Lecture 9

Context-Aware Technologies, Systems and Applications

- Features of Context-aware Computing
- Architectures of Context-aware Computing
- Design of Context-aware Systems/Applications
- Examples of Context-aware Applications

Context & Context-Aware Computing

- Dey & Abowd: "Context is any information that can be used to characterize the situation of an entity"
  - Entity: person, place, object that is considered relevant to interaction betw. a user & an application, including the user & application themselves.
- Dey: "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."
- Moran,Thomas & Dourish: "Context-aware computing strives to acquire and utilize information about the context of a device (application) to provide services that are appropriate to the particular people, place, time, events."

History of Context Aware Computing

- The early 1990's saw the arrival of context-aware computing with the introduction of small mobile computing devices (by Schilit, et al, led by M. Weiser).
- Olivetti Lab's Active Badge (1992) used infrared communication between user badges and sensors placed in a building to monitor movement of users for forwarding calls.
- PARC's PARCTab system (1993) uses location information to allow applications to adapt to user's environment.
- Context-aware applications serve as tour guides by presenting information about the user's current environment. Ex: Cyber Guide from Georgia Tech (1996).
- In the later 90's, frameworks built to support context-aware applications began to be developed. Ex: Georgia Tech's Context Toolkit (1999).
- Since 2000, more and more researches and applications

Examples & Classifications of Context

- Identity: user characters, needs
- Spatial: location, orientation, speed
- Temporal: date, time of day, season
- Environmental: temperature, light, noise
- Social: people nearby, activity, calendar
- Resources: nearby, availability, energy
- Computation: CPU, OS, memory, interfaces
- Network: wire/wireless, bandwidth, error rate
- Physiological: blood pressure, heart rate, tone of voice
- Psychology: preference, emotion, tiredness, …

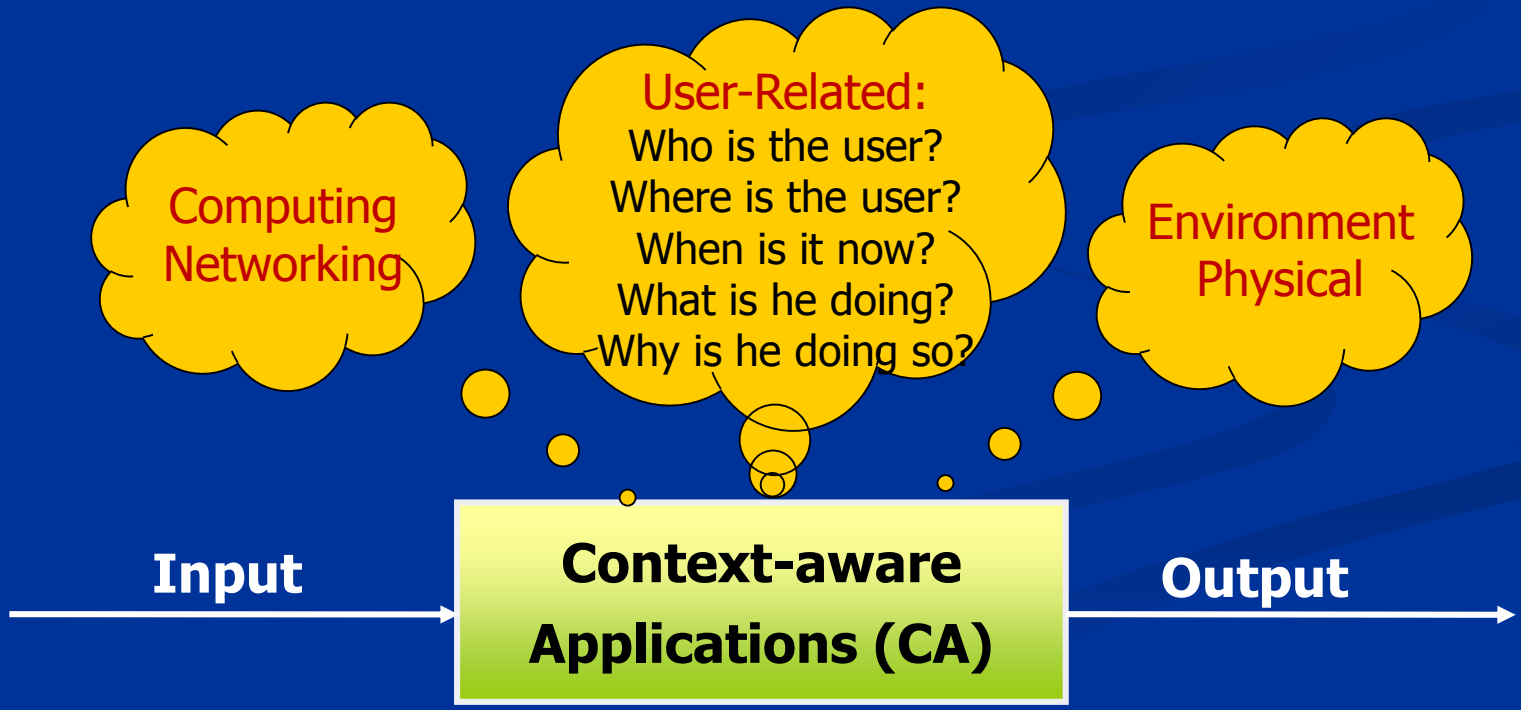
- External Context (physical)
  - Measured by sensors, Ex: location, light, sound, pressure, etc.
- Internal Context (logical)
  - Captured from user's interaction, Ex: user's goal, emotion, etc.

