Delivery of SmartRoom Services Using Mobile Clients

Andrey Vdovenko*, Dmitry Korzun*†
*Petrozavodsk State University (PetrSU), Russia
†Helsinki Institute for Information Technology and Department of Computer Science and Engineering,
Aalto University, Helsinki, Finland
{vdovenko, dkorzun}@cs.karelia.ru

***Abstract*—The SmartRoom system provides a set of digital services for collaborative activity in a room. The information is shared in a smart space—SmartRoom space (Smart-M3 based implementation). Human participants become users accessing the services from clients running on personal mobile devices (e.g., smartphone). We consider the problem of service delivery to SmartRoom users. On the client side the delivery is either implemented natively or as js web application. Services and their users are represented in the SmartRoom space. They are dynamically related to achieve the personalization property. The latter can be supported with known methods of semantic matching applied between the user interests and service description.**

Keywords*—Smart Spaces, Smart-M3, Collaborative environment, Mobile devices, User interface, js web.

The SmartRoom system constructs a digital environment in a spatial area (a room), where human participants perform collaborative activity (e.g., a conference or meeting). A set of digital services is provided [1]. They are implemented in a distributed manner based on sensing, processing, network, and user interfacing capabilities of surrounding devices and remote computers. A participant accesses the services from her/his mobile device (smartphone, netbook, etc.).

The Smart-M3 platform [2] is used as follows. Semantic information broker maintains a smart space—SmartRoom space. Participating devices host autonomous software agents, which are responsible for services and their users. The agents cooperate via information sharing in the space, where the operations go through the broker.

A special class of agents is SmartRoom clients. A participant launches the client on her/his devices, thus becoming a user of SmartRoom services. The user can access multiple services during the participation. In our previous work [3] we found that the personalization of service delivery is an essential property for SmartRoom clients. In this abstract we report our further development of service delivery and its personalization.

Since in general a client is faced with a large service set available in the SmartRoom environment, personalized (and automated) selection is required. A challenging problem is to build the multi-service user interface (UI) that dynamically relates SmartRoom services with the context and user interests.

On the client side, services are classified onto *off-the-shelf* and *ad-hoc*. An off-the-shelf service is oriented to non-trivial client application logic and UI, which are essentially customized for the use case. It needs a particular implementation on the respective mobile OS. The most solutions are rather static (design-time), with low flexibility in runtime. Example services [3] are Agenda, Presentation, and Discussion.

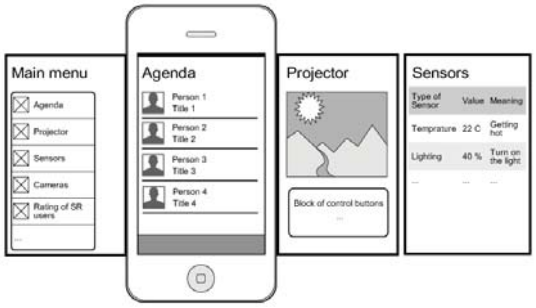

Fig. 1. Screen layout of SmartRoom client: each service is represented with a tab and user navigates on the available tabs

An ad-hoc service is oriented for runtime construction. Its implementation should include generic schemes for composition of service from available components. The runtime flexibility becomes higher, providing more means for personalization. An example is the service page that shows measurements from the SmartRoom sensors including those sensors only that the user is interested to track.

The scheme of combining these two classes of services in mobile client UI is shown in Fig. 1; it is inherited from our previous work [3]. In this abstract let us focus on ad-hoc services.

To support the runtime flexibility, ad-hoc services have lightweight UI, which primarily aims at input and output from/to the user. The application-level logic processing is preferably delegated to the SmartRoom infrastructure [1]. We propose to implement an ad-hoc service as a web-application. Therefore, such a service is identified by its own URL and can be delivered using web-browsing methods.

Runtime construction is supported by sharing information in the SmartRoom space. Services are ontologically represented and stored in the space. The SmartRoom infrastructure (special agents) is responsible for the maintenance and control.

Every service is attached to the SmartRoom ontology [1] through the ontological class *Service*. Its properties describe the service attributes and link the service with other entities in the smart space. Figure 2 shows the ontology class for a SmartRoom service.

This approach allows any client to query the URL and then access the service for delivery to the user. Potentially, some services can be constructed directly in the SmartRoom space, when clients initiate the process requesting for interested service. The SmartRoom infrastructure dynamically constructs pages and publish the links and other attributes into the space.

Personalization can be achieved by relating every service with user interests, similarly as it was proposed in SmartScribo system for multi-blogging [4]. The problem can be solved using known matching methods. A simple example is keyword text matching: a service is related to the user if they have common keywords.

