

# Mobile Multi-Service Smart Room Client: Initial Study for Multi-Platform Development

Andrey S. Vdovenko, Sergey A. Marchenkov, Dmitry G. Korzun

Petrozavodsk State University  
Department of Computer Science



This project is supported by grant KA179 of Karelia ENPI - joint program of the European Union, Russian Federation and the Republic of Finland



13<sup>th</sup> FRUCT conference  
April 25, 2013, Petrozavodsk, Russia

# Table of Contents

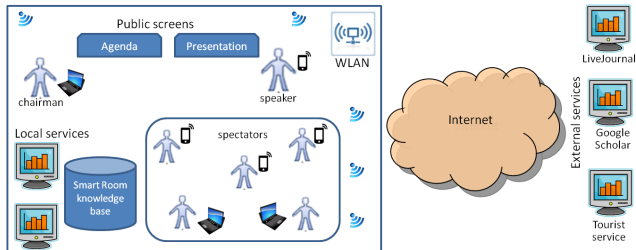
**1** Smart Room

**2** Development of SR clients

**3** Conclusion



# Smart Room system

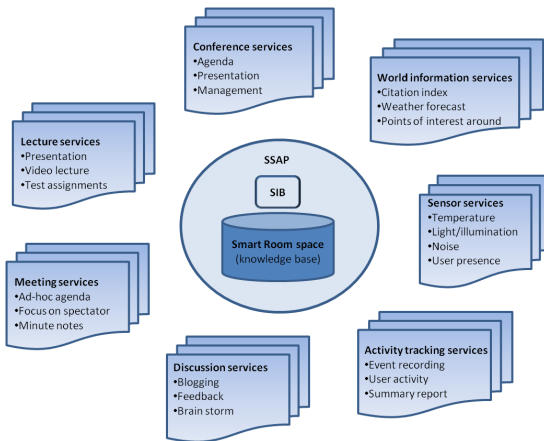


- Smart Room aims at many services:  
informational, control, collaborative work, ...
- Participation of many users
  - ↪ Many (mobile) clients running and accessing SR services
- Users come with own devices
  - ↪ Many mobile platforms



# Multi-Service Property

- Services allow user interact with SmartRoom and get different sort of information
- Access to services go via a **client** from a mobile device of user



# Client for Smart Room

- Basic services
  - ▶ everyone uses them, e.g., slide show
- Personalized service access (UI design problem)
  - ▶ appropriate services from a large set
- SR services appear and disappear dynamically
- Development unification (as much as possible, trade-offs)
  - ▶ Windows Phone and Symbian (Qt) are our reference cases
  - ▶ Desktop solutions (Windows, Linux)
  - ▶ Android and iOS are in progress



# Multi-platform considerations

Platform	Programming language	IDE
Android	Java, partly C, C++	Eclipse
iOS	Objective-C	Xcode
Symbian platform	C++, QML	Qt Creator
Windows Phone 7&8	C#, Visual Basic	Visual Studio 2010
Windows desktop	C#, Visual Basic	Visual Studio

- A lot of programming languages and IDEs
- GUI is platform-aware
  - ▶ Universal GUI frameworks are not mature enough



# Object oriented design

- **KnowledgeProcessor** - SmartSpace interaction
  - **Classes of main services** - e.g. Agenda (list of participants), Projector (presentation, changing slides) and etc
  - **Internal logic** - logic of work with got data
  - **Graphical user interface** - representation of information on user's device
- 
- **Green** - same part
  - **Red** - different part



# Smart Spaces SDK

The primary SDK is SmartSlog

<http://oss.fruct.org/wiki/SmartSlog/>

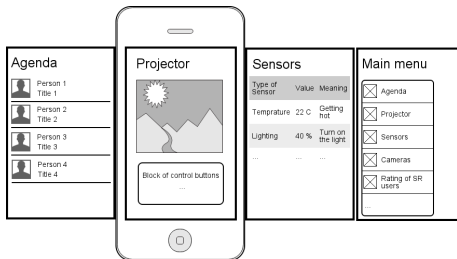
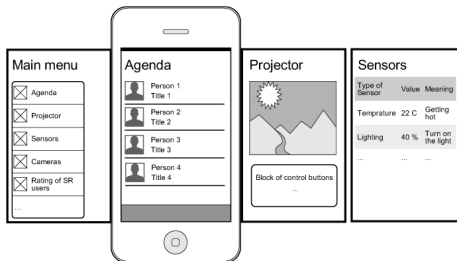
- ANSI C version for mobile Linux family, Qt/Symbian, Android, iOS
- C# version for Windows desktop family, Windows Phone 7&8
- High-level (model-driven, ontology-based) programming
- Modest to device capacity