Another Classification of Context

- Computing Context
  - Network connectivity
  - Communication cost
  - Comm. bandwidth
  - Nearby resources
- User Context
  - User profile/preference
  - User mood/behavior
  - Other's presence
- Environmental Context
  - Lighting
  - Noise level
  - Traffic conditions
  - Weather
- Physical Context
  - Time, Date
  - Location
- Context History – Stored Context of Past
  - Computing, User, Environment, Physical Context

User-Related: 5W Context Awareness

- Who: Deals with identifying current user and object recognition.
- Where: Deals with location identification of user, object, service, …
- When: Deals with temporal aspects of past, present & future
- What: Deals with identifying activities of user or object
- Why: Deals with subtle context such as user's need, emotion, …



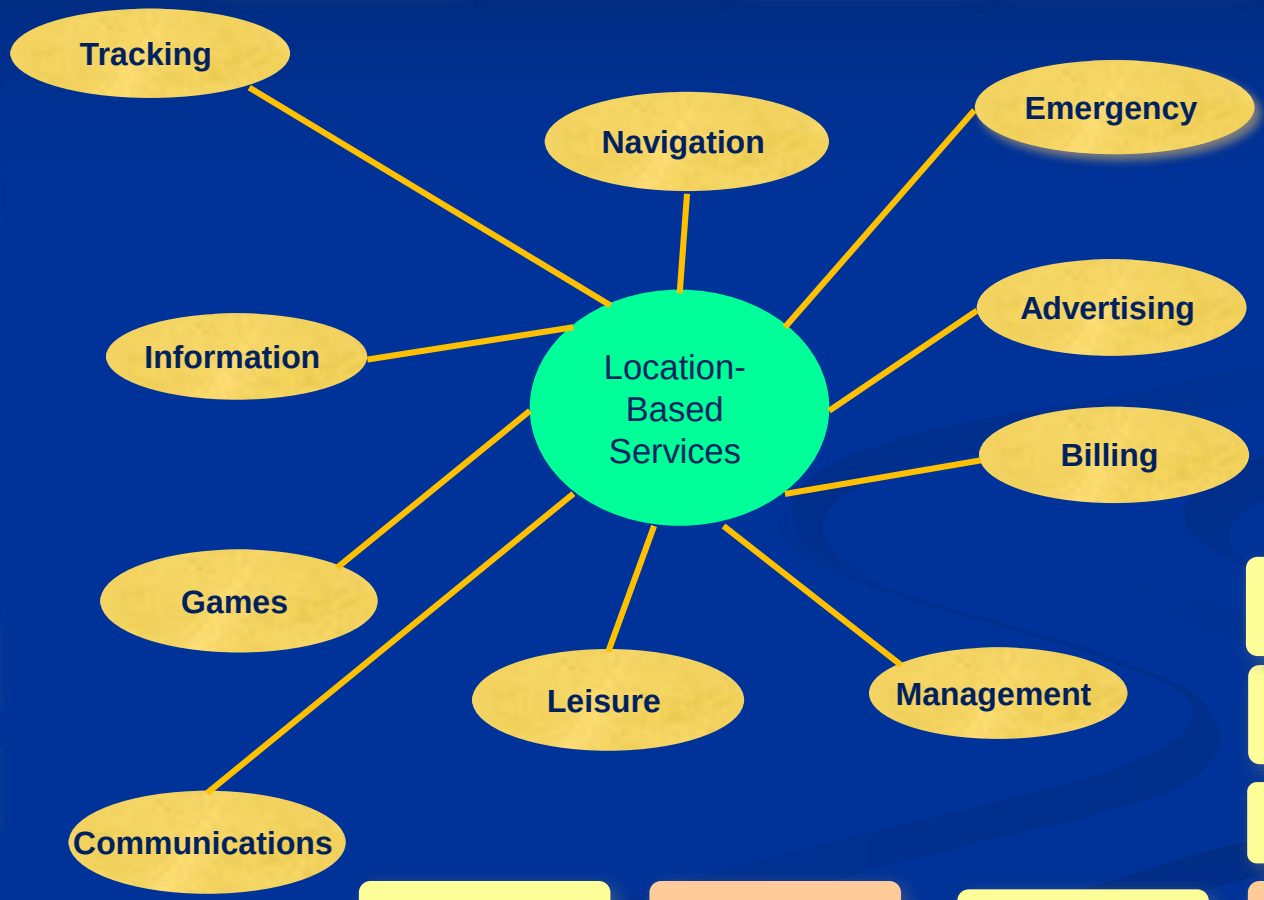
Various Context-Aware Computing

- Location-Aware Computing
- User-Aware Computing
  → Preference/Need/Intention/Emotion Aware Computing
- Energy-Aware / Power-Aware Computing
- Resource-Aware Computing
- Service-Aware Computing
- Network-Aware Computing
- Environment-Aware Computing
- Situation-Aware Computing
- Safety/Security/Privacy-Aware Computing
- Chance/Opportunity-Aware Computing
- . . .

Location-based Applications

- Finding services based on location
  - physical services (stores, restaurants, ATMs, ...)
  - electronic services (hot spots, printers, ...)
- Using location to improve (network) services
  - incoming or outgoing communications adapts to location
- Using location to provide information
  - tourist guides
  - advertisements
- Making others aware of user location
  - presence (individual)
  - popularity, movement (group)
- Security
  - grant access based on user's location

Location-based Applications

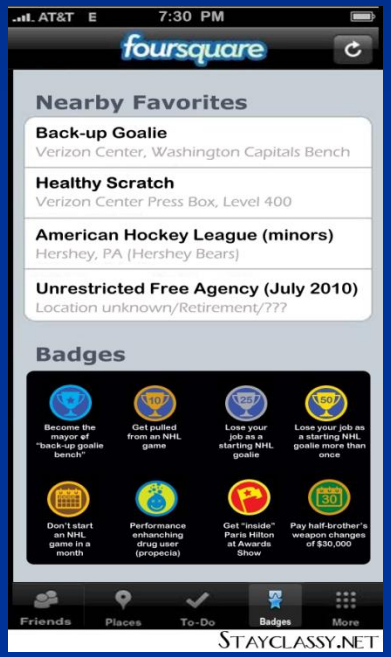


Location-based Social Networks

LBSN Software/Apps

Loopt

Foursquare



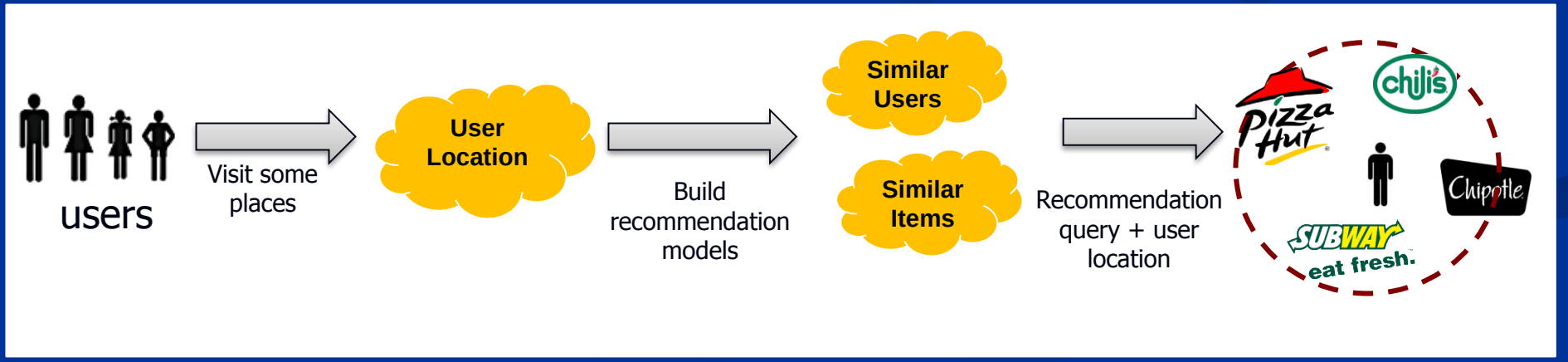
Facebook Places

Dianping

- Users share photos, comments or check-ins at a location
- Expanded rapidly, e.g., Foursquare gets over 3 million check-ins every day (2011)