We further classify ad-hoc services onto "elementary", "one-page", and "complex". The classification reflects the way of representing the service on the client side. Elementary service is responsible for a little piece of information (e.g., data from a sensor, citation index from Google Scholar). One-page service fits one web-page, though it may contain a lot of structured output information and require structured input from users (e.g. EventRecorder [5]). Complex service requires several web-pages and possibly essential data processing at the client side (e.g., image search function from World Around Me [6]). The web-based approach allows combining several services, when several unit services can be composed into one-page service.

Service composition is important for personalization, when the service representation depends on the user context (collected in the smart space). The composition depends on "semantic matching" of services with the user. For instance, appropriate services have many keywords appeared in the user interests. That is, personalization can be implemented by matching keywords in users interests (user profile) with keywords in service description. Matching may use synonyms. Services that are recommended to the user are bind to him/her. She/he can also include other services by explicit requests.

The first step is for client: getting *Interest* property of *Person* class from the smart spaces. After that the result is split onto separate words for use in matching. Then the list of available objects of *Service* class is queried from the smart space. For every service we get *Description* property and make matching of words from *Interest* of this *Person*. If we find a match then we mark this service as recommended. After matching we set available platform-aware services and send a request for URLs of ad-hoc services. The SmartRoom infrastructure then performs a required composition of the services.

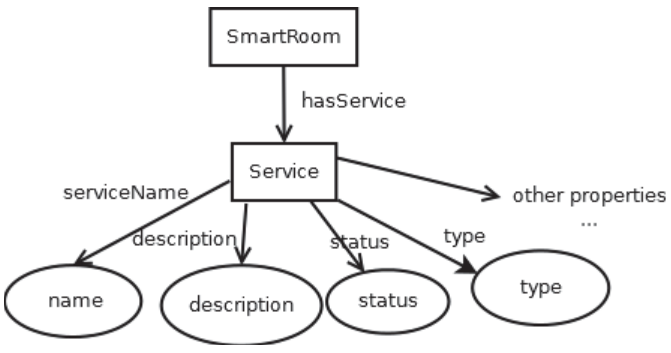

Fig. 2. Ontological representation of a service

Every ad-hoc service has the same structure of source code. It consists of the folders js (javascript sources), css (cascading style sheets sources), html (web-pages), and file index.html in the root directory. Note that HTML5 and jQuery Mobile are widely used on many mobile platforms. They provide means for cross-platform development.

The mobility property leads to the connectivity problem when a client accesses and uses services. In WLAN settings, network connections are subject for instability. Note that the problem appears in a personalized way: at the same time a subset of clients can suffer only. It leads to additional client-side mechanisms for reconnection to the space and restarting when the client has crashed. Many platforms provide a mechanism to watch on network connection, e.g., Windows Phone have special namespace for classes and events for detecting when and why connection was lost or other.

User registration is an initial step for a client before using the SmartRoom system. Every user of SmartRoom has installed client on her/his mobile device. User can register their account through Registration-service. This way needs only input of login and password (by the user in appropriate fields). After that a personal spaces is created in the smart space and the user can join and leave the SmartRoom system during the participation.

Anonymous users (no registration is needed) are also allowed. In this case, client does not require any specific registration. The service delivery in anonymous mode is reduced and its personalization is low or even impossible.

REFERENCES

[1] I. Galov and D. Korzun, "Smart room service set at Petrozavodsk State University: Initial state," in *Proc. 12th Conf. of Open Innovations Association FRUCT and Seminar on e-Tourism*, S. Balandin and A. Ovchinnikov, Eds. SUAI, Nov. 2012, pp. 239–240.

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

[3] A. Vdovenko, S. Marchenkov, and D. Korzun, "Mobile multi-service smart room client: Initial study for multi-platform development," in *Proc. 13th Conf. of Open Innovations Association FRUCT and 2nd Seminar on e-Tourism for Karelia and Oulu Region*, S. Balandin and U. Trifonova, Eds. SUAI, Apr. 2013, pp. 143–152.

[4] D. G. Korzun, I. V. Galov, and S. I. Balandin, "Proactive personalized mobile mutliblogging service on SmartM3," *Journal of Computing and Information Technology*, vol. 20, no. 3, pp. 175–182, 2012.

[5] I. Galov, R. Kadirov, A. Vasilev, and D. Korzun, "Event recording in smart room," in *Proc. 13th Conf. of Open Innovations Association FRUCT and 2nd Seminar on e-Tourism for Karelia and Oulu Region*, S. Balandin and U. Trifonova, Eds. SUAI, Apr. 2013, pp. 20–28.

[6] A. Vdovenko, A. Lukovnikova, S. Marchenkov, N. Sidorcheva, S. Polyakov, and D. G. Korzun., "World around me client for windows phone devices," in *Proc. 11th Conf. Open Innovations Framework Program FRUCT*, S. Balandin and A. Ovchinnikov, Eds. SUAI, Apr. 2012, pp. 205–206.