Enhancing the SmartRoom System with e-Tourism Services

Andrey S. Vdovenko, Sergey A. Marchenkov, Dmitry G. Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{vdovenko, marchenk, dkorzun}@cs.karelia.ru

***Abstract*—The SmartRoom system is a service-oriented application for assisting such collaboration activity as conference or meeting in a room equipped with computing and presentational devices and Internet access. The development is open source and based on the Smart-M3 platform. In this paper, we consider advanced scenarios for SmartRoom to enhance the latter with e-Tourism services. We introduce a smart space based architecture for this enhancement. We provide an ontology for representing and sharing the tourism-related information for service construction. Based on the architecture and ontology, several case study services are designed. In particular, we implement a service for collaborative construction of a social program for conference participants. The implemented service is integrated into the SmartRoom system and demonstrates the feasibility of the proposed design.**

I. Introduction

Tourism is large industry nowadays. Its growth shows a significant year to year increase, and support from e-services is clearly demanded on the market (e-Tourism). More and more people become aware of e-Tourism advantages for planning their activities [1], [2], [3]. Existing computing environments for collaborative work can be enhanced with the e-Tourism services to take own niche on the market. For instance, such services can augment the primary collaboration activity by automation and assistance in construction of a social program and travel plan for the activity participants.

Development of e-Tourism services can utilize the emerging approach of smart spaces. In general, it creates computing environments for heterogeneous devices to share their resources [4], [5]. The use of smart spaces in e-Tourism was recently studied in [2], [6], [7]. In this paper, we consider the case of SmartRoom system [8], which demonstrates the use of smart spaces for one demanded application scenario. SmartRoom provides a set of services for assisting such collaboration activity as conferences or meetings. Personal mobile devices are primary access and control points for users [9]. The core services aim at support for intensive in-room collaborative work using surrounding devices for hosting the system.

The SmartRoom system creates a domain-specific knowledge sharing environment (virtual shared workspaces) for collaboration activity of human participants. The system consists of software agents that construct and deliver services in a shared smart space—*SmartRoom space*. It makes localization and relation of information in regard to the in-room area and to information sources of participants. The implementation is based on Smart-M3 [10]; the latter provides means for creating and deploying a smart space in a given computing environment. A semantic information broker (SIB) maintains its smart space utilizing technologies from the Semantic Web. Agents act as knowledge processors (KPs) that share information and form cooperatively the smart space and its services.

Based on our previous work [8], we consider SmartRoom as a case study of the e-Tourism enhanced systems for collaborative work. This paper makes the next development step toward the SmartRoom enhancement. The development includes design solutions and their implementation as well as experimental deployments with the implemented software prototype. We propose e-Tourism scenarios for advanced use in SmartRoom collaborative activity. We develop a smart space based architecture for this enhancement. We provide an ontology for representing and sharing the tourism-related information for service construction. Based on the architecture and ontology, several services are designed for the case study. In particular, we implement a service for collaborative construction of social program for conference participants. Integration of this service into the SmartRoom system successfully demonstrates the feasibility of the proposed solutions.

The rest of the paper is organized as follows. Section II introduces possible scenarios for enhancing SmartRoom with e-Tourism services. Section III describes our architecture for integrating the services. Section IV presents our ontology to extend the existing SmartRoom ontology for the service integration. Section V provides our design solutions for the selected services. Section VI considers implementation details for the case study—social program service. Section VII concludes the paper.