LBSN Recommendation

- Location Recommendations in LBSN
  - Recommend locations using a user's **location histories** and **community opinions**
  - Location bridges gap between **physical world** & **social networks**
- Existing Solutions
  - Based on item/user collaborative filtering
  - Similar users gives the similar ratings to similar items



Context-Aware Applications

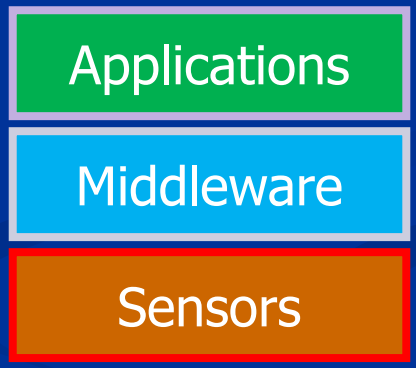
-- CA applications shown in video below --

- Context Aware Dynamic Lighting
- Gimbal Context Aware Platform
- Outdoor Location-aware Application
- Indoor Location-based Application
- Location-Aware Music Album
- Location Aware Services

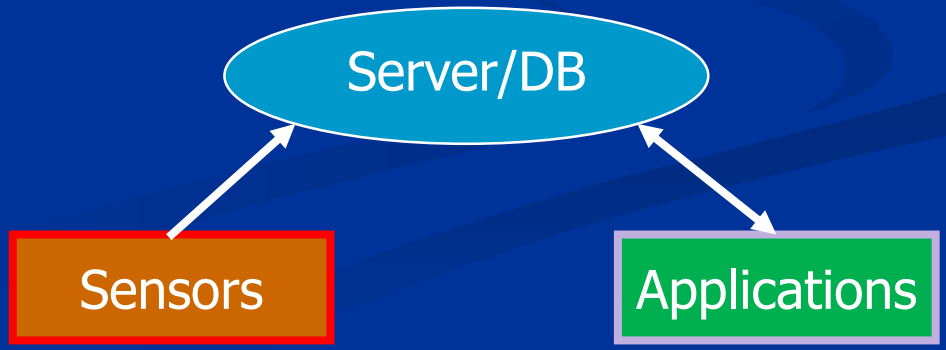
Context-Aware Computing Architecture

- Context Direct Processing
  - Tightly coupled
  - No extensibility

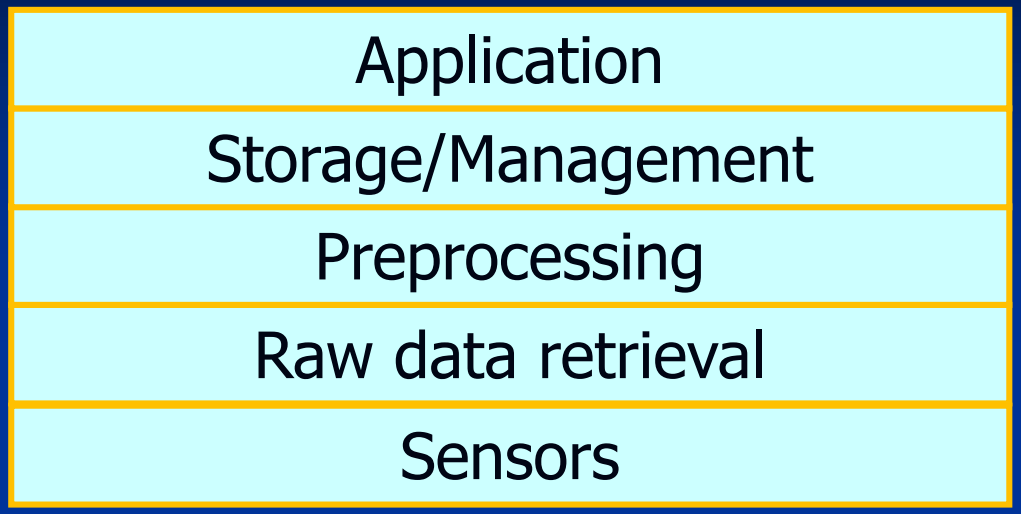
- Context Middleware
  - Hiding low-level sensing details
  - Extensible/Scalable



