

# Using Declarative Models in Multi-device Smart Space Environments

Sailesh Sathish

# Introduction – Smart Space

- What is smart space? Smart space is any smart or intelligent environment with which users can interact
- “Devices just should work together”
  - Smart spaces can be logical instead of tied to a physical location
  - How do smart spaces differ from distributed systems in general?
    - Physical (proximity of) smart objects
    - Real-world sensor input
    - Interacts with physical environments
    - (Multi-)User interaction
    - Dynamic reconfiguration
    - UI is a combination of multiple agents working in concert
- **Smart space is a multi-user multi-device dynamic interaction environment that is aware of its physical environment working on top of heterogeneous radio technologies/software distribution platforms.**

# Smart Space Use Case

- Tommy is a graduate student at the university in Tampere. He loves music and listens to a lot of music from the web. Tommy generally uses his mobile to download music from the Internet but when at home, he uses his PC for access to web and song downloads. The University of Tampere has a smart lounge that students can come in to relax, read and chat with others. The lounge has several smart devices such as a stereo system, high-end speakers, kiosk PCs for browsing the web, a set of PDA's that users can use and location sensors for users within the room. Access to the smart room is controlled through use of smart cards that contain information about the student using the card. Tommy enters the room using his smart card. Tommy notices that his friend John is the only other person in the room. John is reading his favorite magazine. Tommy decides to listen to some music. He takes his mobile phone, opens the browser application and logs in to his favorite music service. He selects the top 10 songs for the week and selects them to be played. The media player applet loads on his browser. The mobile UI, at the bottom panel, now shows a stereo icon, and two speaker icons. The stereo icon by default is focused on. Tommy presses the joystick button on his mobile to enable the stereo icon after which the icon brings a pop-out showing a play button, a pause button, a rewind button, a forward button, a next button and a dial indicating volume control. The first song now starts playing on the browser. Tommy uses his phone headset to listen to the song. He then moves to the stereo in the room and uses the volume dial to increase the volume. Correspondingly, the media player increases the sound volume from within the browser application. Tommy then presses the next button on the stereo player and the next song within the top 10 song list is selected. The song now starts playing. Tommy then asks John whether he would like to listen to this particular song as this is probably more to John's liking. John replies on the affirmative. Tommy now clicks the speaker icon in the bottom UI layout. The icon now shows a pop-out with a single button showing re-direct audio. Tommy clicks on re-direct audio and removes his headset. The audio now gets re-routed to the particular speaker set in the room that Tommy selected.

# Smart Space Infrastructure

- Requirement for an efficient communication model between participating applications, services and environment
- Dynamic discovery
- Security, privacy etc.
- Predominant approach is to provide a centralized communication hub where providers and consumers of data can communicate
- Blackboard design pattern: approach for non-deterministic environments where a direct solution path to a problem cannot be found
- Data providers put data on the blackboard while consumers at next level processes the data, consumes or put a higher abstraction back on blackboard

# Blackboard Candidate: Interconnected Delivery Context Client Interface (DCCI)

- Delivery Context Client Interfaces (DCCI) as a consumer interface for data consumers
- DCCI is a DOM based API providing a hierarchical representation for static and dynamic context data providers
- DCCI supports organized data representation subject to ontology and supports the DOM event model for asynchronous notifications
- DCCI as of now is meant for local devices with no support for remote properties
- Composite DCCI environment: Combine individual DCCI trees within devices to form composite DCCI tree to form (sort of) blackboard central
- Need for a remote DOM synchronization protocol: Remote Events for XML (REX) as a candidate

# DCCI Requirements

- Lack of a standardized provider interface for context providers – should not mandate specific XPath address from providers, rather the support to be provided by framework
- Lack of a proper security and privacy policy mechanism
- Extensions to DCCI: How to provide service invocations through the DCCI?
- Requirements for remote DOM synchronization protocol: session establishment procedures for composite tree, redundant nodes, validation against composite ontology, dynamic node changes and topology, response times and network reliability
- Standard URI for smart space blackboards?
- Others

# Smart Space Applications: State Machine Approach

- Smart space applications, since they adapt based on environmental conditions, can be seen as state machines where they occupy a certain state in time-space dimensions
- Adaptation seen as state transitions triggered by changes in context where applications jump from one state to another
- Each state characterized by state specific actions that can be rendered by ambient devices or services with actions specific to the entity to which it is assigned
- A smart space provision could encompass a standardized framework for developing state machines capable of performing transitions (adaptation) based on centralized context access mechanism

# State Chart XML

- General purpose event based state machine language
- Supports a standard set of actions that can be executed when a specific state is entered into
- SCXML allows the use of custom actions through the common SCXML paradigm
- Any declarative markup that contains state machines should be able to add event handlers for notification through the context model
- Information about target object that triggered the transition can be used to upload specific processing code either through direct sessions or through information passing model of blackboard



# Summary

- Organized representations of context data can be combined in a multi-device smart space scenario to form a composite model that represents the collective capability of the environment
- Remote synchronization protocols such as REX can be extended to support multi-tree synchronization and composition
- On application development front, a state machine based approach could allow for creation of efficient, adaptive and dynamically extensible applications
- There should be support for universal mode of discovery coupled with support for event handling and remote invocations
- Security and privacy always an issue!
- Response times, synchronization and reliability!
- And, loads of other unseen problems.