ISSN 2305-7254

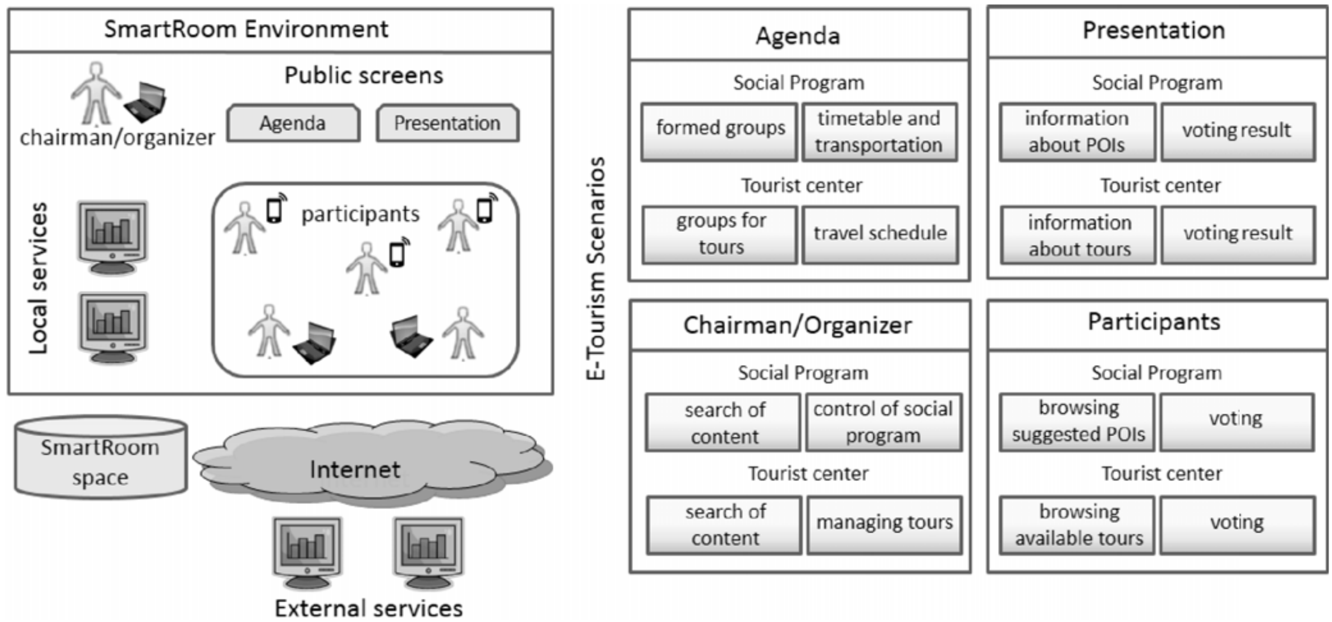


Fig. 1. Architectural scheme of enhancing the SmartRoom system with e-Tourism scenarios

II. E-TOURISM SCENARIOS FOR SMARTROOM

The core SmartRoom services are Presentation-service and Agenda-service [8] (see Fig. 1, left): they are responsible for information visualization on Presentation and Agenda screen, respectively. Participants show their presentations using Presentation screen and have control on the multimedia visualization. In parallel, the current activity program is visualized on Agenda screen. The public screens are large-format displays visible to all participants in the room. Other core SmartRoom services information processors. For instance, Conference-service constructs and dynamically maintains the activity program. Each activity participant becomes a SmartRoom user via her/his SmartRoom client—an agent running on the personal mobile device [9] (e.g., smartphone or tablet).

The basic SmartRoom design supports enhancing the system with new services [11], [8]. Enhancement of the core services introduces additional assistance for the collaborative work. The conceptual model is presented in Fig. 1 for the studied e-Tourism case. We define and analyze the following e-Tourism scenarios for this enhancement.

Social program for conference participants. In this scenario, SmartRoom system assists such collaborative activity as conference. Some SmartRoom participants are interested in information about a possible social program. The latter includes points of interests (POIs), which conference participant can visit during the social event. The organizers provide predefined POIs (e.g., a preliminary tour plan). Then a participant can make decisions on her/his plans related to the social activity on conference: which POIs are of personal interest as well as preferred time of the visit.

This decision-making process is iterative: a participant updates her/his decision depending on observable plans of others. Based on the collected decisions, the organizers then finalize the social program construction. The result includes: (1) which groups to which POIs have been formed, (2) timetable, and (3) transfer support. The scenario is primarily oriented for use inside the room where the conference goes on. The construction process and its results are dynamically shown on public screens and on personal mobile devices.

SmartRoom in travel agency. This scenario assumes deployment of SmartRoom in a travel agency or tourist center. Participants are tourists that visit that agency. The services assist in selection of interested tours. In this case, the SmartRoom system supports such collaborative work as tourism activity. Agency offers many tours available for sale. Each tour contains information about POIs, related events, schedule for visits, weather forecast, transport, accommodation, etc. Most of this information is extracted in automated manner from the external web services. A participant makes decisions on her/his plans related to selection of tours.

Based on the collected participant's decisions and interests the tourist agency constructs a plan for tours. The result includes (1) formed groups for tours and (2) travel timetable. Presentation service and Agenda service become responsible for visualization of information on tours, timetable, and groups. Importantly that users can receive this information also using their personal mobile devices.

The above scenarios assume semantic relation of heterogeneous multi-source information. For instance, in addition to photos and textual descriptions, historical information can be associated with POIs [7]. The scenarios can be advanced by utilization of runtime information on user presence in the room, including physical and virtual (remote) presence. This information can associated with network activity of personal mobile devices. Basic SmartRoom scenarios with user presence detection were defined in our previous work [12]. Now let us consider how they can be used in the e-Tourism-oriented SmartRoom services.

User arrival. Before starting the collaboration activity, the users arrive and gather in the room (first-time join) and preparing/waiting the forthcoming activity. Detection of user arrivals activates personalized welcome services and provides runtime initialization for the collaboration activity. For example, everyone can see who is ready to participate. In additional, newcomers can be gently offered to install the SmartRoom client.

