# Smart Spaces

## Chapter 5:

### Smart-M3 Applications: Architectures and Designs

---

### Outline

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

---

### §1. Design of a Smart-M3 Application

- Multi-Agent scenario (execution steps)
- Shared knowledge (smart space)
- Cooperation of multiple agents (KPs)
- Presence detection
- Reasoning: deducing new knowledge
- Application Layout (hardware & software)
- 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, Symbian, 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**

---

**13th FRUCT Conference**

- [http://fruct.org/conference13](http://fruct.org/conference13)
- Registration
- Mandatory parts of the course:
  - 25.04 (Thursday), 10.00-12.00 Section *Internet of Things and Smart Spaces I*, IT-Park Conference Hall, room 103/IT-park
  - 25.04 (Thursday), 15.30-17.30 Section *Internet of Things and Smart Spaces II*, room 152
Smart Spaces Content

- 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, ...

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

Simple project: details

- See AppDesign_ru.pdf

§2. Architectural patterns

- Participating KPs
  - Different roles in application
  - Different knowledge interpretation
  - Different cooperation protocols within the multi-agent system
- Challenges
  - Dynamics: joining and leaving the space
  - Presence detection: personalization and proactivity
  - Smartness: knowledge reasoning

Knowledge producers and consumers

- Accumulation and provision
- Similarly to a shared database with readers and writers
- No “smartness”
- Do not use in your projects in its pure form

Knowledge processing 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 processing 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. SOFIA project results

Traditional user services
- Personalization and functionality expansion of the popular user services
- SuperTux game example

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
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
- Use of Presence Detection for making “smartness” (personalization, proactivity)