Smart Spaces. Ch.5: M3 Applications: Architectures and Designs

Smart Spaces

Chapter 5:

M3 Applications: Architectures and Designs

Outline

§1. Design of M3 Application
§2. Architectural patterns
§3. Examples and case studies
§4. Student projects

§1. Design of M3 Application

- Multi-Agent scenario (execution steps)
  - Shared knowledge (smart space)
  - Cooperation of multiple agents (KPs): indirect interaction
  - Event-based model: subscription
  - Reasoning: deducing new knowledge
- Application Layout (hardware & software)
  - Devices, services (processing), end-users (UI)
- Ontology (problem domain)
  - Knowledge structure model
  - Smart space composite model
- KP design
  - List of all KPs and their mapping to the layout
  - KP implementation template (fill for each KP)

Application layout

Devices, services, end-users

- **Scale:** embedded, mobile, stationary, ...
- **Owner:** personal, multi-user, public, ...
- **Processing:** sensor/producer, consumer, reasoner, a combination
- **Role:** functions in the smart space
- **Interaction:** human, machine
- **Platform:** Linux, Android, Windows, ...

Ontology

- Ontology class graph
  - Classes and properties
- Ontology instance graph
  - Individuals and properties
- Support for reasoning (query-based)
- Context awareness
- Your ontology development toolkit

Smart Space Content: Knowledge Base

- Problem domain and environment
- Knowledge and its classes
- Relations among classes
- Providers, consumers, ...
- Derivative knowledge and reasoning
- Space compositions:
  personal space, multi-user space, application space, ...

KP Design

- Function (in scenario)
- Devices and platforms
- Processing type (architectural role, user)
- External interfaces (e.g., UI)
- Knowledge used (shared & local) and algorithms
- CASE tools

Simple project: steps

1. Idea (brief description)
2. Architecture: cooperation of KPs
   - smart space + KPs + devices
   - scenarios + data flows + presence detection
3. Ontology and knowledge base
   - class graph + instance graph
   - reasoning: query-based
4. KP design
   - Plan of implementation
5. Simple code and demo
   - labs

Characteristic Properties

- Not a database or a web service
- User localization: "space concept"
  - Surrounding devices
  - External services if needed
- Smart services: when, what, to whom, how, …
- Use of shared data as a system
- Semantic linking: "hub property"

Service Intelligence ("Smartness")

- Multitude of scenarios (non-fixed priory)
- Context-awareness
- Adaptability
- Personalization
- Dynamic join/leave of participants

§2. Architectural patterns

- Participating KPs
  - Different roles in application
  - Different knowledge interpretation
  - Different cooperation strategies within the multi-agent system
- Challenges
  - Dynamics: joining and leaving the space
  - Smartness of services:
    - Knowledge reasoning (over the shared content)
    - Service personalization and context-awareness
    - Proactive service delivery

Knowledge producers and consumers

- Accumulation and provision
- Similarly to a shared database with readers and writers
- No "smartness" (intelligence)

- **Do not use in your projects in its pure form**

Knowledge interaction chains

- Pipes: linear chains
  - E.g., a simple weather application
- Tree-based
  - Each fact produces several new facts
  - One-to-many synchronization
  - Epidemic dissemination
- Network flows
  - Cycles are possible
  - Iterative processing
  - Feedback

Knowledge mediators

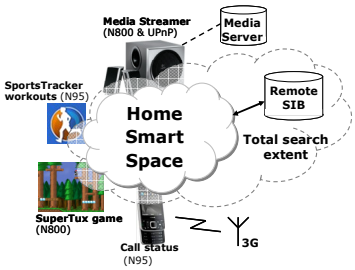
- Smart space analyzers
  - Big Brother approach (for proactivity)
  - Services and their composition (see also knowledge interaction chains)
- Function delegation
  - Mobile client has low capacity
  - Ideal case: client shares small piece of personal info and consume the service

§3. Examples

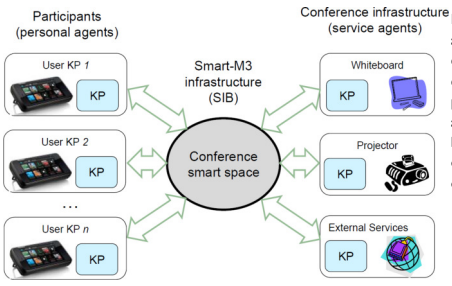
1. SuperTux game example (NRC, Helsinki)
2. Smart Conference System (SPIIRAS)
3. SmartScribo System for multi-blogging (PetrSU)
4. Smart Room (PetrSU)
5. Social Networks service (FRUCT)
6. SmartDiet: Personal Wellbeing Assistant and Diet Planner Mobile Service (TUT, Tampere)
7. Open International M3 Semantic Interoperability Workshop, http://www.fruct.org/eit-m3

Traditional user services

- Personalization and functionality expansion of the popular user services
- SuperTux game example
- Simple service composition

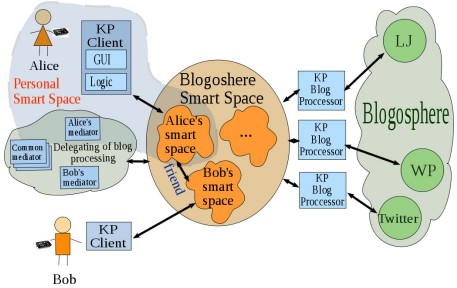


Smart Conference System



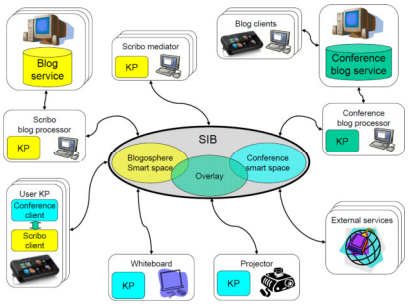
Intelligent assistance in complicated conference processes, automating the burdensome work of conference organizers

SmartScribo System



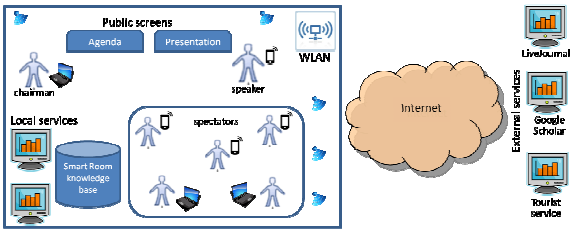
Bloggers access multiple blogs through the smart space, where relevant blog discussions are dynamically collected

Integration example



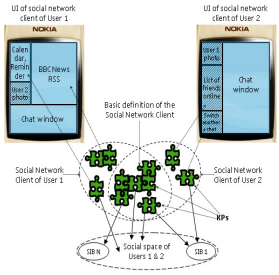
- One application integrates a service from another application
- Unidirectional flow

Smart Room



Social Networks service

- Personal life-assistant services, which continuously collecting, organizing and reasoning over the user data
  - client applications for different devices
  - user databases on top of SIBs
  - interfaces to other social networks
  - Ad-hoc telcos and other meeting services
  - Shared whiteboard and pool of ideas
  - Automated community management
- Community members can focus on deliverables and not on processes



§4. Student Projects

- Students show their projects
- Focus is on the scenario, architecture, and design
- Making "smartness" (personalization, proactivity, etc.)