- Context Server
  - Multiple remote accesses
  - Appropriate protocols, QoS



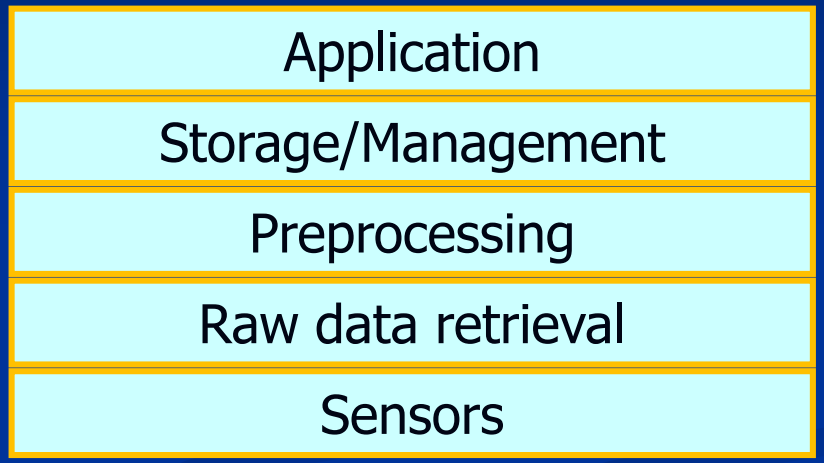
Abstract Layer Architecture



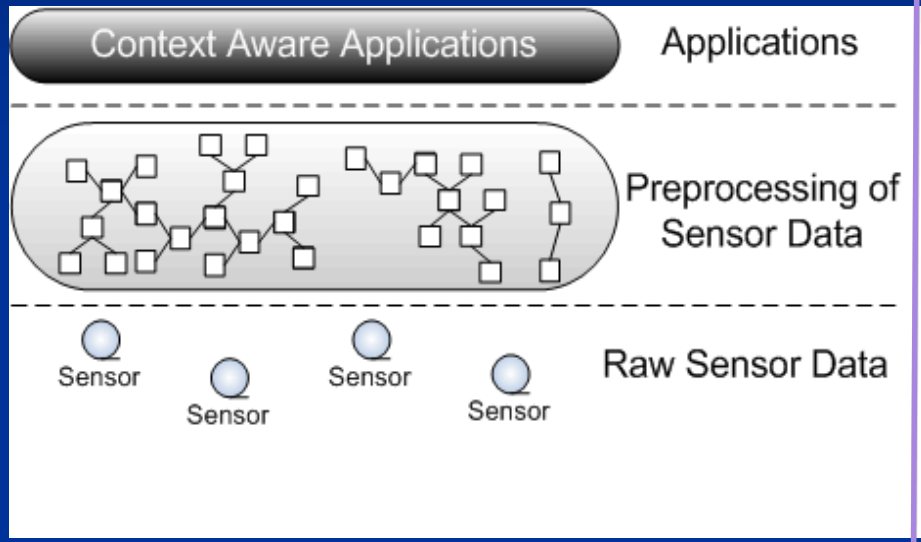
- Sensors
  - Physical sensors
    - sensor, camera, microphone, accelerometer, GPS, biosensors, etc.
  - Virtual sensors
    - From software: browsing an electronic calendar, a travel booking system, emails, mouse movements, keyboard input, bandwidth, etc.
  - Logical sensors
    - Combination of physical and virtual sensors with additional information (e.g. context history) from databases

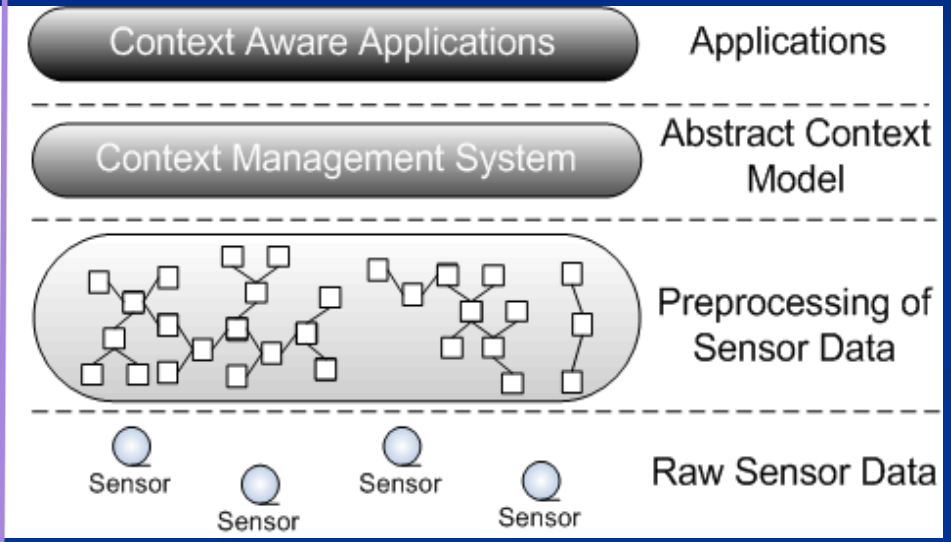
Abstract Layer Architecture (Cont)

- Raw data retrieval
  - Drivers and APIs
  - Query functionality
  - Exchangeable
- Preprocessing
  - Reasoning and interpreting
  - Extraction and quantization
  - Aggregation and compositing
- Storage/Management
  - Public interface to the client
  - Synchronous (pull/polling) and asynchronous (push/subscription)
- Applications
  - Actual reactions on different events and context-instances are implemented to provide desired information/services



Abstract Layer Architecture (Cont)





- Simple/Specific Context-Aware Architecture

→Context direct processing/use

E.g., Active Badge, Cyber Guide

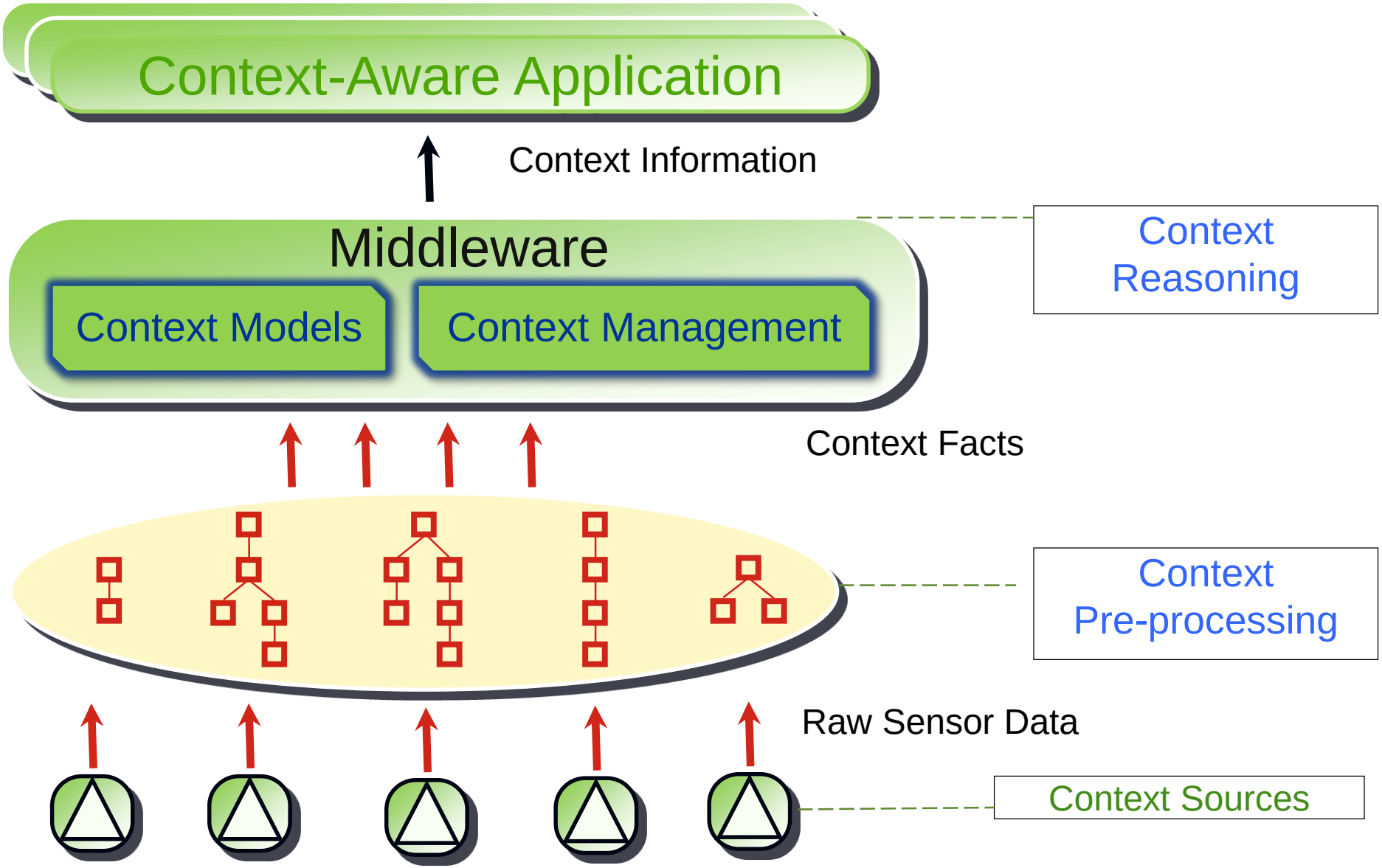
- Complex/General Context-Aware Architecture

