# Access to Smart Room Service Set from End-user Mobile Devices

Andrey Vdovenko*, Dmitry Korzun*†
*Petrozavodsk State University (PetrSU), Russia
†Helsinki Institute for Information Technology (HIIT), Finland
{vdovenko, dkorzun}@cs.karelia.ru

**Abstract**

The Smart Room project develops smart rooms at Petrozavodsk State University. A smart room provides a service set to automate such research and educational activity as conferences, lectures, and meetings. For participants the services are accessible via end-user mobile devices. A mobile device runs a smart room client, which registers the participant in the smart room space, shares required personal data, and accesses available services. In this extended abstract we consider our initial GUI design and development plan for smart room clients. The key requirements we focus on are (1) unification for various mobile platforms, e.g., Windows Phone, Android, and iOS; (2) personalized access to the service subset most appropriate for the user at given time and context.

**Index Terms:** Smart Room, Mobile devices, GUI, Cross-platform.

Petrozavodsk State University (PetrSU) started the project "Smart Room". It develops rooms equipped with various computational devices and corresponding software aiming at intelligent assistance for university research and educational processes. PetrSU smart room generalizes Smart Conference system [1], [2], which was developed in 2010-2012 by FRUCT partners: SPIIRAS, YarSU, LETI, and PetrSU. The scenarios include automated construction of the agenda and dynamic adaptation to context, participant interests and limitations. It supports collective generation of new knowledge using available services. The development is based on smart spaces [3], Smart-M3 platform [4], and SmartSlog SDK [5].

In the basic scenario, a user participates in the activity by coming into the room and running a smart room client (SR-client) on her/his mobile device. The client becomes aware of available services and allows the user to select and access desired services. Services may be ongoing conference blog (communication between the participants and the presenter), browsing slides (current presentation or others), sensor data (temperature, light, etc.), remote control of cameras located in the room, the participant research ratings (e.g., the citation index from Google Scholar), and many others. Therefore, SR-client is a key user tool for participation in the smart room.

First, we analyzed which mobile platforms are suitable for implementing SR-client and selected the following ones: Windows Phone, Android, iOS, Symbian, MeeGo, Maemo. The problem we solve in this direction is possibility to create a unified user interface, with very little effort and common to all platforms. Note that some of the above platforms already support (or Qt support is announced) Qt as a unified tool for GUI construction. Table I summarizes which languages and development environment we selected for each platform.

Second, we analyzed available technologies that can allow constructing a multi-service client. The GUI architecture and standard interface elements must be common for the all

This research is a part of grant KA179 "Complex development of regional cooperation in the field of open ICT innovations" of Karelia ENPI programme, which is co-funded by the European Union, the Russian Federation and the Republic of Finland. The article was published with financial support from the Strategic Development Program of Petrozavodsk State University.

TABLE I
PLATFORM DEVELOPMENT ENVIRONMENTS

| Platform | Programming language | Integrated development environment |
|---|---|---|
| Android | Java, partly C, C++ | Eclipse |
| BlackBerry | Java | Eclipse |
| iOS | Objective-C | Xcode |
| Symbian platform | C++, QML | Qt Creator |
| Maemo | C++, QML | Qt Creator |
| MeeGo | C++, QML | Qt Creator |
| Windows Phone | C#, Visual Basic | Visual Studio 2010 |

platforms. The architecture is designed with the purpose to let the user to comfortably take advantage of many opportunities and information. There are many services available in the room, and the user may select a personalized subset of them.

Service navigation and control can be based on menu or use an advanced screen with tabs and special controls. Both methods require plug-ins, which become available for each active service at an appropriate point. The user starts a specific page on which (s)he clicks for each active plugin. After that the user has ability to quickly navigate the service features.

The menu-based method is inappropriate when a large number of plug-ins appears. Placing them into a single menu makes the GUI cumbersome, especially for such mobile devices as phones with small displays. In this case, the second method is better; Fig. 1 shows an example how an advanced screen can look on the SR-client.

On tablets, the layout of available plugins can use the same idea as for mobile phones. Since a tablet has a larger display, the user can simultaneously track many plug-ins on the same window and use multiple tabs for the content. The layout can be implemented with window managers (e.g., Xmonad for Linux and AquaSnap for Windows 7). Another option is specialization for each platform. For instance, in Windows Phone we can use so called docks, which are similar to the working space layout mechanism of Visual Studio.