Use of native code is required for some platforms





# User interface



- Each service as tab
- Menu with all services
- Static and dynamic services

- Static - implemented for each platform
- Dynamic - download from available services

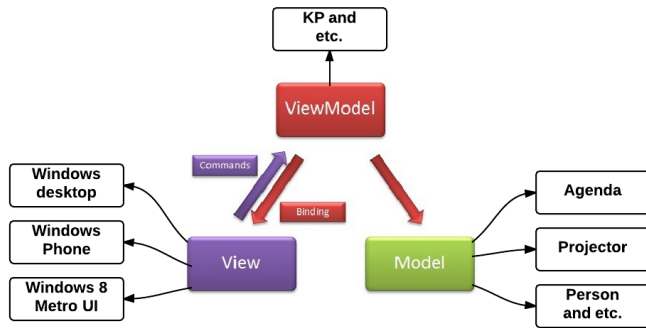


# HTML5 solutions

- HTML5 application with specialized mobile frameworks (PhoneGap, jQuery Mobile, ...):
  - ▶ "+" one code for any platform
  - ▶ "+" application can be used in any browser
  - ▶ "-" bad performance on old mobile devices
  - ▶ "-" some difficulties in implementation
  
- **HTML5+CSS+JavaScript pages**
  - ▶ "+" many platforms support HTML5
  - ▶ "+" one page for all platforms
  - ▶ "+" cross-platform
  - ▶ "-" some difficulties in implementation



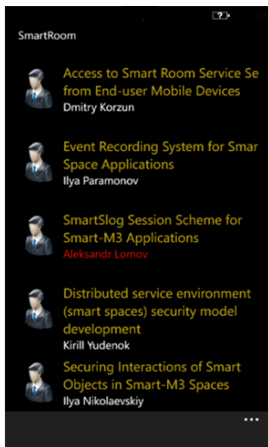
# MVVM pattern



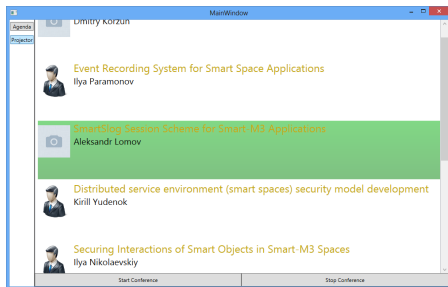
- **Model** is data required for the application;
- **View** is graphical user interface;
- **ViewModel** contains data for View and commands for work with Model.



# Graphical user interface examples



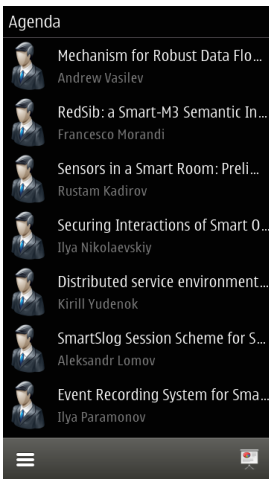
Agenda on WP



Agenda on Windows



# Graphical user interface examples



Agenda on Symbian



Projector on Symbian



# Android and iOS

## ■ Android

- ▶ Java
- ▶ ANSI C SmartSlog
- ▶ IDE Eclipse

## ■ iOS

- ▶ Objective-C
- ▶ ANSI C SmartSlog
- ▶ IDE XCode

## ■ Every platform use MVC pattern for default

Come to our demo to see what we have



# Results

<b>Service</b>	<b>Platform</b>
Agenda	Windows family, Symbian, Android, iOS
Projector	Windows family, Symbian
Autorization	Windows family, Symbian, Android
Blogging	In future
Sensors	In future
Google Scholar	In future

- We have solution for dynamic services;
- We almost have basic services for each platform;

