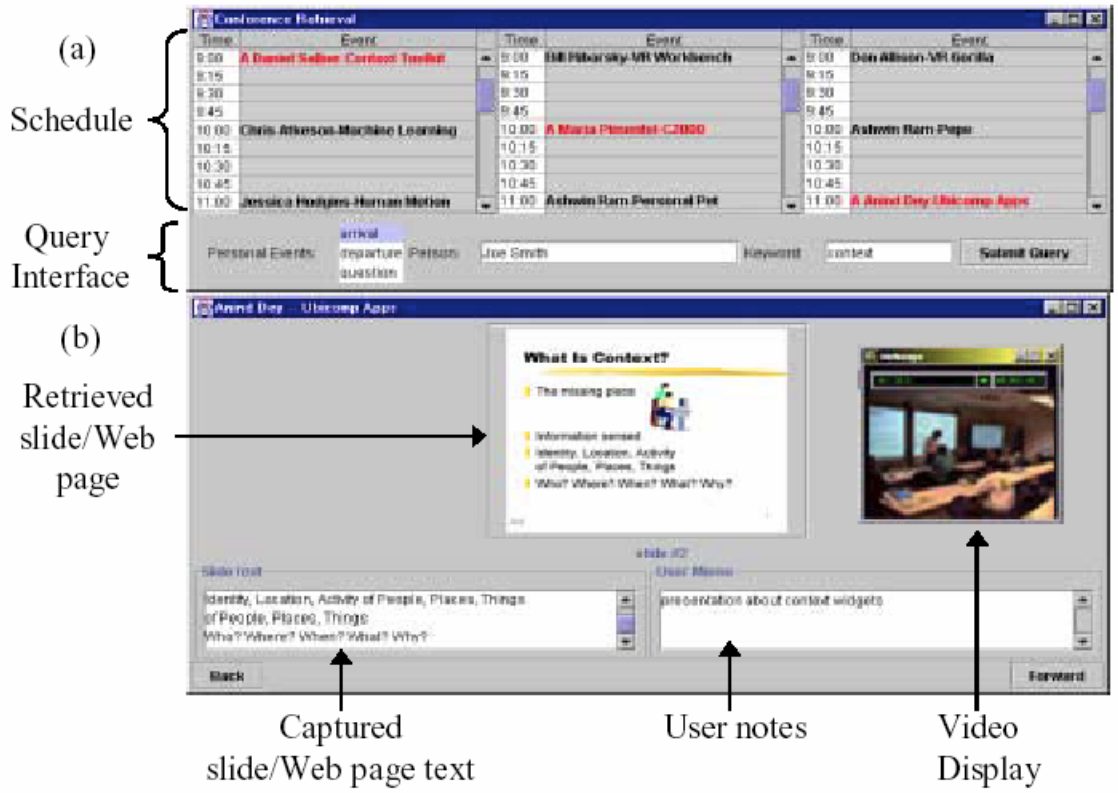


Application 5: Conference Assistant - Application Description -

- Group-based: location of friends, attended sessions, current interests in sessions

Application 5: Conference Assistant - Application Description -

- Post-conference: access to conference web-site to retrieve slides, videos etc. of attended/unattended, personal notes



- Entities and their Context Attributes

Entities	*Attributes*
People	
Attendee	Name (Contact info, interests, colleagues) Room and times of arrival/departure Level of interest
Presenter	Name
Places	
Room	Name
Things	
Presentation	Room, time
Slide	Number, title, thumbnail, time
Web page	URL, title, thumbnail, time
Question	Related slide or Web page, time
User notes	Related slide or Web page, time

