

Testing Pervasive Computing Systems in the Large via Multi-Agent Simulation

N. C. Narendra

IBM Software Labs India

narendra@in.ibm.com

Agenda

- Overview & Motivation
- Technical Challenges
- Architectures – System & Simulation
- Proposed System Architecture
- Proposed Simulation Architecture
- Special Feature: Lifecycle-based Simulation
- Special Feature: Simulating Adaptation
- Future Work

Overview & Motivation

- So far, Pervasive Computing (hereafter, *PerComp*) systems tested “in the small”, i.e., in Lab scenarios
- Large scale testing needed before deployment in the “real world”
- Example: A Hospital:
 - 200 beds, 40 nurses, 40 physicians and specialists – most of them constantly on the move
 - About 100 pervasive devices – mobile phones, PDAs, etc., co-existing with desktops
 - Complex workflows – medical tests, operations, long-term medical care for in-patients, coordination among several healthcare professionals

Overview & Motivation (2)

- Difficult and expensive to conduct such large-scale testing in the lab environment
- Also difficult to find a real-life “guinea pig” for testing
- Even if a “guinea pig” is found, need to provide performance guarantees before deployment
- Hence simulation needed
- Can employ well-defined techniques from multi-agent simulation, by modeling the system components as cooperating agents
- Goal: develop “rules of thumb” for PerComp system development

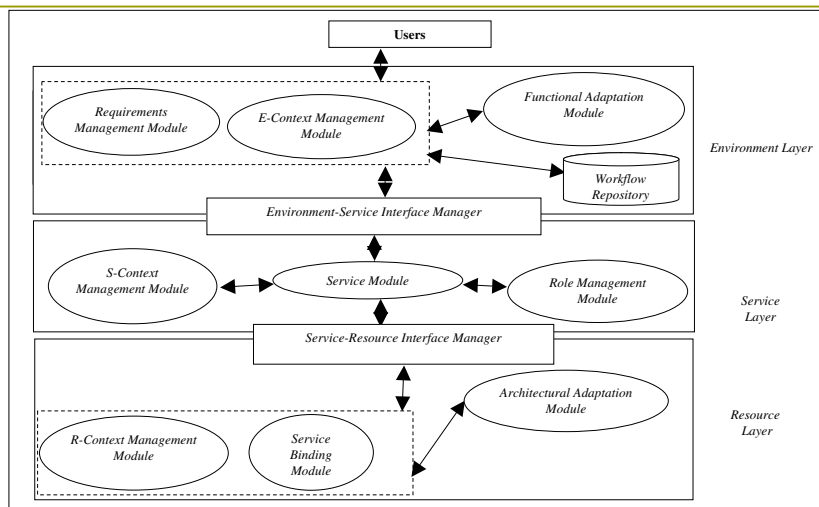
Technical Challenges

- Workflow simulation needed – to model users’ activities
- Physical characteristics of devices, and of underlying communication infrastructure, need to be modeled
- Modeling of *dynamic binding* needed:
 - Binding of users’ activities to the appropriate human and physical resources
- Adaptation also need to be modeled, to account for ad-hoc nature of usage in PerComp systems – two types:
 - Functional – changes in users’ requirements and workflows
 - Architectural – changes in resource/device availability, or in communication infrastructure parameters

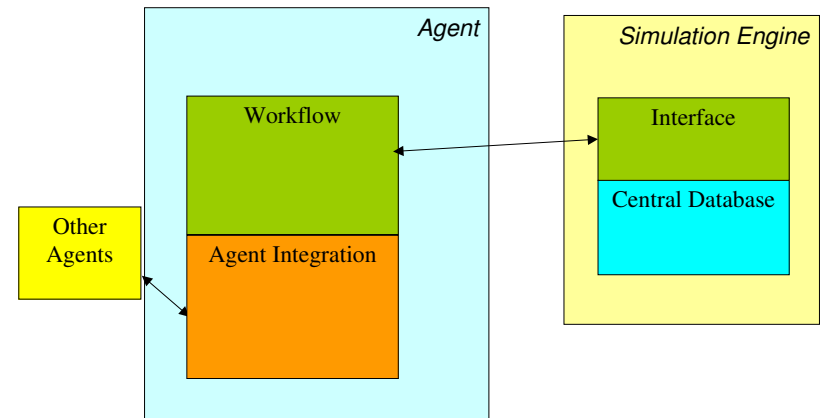
Architectures

- Hence two architectures needed:
 - System Architecture – for modeling the PerComp system to be simulated
 - Simulation Architecture – for the simulation environment itself

Proposed System Architecture



Proposed Simulation Architecture



Special Feature: Script Based Simulation

- Behavior of each component specified as a script, comprising one or more Event-Condition-Action (ECA) rules:
 - When (Event)*
 - If (Condition)*
 - Then Action;*
- Large-scale behavior modeled as combination of several ECA rules

Special Feature: Script Based Simulation (2)

- Attributes to be modeled:
 - Users' workflow activities
 - Message passing among components
 - Point to point
 - Publish-subscribe
 - Multicast
 - Device characteristics
 - Memory capacity
 - Screen size
 - Communication mechanisms available (SMS, email, fax)

Special Feature: Simulating Adaptation

- Determining impact of small-scale changes on the system
- Two types of changes:
 - Functional – changes in users' requirements, leading to changes in workflow sequences
 - Architectural – changes in communication mechanisms, devices used, which could in turn lead to functional changes
- "Butterfly Effect" – do small-scale perturbations produce large changes in the system?

Future Work

- Develop simulation environment
- Test it using real-life scenarios from several domains, such as:
 - Medical
 - Logistics & transportation

Thank You!

Backup Slides

Entities to Model

- Doctors
 - Physicians
 - Specialists
- Nurses
- Technicians
 - X-Ray
 - CAT
 - MRI
 - Others
- Hospitals

Scenarios to Model

- Medical Tests
- Surgical Procedures
- Emergency Medical Treatment
 - Can be modeled as an event triggering adaption

ECA – Events

- Sender (Entity)
- Topic - Header
- Body - contains the following:
 - Control message
 - Data flow message for data transfer

ECA – Conditions

- Operators: =, >, <, <>
- Attributes:
 - Entity attributes
- Values: any real number

ECA – Actions

- Control message
- Data flow message for data transfer