As a development tool, it is possible to employ the Qt framework, which allows creating crossplatform client version for some mobile platforms, namely Symbian, Harmattan, Win-

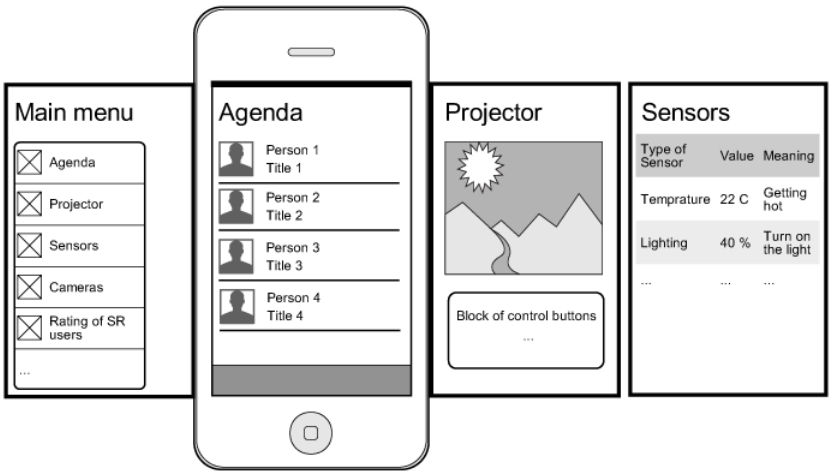


Fig. 1. Advanced screen layout of SR-client

dows (tablet). The rest platforms include Windows Phone, Windows 8, Android, and iOS. For them we shall use specialized tools.

Since a SR-client is faced with a big service set available in the smart room, (personalized) selection is needed for the user. It is a challenging problem to flexibly and dynamically build appropriate multi-service interface elements in dependence on the context and user interests.

We considered several solutions. The first one is a single application written using HTML5 libraries (e.g., PhoneGap, Titanium, Rhodes). This solution is cross-platform since the mobile platforms provide HTML5 support. Unfortunately, mobile devices of the previous generation likely meet performance problems and limited functionality.

The second solution is to represent each service as a separate HTML5 plus CSS3 page using JavaScript for internal logic of the device page. The SR-client collects information from services and constructs the service pages. The key services are represented with static pages. Some services can be constructed dynamically, based on a set of small services, e.g., representing data parameters from sensors.

Currently, we started client implementation with static pages. We expect that the static approach has no essential performance problems related to the capacity of previous generation of mobile devices. The dynamic page construction is at the design phase.

## REFERENCES

[1] D. Korzun, I. Galov, A. Kashevnik, K. Krinkin, and Y. Korolev, "Integration of Smart-M3 applications: Blogging in smart conference," in *Proc. 4th Conf. Smart Spaces (ruSMART'11) and 11th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN'11)*. Springer-Verlag, Aug. 2011, pp. 51–62.

[2] I. Paramonov, A. Vasilev, N. Kozhemyakin, I. Timofeev, A. Subbotkin, E. Krylov, D. Korzun, and I. Galov, "Multiplatform client and pecha kucha support for smart conference system," in *Proc. 11th Conf. Open Innovations Association FRUCT*, Apr. 2012, pp. 197–198.

[3] D. G. Korzun, S. I. Balandin, V. Luukkala, P. Liuha, and A. V. Gurtov, "Overview of Smart-M3 principles for application development," in *Proc. Congress on Information Systems and Technologies (IS&IT'11), Conf. Artificial Intelligence and Systems (AIS'11)*, vol. 4. Moscow: Physmathlit, Sep. 2011, pp. 64–71.

[4] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö, "Smart-M3 information sharing platform," in *Proc. IEEE Symp. Computers and Communications*, ser. ISCC '10. IEEE Computer Society, Jun. 2010, pp. 1041–1046.

[5] D. G. Korzun, A. A. Lomov, P. I. Vanag, J. Honkola, and S. I. Balandin, "Multilingual ontology library generator for Smart-M3 information sharing platform," *International Journal on Advances in Intelligent Systems*, vol. 4, no. 3&4, pp. 68–81, 2011.