→ Context Middleware

→ Context Server

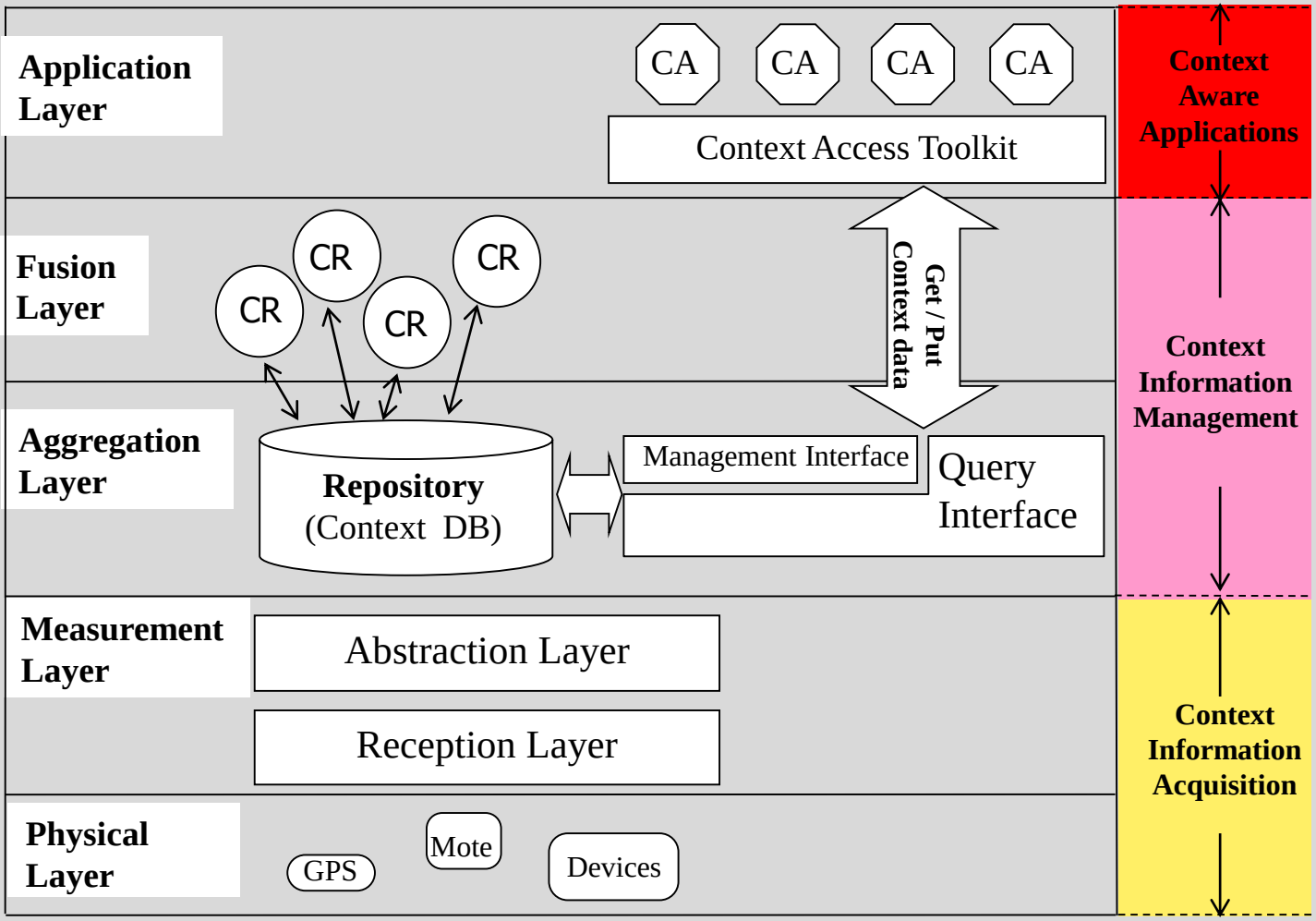
Architecture of Context-Aware System (1)

