Outline

- Embedded Software/Systems Research
  - Building the Foundation
- Context-Aware Mobile Computing Research
  - Introduction
  - Survey
- Q&A
Building the Foundation [Lee05]

Embedded Software Issues

- Resource Constraints Not at the Heart
- Time Omitted in Abstractions
- Inherent Concurrency
- Rarely Used OO Techniques
- Avoided Memory Hierarchy & Dynamic Memory Management
- Not Much Attention Paid to Non-Functional Properties
  - Time, security, fault tolerance, power consumption, memory management, etc.

Due to Mismatch of the Provided Abstractions and the Application Domain
Building the Foundation (Cont’d)

- Embedded Software Issues
  - “Hard to Guarantee” Real Time
  - Indirect Timing Specification
    - Avoided concurrency & benchtesting
    - Prioritization
  - Progression of Time Ignored by Abstractions
  - Incomprehensible Concurrency
    - Semaphores, Mutexes, Rendezvous, etc.
  - Unreliability
    - Module Dependency
    - Lack of Compositionality of Core Abstractions

Because Timing Properties Are Not Part of Program Specs
Building the Foundation (Cont’d)

Actor-Oriented Models as Better Concurrency Models

- Key Idea: Flow of Streams of Data, Not Flow of Control
  - Ports as the Interface to a component
  - Concurrency of components

Interpreting & Specializing the Idea

- Ptolemy II (a Lab)
  - Using a Visual Editor
  - Block diagrams as syntax & various semantics defined by “directors” with which diagrams are annotated

E.g., CSP (Communicating Sequential Processes) and DE (Discrete Events) Modeling Sensor Networks
Context Is Key [Coutaz05]

Introduction to Context

- Part of a Process of Interacting with an Ever-Changing Environment Composed of Reconfigurable, Migratory, Distributed, & Multiscale Resources
  - Continuity
  - Ubiquity

Context Issues

- Part of a Process
- Holistic Treatment
- Mismatch between the System’s Interaction Model and Users’ Mental System Model

Structured, Flexible Approach to Challenges of Large-Scale Ubiquitous Computing

Need for New Interaction Techniques
Context Is Key (Cont’d)

Framework for Context-Aware Systems

- Ontological Framework
  - Context as an information space that can be modeled as a directed state graph, where each node denotes a context, and edges denote the conditions for change in context

- Runtime Infrastructure Model
  - Sensing: numeric observables
  - Perception: symbolic observables
  - Situation and context identification
  - Exploitation

- Adaptation & Development

Automatic Acquisition of Situation & Context, and Ultimately the Acquisition of the Entities, Roles, and Relations from Which Situations & Contexts Emerge
References