User joins and leaves. Such dynamic evens are frequent in SmartRoom. In the case of conference activity, real-time status of every user provides information for maintaining the activity agenda. For example, the system moves or cancels a planned presentation if the speaker is absent, or excludes the participant from social program.

Activity statistics. During the collaboration activity, personalized information about network activity is accumulated. At the end of the activity a summary report is generated, which describes the contribution for each user. Similar statistics can be computed and provided to users during the activity.

III. HIGH-LEVEL ARCHITECTURE

The tourism-aware services for enhancing SmartRoom are summarized in Table I. They include Social Program service, Search service, and Web Page service. The architecture for implementing the above e-Tourism scenarios based on these services is shown in Fig. 2. In this case, a service is implemented as a software agent (KP) running on a computer (inside or outside the room). All services interact by sharing information in the SmartRoom space.

TABLE I. SERVICES FOR ENHANCING SMARTROOM

Service	Description
Social Program	Control and construction of a social program based on available POIs and tours to visit and the decisions of participants.
Search	Information search for POIs and tours in external data sources (e.g., photos in Flickr).
Web Page	Generation of web pages based on service templates and sharing the links in the SmartRoom space.

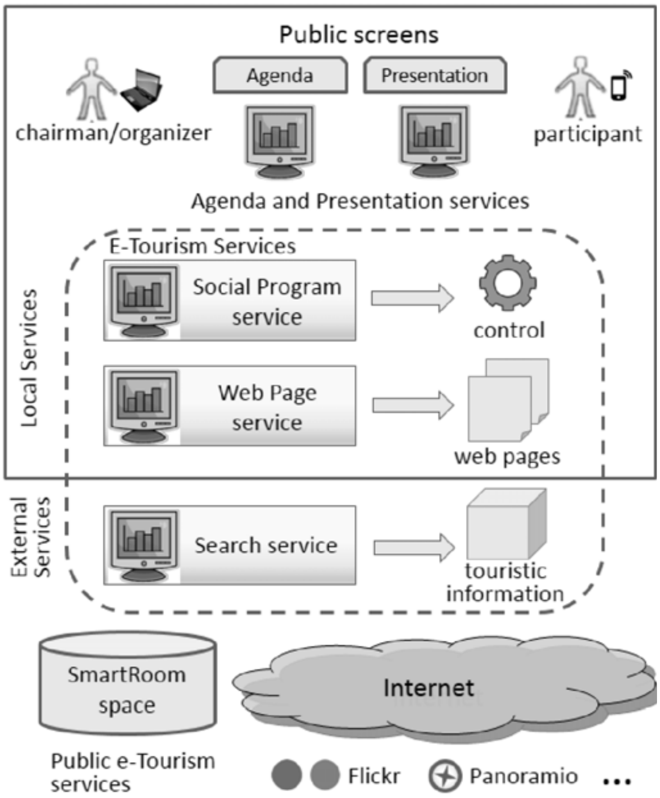


Fig. 2. Architecture for integrating the e-Tourism services to SmartRoom

Social program service provides each SmartRoom participant with information about possible POIs in the social program. The corresponding agent runs on organizer's computer and provides the graphical interface for browsing POIs information, watching decisions of the participants, and constructed social program. The service interacts with Search service and Web Page service.

Search service implements discovery and extraction of photos as well as associated descriptions of the related POIs. The search is limited with a certain area around the given location (e.g., the place of conference). Other appropriate touristic information can be found and extracted as well. The result is shared in the SmartRoom space and then is used by Social program service. Preferably, the corresponding agent runs on a powerful server machine.

Web Page service constructs and keeps pages that contain information to deliver to users. This use of web technologies simplifies the service delivery and information visualization on the client side. Note that for such web pages only their links (URLs) are shared in the smart space. In recent implementation, the service supports Presentation service, Agenda service, and SmartRoom client. The service runs on a host accessible as a web server in the Internet.

In the considered e-Tourism scenarios, Presentation service and Agenda service become also responsible for visualization of the produced touristic information. The latter is represented in the form of web pages. Presentation service displays the page with available POIs and the voting process about them. Agenda service displays the social program, which is constructed collaboratively by the participants.

IV. ONTOLOGICAL REPRESENTATION

Let us present an ontology for use of the considered e-Tourism scenarios in SmartRoom. The latter already has the specific ontology [8], which defines how the data related to different services and users are represented in the SmartRoom space. The ontology consists of two parts: service ontology and user profile ontology. We extend this SmartRoom ontology for Social Program service as shown in Fig. 3. The key properties are hasClientUrl, hasAgendaServiceUrl, and hasPresentationServiceUrl. They are used for accessing to user interface if the latter is implemented as a web application (see our architecture in Fig. 2 above). Values of the properties are published by Web Page service.