NICTA



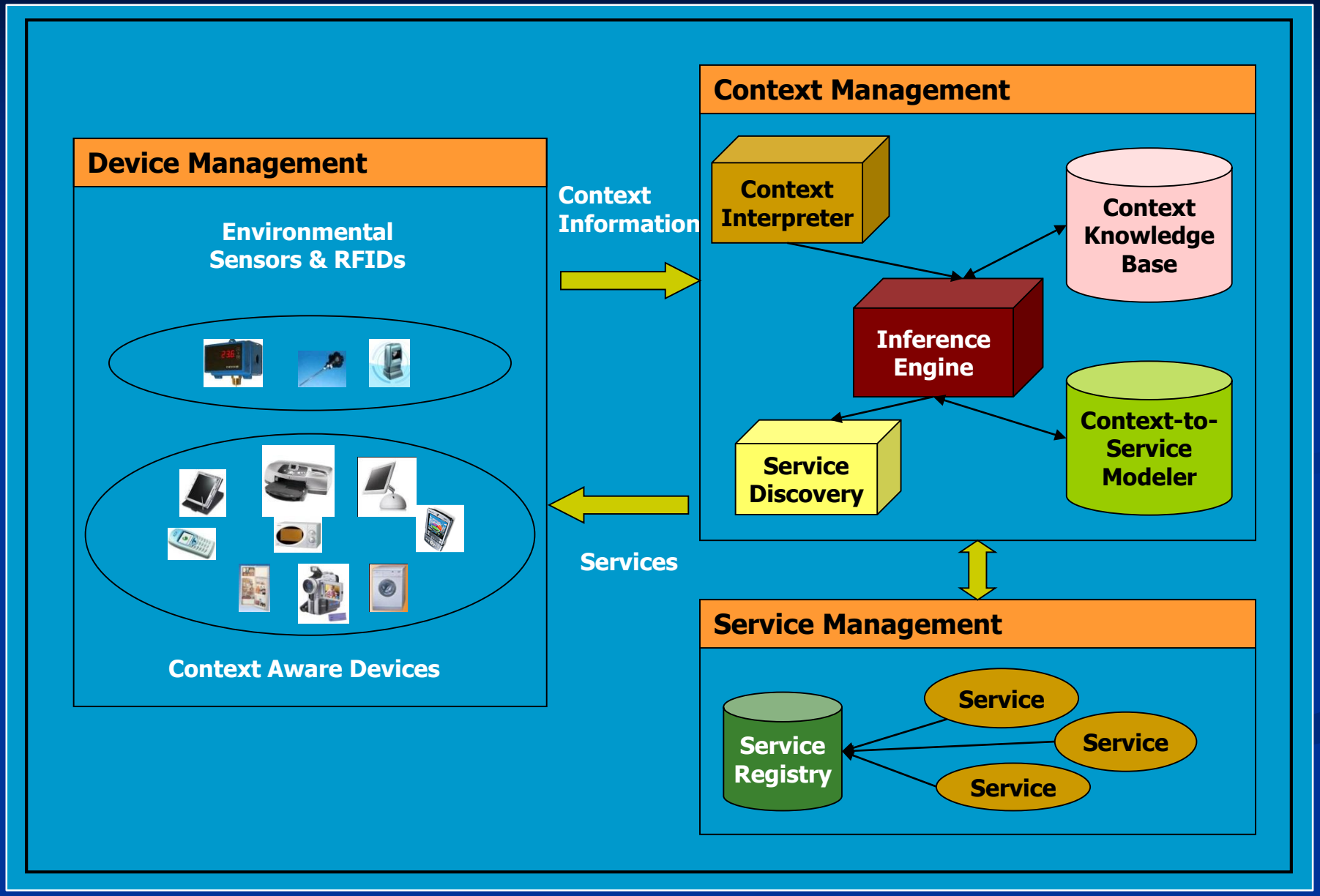
Architecture of Context-Aware System (2)

From SRIT



CR: Context Refiner; CA: Context-Aware Application

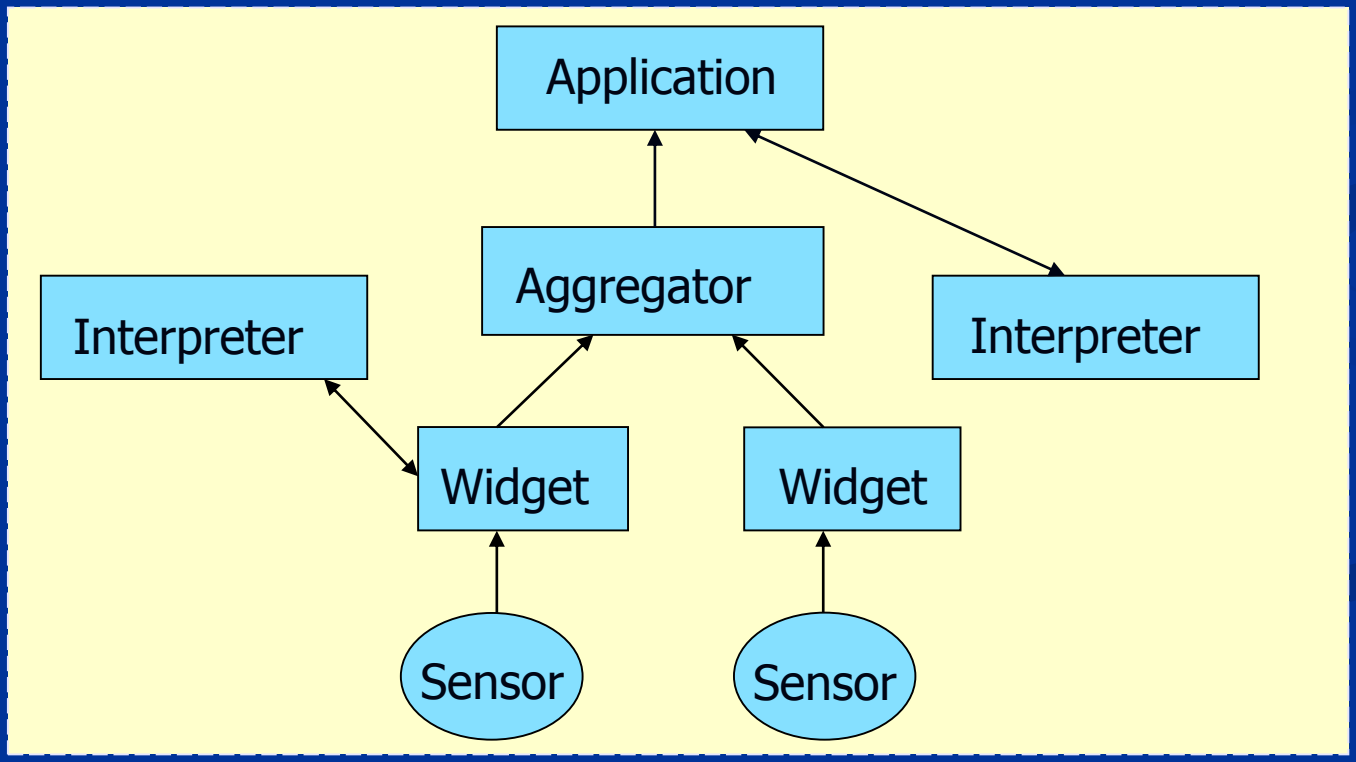
Architecture of Context-Aware System (3)