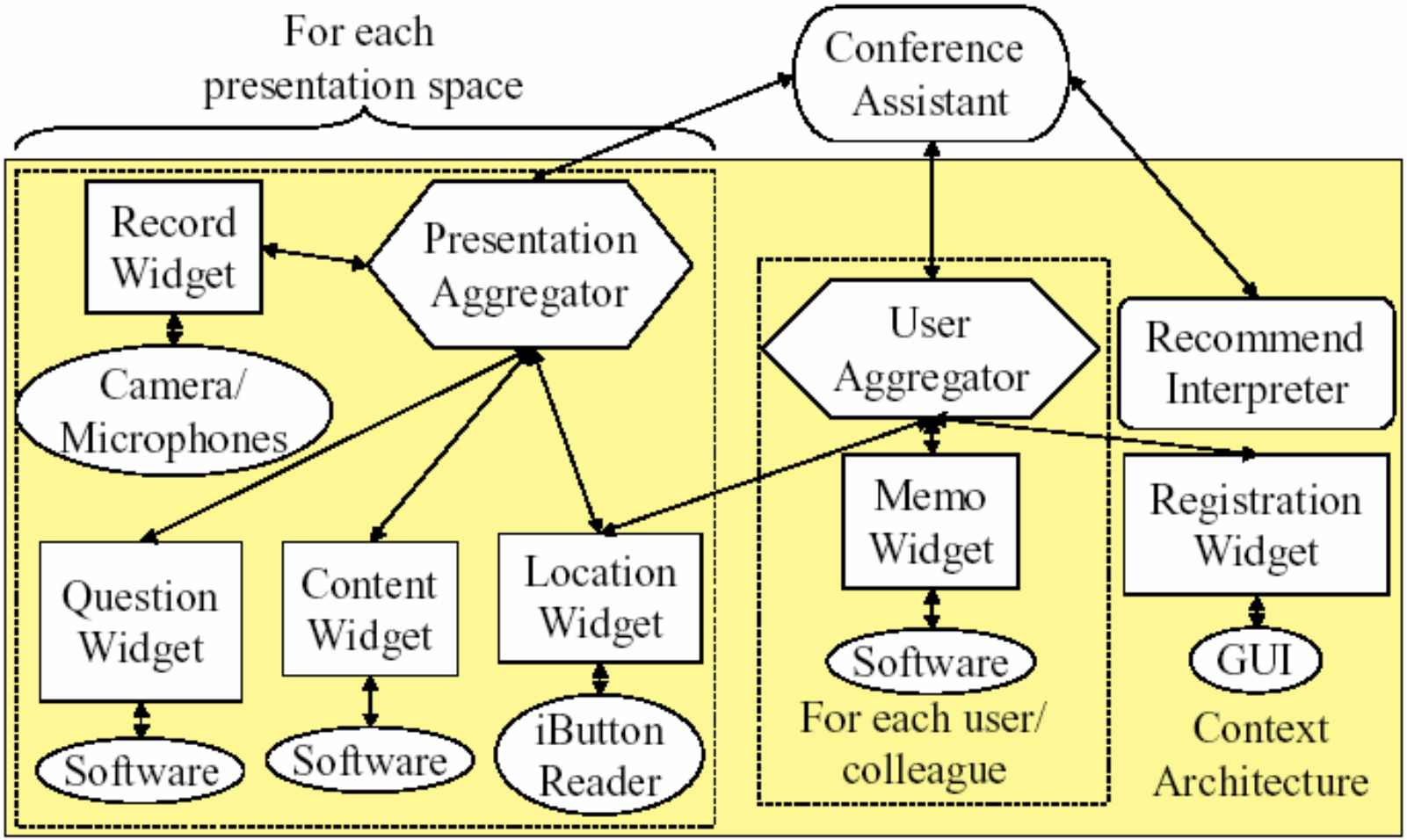
Application 5: Conference Assistant - Applying the Design Methodology -

- Quality of Service Requirements
  - user location (3 non-contiguous rooms, low resolution)
    - high coverage
    - mediate frequency, timeliness
  - current slide, web page
    - high timeliness (instant presentation upon room entry)
  - level of interest of users
    - high coverage, availability
    - low timeliness

Application 5: Conference Assistant - Applying the Design Methodology -

- Sensors
  - Current slide, web page
    - inter-application communication (COM/DCOM)
  - Level of interest
    - manual user input
  - User location
    - Active Badges (Olivetti) [IR-transmitting badges]
    - iButtons (Dallas Semiconductor) [button devices]
    - 3D-iD (Pinpoint) [RF active tags]

Application 5: Conference Assistant - Building the Application -



Issues

- Representation and acquisition of context
- Privacy
- Ambituity in context data
- Higher-level programming abstractions

Requirements and Conceptual Framework for Handling Context

- Guiding principle:
  Separation of concerns between
  - context acquisition and
  - use of context
- Abstractions to
  - acquire
  - collect
  - manage

  context

in an application-independent fashion.

- Analog to the development of GUIs.

- Software Design

Aggregators
Attendee
Presenter
Room
Presentation
Slide
Web page
Widgets
Name (Attendee)
Room (Attendee)
Times of departure/arrival (Attendee)
Level of interest (Attendee)
Name (Presenter)
Name (Room)
Room (Presentation)
Number (Slide)
Title (Slide)
Thumbnail (Slide)
URL (Web page)
Title (Web page)
Thumbnail (Web page)
Current slide (Question)
Time (Question)
Current slide (User note)
Time (User note)
Interpreters
Tag ID to Name (Attendee)
Zone ID to Room (Attendee)
iButton ID to Name (Attendee)

Component	Type	Responsibility
Attendee	Aggregator	Aggregates information about a user
Presentation	Aggregator	Aggregates information about a presentation
Registration	Widget	Acquires contact info, interests, and colleagues
Location	Widget	Acquires location and arrivals/departures of users
Content	Widget	Acquires title, URL/slide number, thumbnail from slide or Web page
Question	Widget	Acquires audience questions and relevant presentation info
Memo	Widget	Acquires user's notes and relevant presentation info
Record	Widget	Acquires audio/visual presentation info
Name	Interpreter	Transforms iButton ID into user name