Piecemeal Development of Intelligent Smart Space Applications

EU-RECOCAPE
3rd Dissemination Event, Feb. 26th, 2014
Jarkko Kuusijärvi (jarkko.kuusijarvi@vtt.fi)
VTT Technical Research Centre of Finland
Motivation

- Smart spaces exploit IoT for communication and networking.

- Smart spaces are inherently dynamic heterogeneous and trade-offs between different qualities (e.g., performance and security) at run-time is not a trivial task.

- Smart-space application development is difficult because of the number of different entities interacting with each other.

- Smart-space software development needs a workflow to guide different development teams to faster time-to-market with new applications.
Introduction

- PSE (Piecemeal Service Engineering) approach has been developed through 4 years of smart-space application development.

- Needed because the use of different ontologies, rules and adaptation methods in application development was not easy. Especially, the use of dynamic rules may be difficult for application developers.

- Based on our over ten-year experience on
  - Service architectures, semantics, dynamic systems, semantic databases, ontologies, etc.
Piecemeal Service Engineering (PSE) Workflow

Specification
- Smart-space Application Innovation
- Smart-space Application Co-creation

Development
- Behavior-Driven Application Development
- Incremental Testing of Smart-space Applications

Smart-space Application Development Tools
- Architectural Knowledge Base
- Knowledge Creation and Management Tools
Piecemeal Service Engineering (PSE)

- PSE is an incremental development approach for intelligent applications
  - It supports abstraction, aggregation, and adaptability in smart-space development.
  - Eases the development of complexity by providing the developers tools and knowledge that can be reused.
  - Helps in maximising the reuse of existing knowledge of business and design practices and existing technical assets in the development of new smart-space applications.
  - Utilizes reusable architectural knowledge (principles, ontologies, patterns, and rules).
Case example of RECOCAPE (SALE)

- Working with SECC to bring security features into the service
  - Server side implementation is a combination of smart-space applications (KPs) and a Semantic Information Broker (SIB).
  - Smart space integration with Web Services and web pages to bring the semantic service to multiple platforms.

- Utilizing the PSE approach and working as the ‘Security Team’ in the development process
  - Adding security features on top of existing workflows / use cases and working together with other development teams to integrate functionality.
  - Enforce information security and user privacy.
Case example of RECOCAPE (SALE)

- Develop Security Service Adaptation Layer
  - Confidentiality, integrity, and availability
  - User authentication / authorization
  - Access control

- Reuse existing solutions from smart space applications and SIBs
  - Implement security features on top of open source Smart-M3 SIB implementation to allow secure usage of the advertising platform.
  - PSE applied since security features are implemented as Knowledge Processors running in the smart-space.
  - Use existing ontologies, KPs.
Applying PSE in SALE
(Security team’s viewpoint)
- **Idea identification**
  - What type of application? What is it for and whom will it be delivered? Why?
  - Discuss with group of business, domain, and technical professionals to collect ideas

- **Idea scoping and analyses**
  - Business analysis: potential markets, etc.
  - Domain analysis: information about trends

- Produces initial description of a smart-space application and related analytical reports, expected impacts, available assets, and identified risks.
- **Idea identification**
  - Develop a secure advertisement platform to be used via different devices.
  - Mobile phones will use the service via REST interface
  - Application will be used by consumers with their mobile phones to receive advertisements and offers intelligently according to the context information

- **Produced information/analysis**
  - Description of the application
  - List of existing assets (security use cases, implementations, etc.)
  - Architectural knowledge (security ontologies)
  - New: Integrating smart-space applications with REST services.
Describing a usage scenario of an application

- Defines how the development work is divided into parts, what dependencies there are between them and what are the priorities.
- People from different technology areas are involved in the work.
- The result is a revised version of the SSA scenario description

Breaking a usage scenario into pieces

- Break the scenario description into a set of use cases to illustrate specific aspects.
- Different teams looks the SSA from different points of view
  - Application domain – end-user stakeholders
  - Development and maintenance – life cycle management
  - Security team – information security
- Describing a usage scenario of an application
  - Scenario definitions for actors
    - Mobile
      - End-user
    - Web
      - Admin
      - Advertiser (Product owner)
      - Publisher (Ad publisher)
      - Media creator
      - Admin

- Breaking a usage scenario into pieces
  - Security use cases (Security team’s viewpoint)
    - SIB security implementation
      - Different user accounts, roles
      - Confidentiality, integrity, availability
      - Security ontology
      - Firewall, etc.
    - REST authentication/access control
      - Access to SIB secured
      - For RESTful mobile access to SIB
    - Web security implementation
      - Access to SIB secured
Describe Behavior

- Use the scenario description to
  - Define the agents and services required for the behavior and the interactions between the actors
- Annotate the behavior description (sequences) to available ontologies (e.g., domain, context, and security ontologies)
- Select rules related to the selected ontology and map them
  - If a specific rule not available, describe it in textual format

Transform to Programming Language

- Implement the describer behavior either by some programming language, or
- Implement the behavior of the described rules in SPARQL for dynamic applications.
Describe Behavior

- Use the scenario description to derive security requirements for different actors in a set of sequence diagrams.
- Use cases for different actors with different security requirements.

Transform to Programming Language

- Security ontology based access control / authorization for interaction between the SIB and KPs
  - Encrypting passwords in SIB
  - Access Control KP
  - Authorization KP
- End-user application with required security (certificates, https)
- Web application with required security features (certificates, https, session invalidation)
• Testing of intelligent applications
  • GUI Application – End-user
  • Legacy adapter – Glue software, static or dynamic rules
  • Dynamic agent – Agent that is fully controlled by the dynamic rules assigned for it at run-time

• Tests with RDF datasets and with the SIB
  • Perform unit, regression, and integration tests with RDF datasets or SIB
    • The RDF dataset constitutes a single moment in time in the environment
  • Perform collaboration, system, and field tests with the SIB (actual environment)
• Testing of intelligent applications
  • GUI Application – End-user
    • Web application and Android application
    • Legacy adapter – Glue software, static or dynamic rules
      • KPs to implement security features into Smart-M3

• Tests with RDF datasets and with the SIB
  • Test plan and test cases for the security use cases
  • Perform system (and field) tests
PSE – Lessons learned

- Benefits of the PSE approach grow after smart-space development iterations (because the knowledge base grows)
  - Using PSE approach speeds up the development
  - Reuses software artifacts as much as possible
- Innovation is independent on location, distance and missing expertise.
- Integration of different solutions (i.e., networks, servers, systems, and applications) is straightforward and very fast.
- Development process is much faster with scenarios available for reuse.
- PSE is usable with basically any type of semantic technology
  - Has been used with, e.g., smart spaces implemented by SIBs and Virtuoso.
Conclusions

- Development of intelligent smart-space applications can be solved with new technologies such as semantic web and autonomic computing.

- PSE applies model and ontology orientation for describing knowledge at design-time and at run-time for creating intelligent applications.

- PSE applied and validated in an incremental way in multiple laboratory and industrial case studies.
References

  - More detailed information on the PSE approach
  - More information about dynamicity and rules, which were not covered in the example here.
  - A list of examples and related publications.