Context Toolkit Architecture

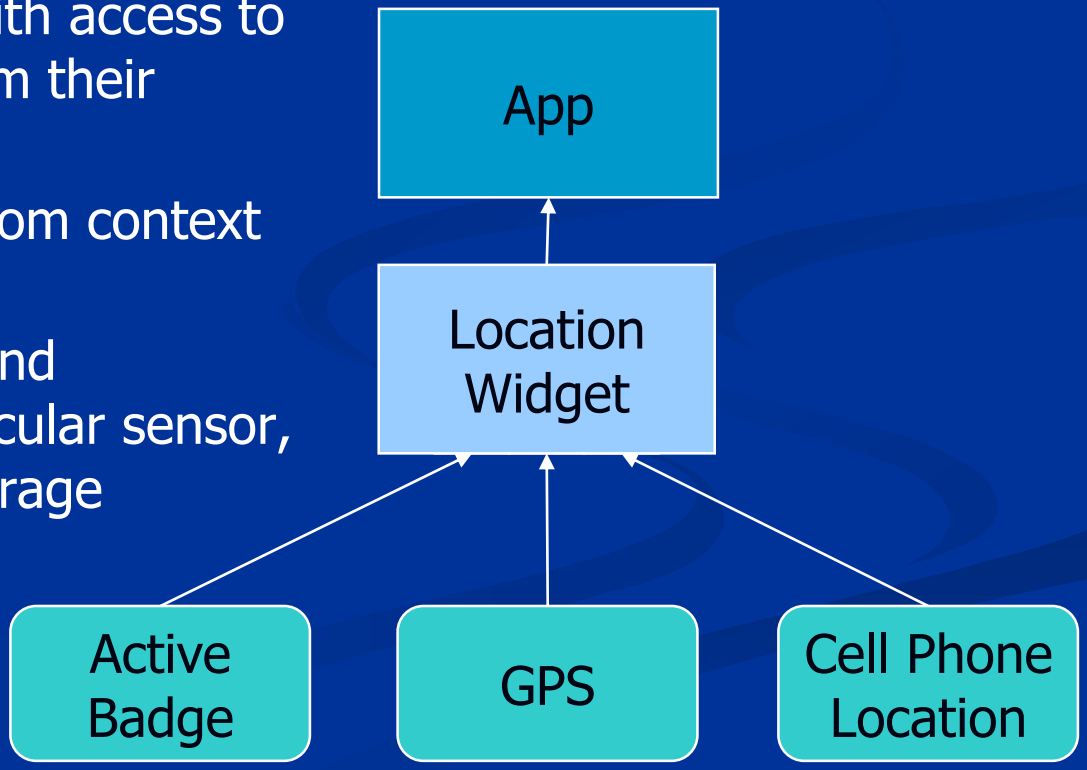
- Toolkit for distributed context-aware apps
  - Framework for acquiring & handling context via standard components
- Three key abstractions
  - Widgets, Interpreters, and Aggregators

Paper by Salber, Dey, Abowd (99)



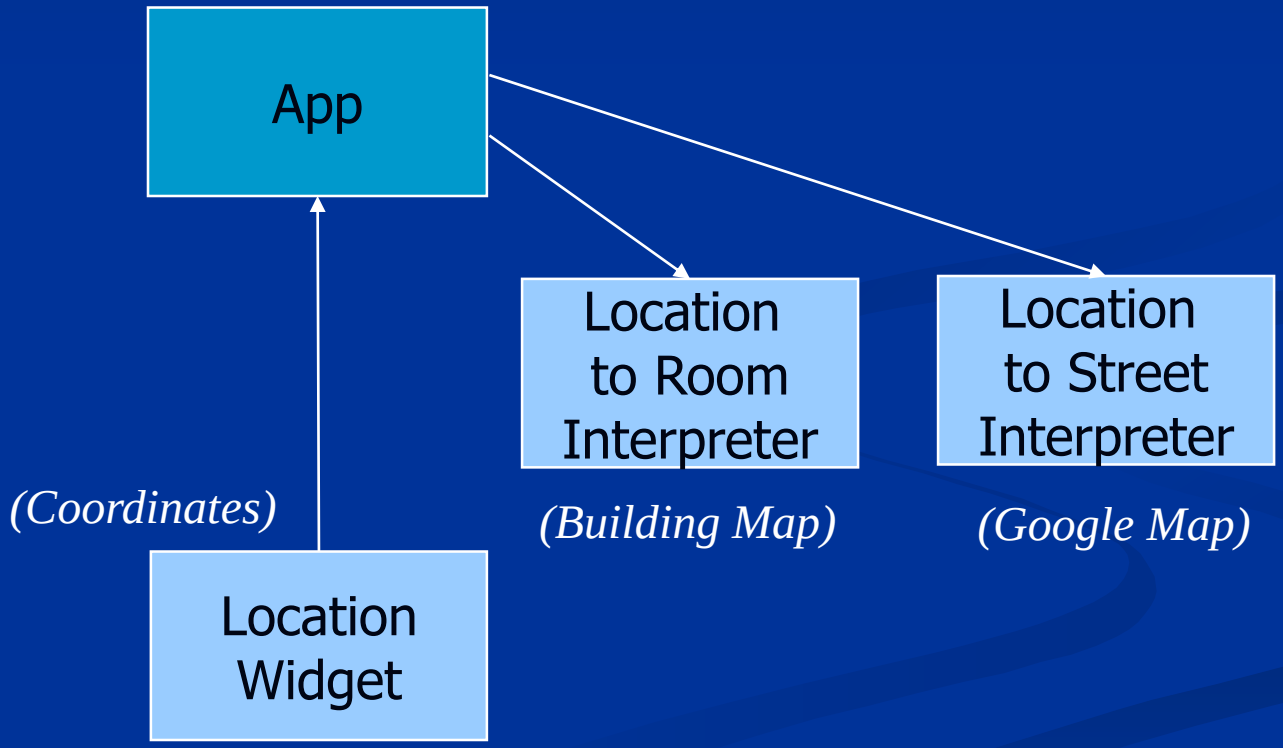
Context Widget

- Widget - a device or control that is very useful for a particular job. *Widget – Wikipedia*, Web *Widget – Wikipedia*, *GUI Widget - Wikipedia*
- A software component that
  - provides applications with access to context information from their operating environment
  - insulates applications from context acquisition
- Responsible for acquiring and abstracting data from particular sensor, separation of concerns, storage



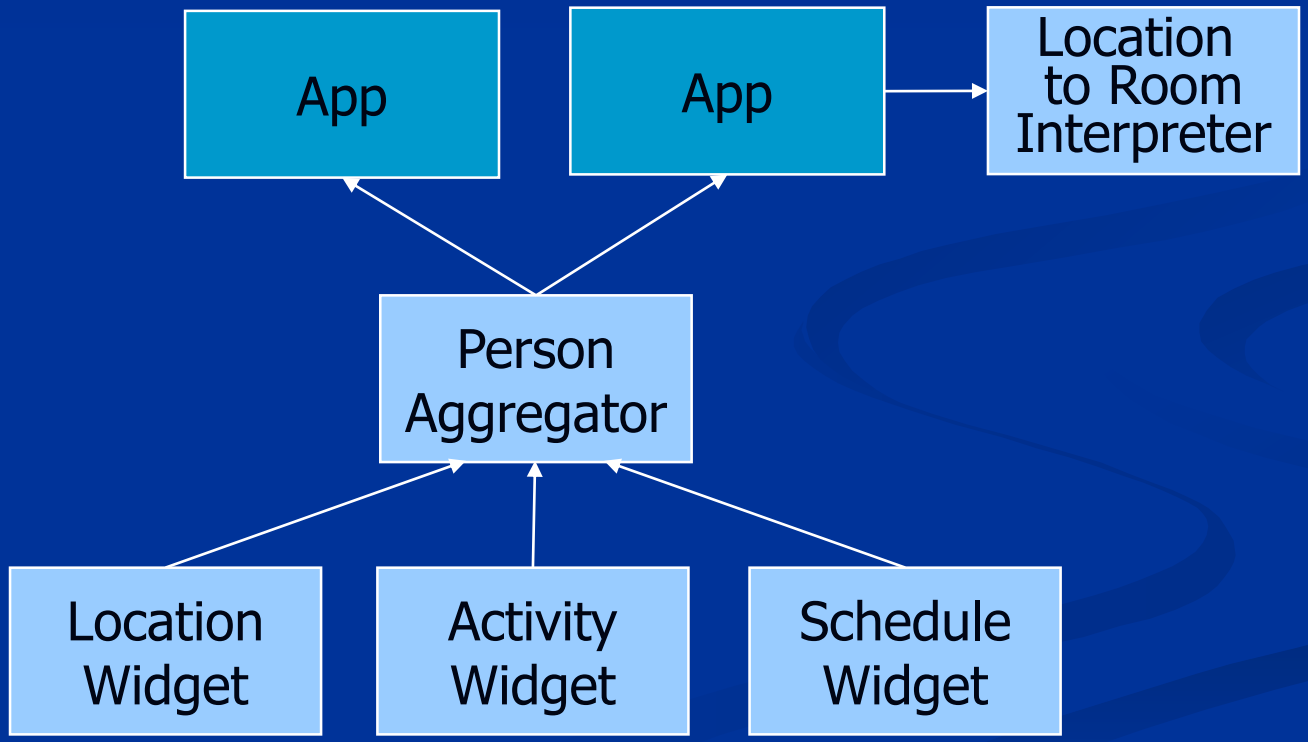
Context Interpreter

- Convert or interpret context to higher level information



Context Aggregator

- Collect contexts relevant to particular entities, e.g., person
- Further separation, simplifies design



Design Process of Typical Context-aware Systems/Applications

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

Design Process: Specification

- Context to use
- Context behaviors to perform
  - Context-aware delivery/presentation of information and services
  - Context-aware automatic execution of services in physical/cyber environments
  - Tagging of contextual information to objects/events for later retrieval and use

Key step in design process: problem specification

Design Process: Acquisition

- Choose and Install relevant sensors
  - Types and numbers of sensors?
  - Sensors: independent, embedded, networked?
  - Where to sense?
  - How often to update and report?
- Context representation
  - Different types, different kind of values, …
  - What's the uniformed/extensible format?
- Store context
  - Files or DBs, what a DB will be used?
  - Each DB for each sensor, one DB for multiple sensors?
  - Timing/Synchronization of data from different sensors?

Design Process: Delivery/Distribution

- Contexts typically captured remotely from applications at different time
- Context captured in sensor-rich environment or device may need to serve multiple applications

  ➔ Need to deliver and distribute context to multiple, remote applications

  - Infrastructure or middleware support
- App/network-level delivery/routing models and transport mechanism

Design Process: Reception

- Application locates relevant sensors/context
  - Sensor/Context service registration
  - Sensor/Context service discovery
- Requests contexts via queries, polls, notifications
  - Query language, event-notification mechanism
  - How often to request?
- Additional interpretation/abstraction/processing
  - Collection, aggregation, filtering, correlation, fusion,...
  - Context semantics/meanings
  - Situation judgment

Design Process: Action/Reaction

- Combine received contexts with previous contexts and system/application states for further analysis
- Perform actions based on the analysis results
- May treat context collection/processing as a separate service

- Check, Evaluation, Improvement, …
  according to the original "Specification"

System Issues (1/2)

- Programming model
  - Programming the physical world
  - Unreliable sensors, recognition algorithms, plus standard distributed computing issues
- Interoperability
  - Sensors, services, and devices
  - Useless if everyone has proprietary / custom systems
  - Need standard data formats, protocols, and frameworks
  - Varying capabilities of sensors, services, and devices
- May need a middleware layer to decouple applications and context sensing
  - Collect raw context, translate to application-understandable format, disseminate it

System Issues (2/2)

- Centralized context server or Distributed architecture
- Power management
  - System kept very low-end
  - Motion detection uses interrupt instead of polling to enable sleep mode in 99% of the time
  - Recharge rule/approach, e.g., power/energy harvest
- Transparency
  - Hiding the technology does not suffice
  - In the battery-charged prototype, users forget to replace the battery because the effect of technology is invisible!
- Experiment and Test
  - Verify and improve the system

Homework

- Find some context-aware case studies and their related materials (paper, presentation, demo, etc.), study their details, and write a report about its/their context used, and related context-aware technologies, systems, applications, etc. Some references:

- CoCo - "… Context-Aware Content Delivery …." by T. Hayashi, et al
- Review of Context-Aware CoCo System – by A. Tsujiguch
- Context-aware Computing, Intel Labs, IDF2010
- Context Is Everything (Video by Lama Nachman, Intel)
- Real World Context-Aware Applications (Video by A. Dey, 2010)
- Location Based Apps (Video)
- Others you like → Important to get materials from Web!!