When organizers add a new instance of the suggested social program, the corresponding individual of class SocialProgram is created and published in the SmartRoom space. The individual establishes a relation with the service individual. The following data properties are also set: socialProgramTitle, socialProgramLatitudeCoordinate, and socialProgramLongitudeCoordinate. The last two properties are needed for automated search of POIs.

Ontology for representing POIs in the SmartRoom space is shown in Fig. 4. The primary individuals are Placeslot, Place, and Photo.

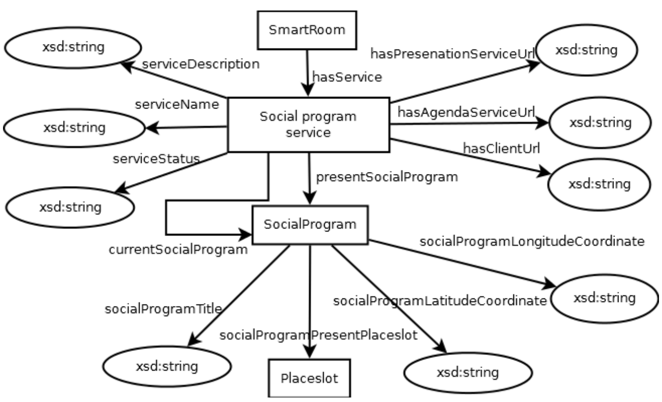


Fig. 3. Ontology for Social Program service

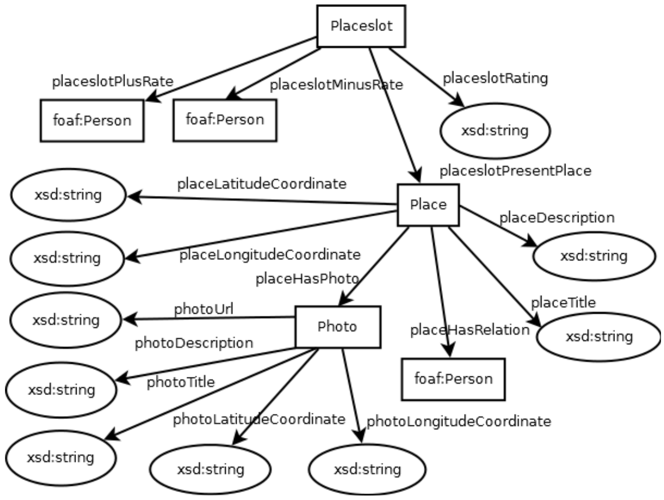


Fig. 4. Ontology for POIs

Individual Placeslot has object property placeslotPresentPlace to link with Place. The other object properties placeslotPlusRate and placeslotMinusRate are to account voting results of the participants. When new voting starts those properties are cleaned. Individual SocialProgram (Fig. 2) has no direct relation to Place (Fig. 4). The semantic relation uses an intermediate individual Placeslot and object property SocialProgramPresentPlaceslot.

Individuals Place and Photo are common for many application domains [13]. The developed ontology provides support for exploring new knowledge based on relations between Person and Place. New POIs and related touristic information can be searched automatically, based on recent individuals Place and Photo in the SmartRoom space.

Our ontology provides semantic relations between participants and POIs, similarly as it happens in [14], [15]. This type of relations is established in the SmartRoom space, not affecting external data sources. Such a relation is presented in individual Place with property placeHasRelation. For instance, historical information can be associated with a POI. Also, a relation of a given POI and participant's region can be established, e.g., the architect of the building was from certain country.

V. SERVICE DESIGN

Consider design details of the services listed in Table I (see Section III). Social Program service is oriented to end-users; the program is visualized on public screens or on personal mobile devices. Search and Web Page services support this primary service in content provision and delivery.

A. Social Program service

Social Program service is constructed by a software agent. It provides control to organizers and implements the construction of a social program based on content in the SmartRoom space. The control function includes constructing a social program, adding new POIs to the program, specifying the initial information, initiating a search of media inforamtion.

The primary individuals for ontology-based construction of a social program are Social Program and Placeslot. Object property socialProgramPresentPlaceslot relates those individuals. Individuals Place and Photo keep information on POI and its pictures, respectively. This information is provided by the other services. The service cooperation uses notifications generating by Search and Web Page services.

In social program construction, a participant provides her/his individual preferences and makes voting, similarly as it is described in [16]. When a participant makes her/his vote for POI, then Web Page service sends a notification to Social Program service. The latter, in turn, processes the notification and updates the program. Vote recalculation is stored in individuals Placeslot due to properties placeslotPlusRating and placeslotMinusRating for responsible individuals Person. After that, Social Program service forms a new JSON object and sends it to Web Page service. This object is used further as content for web pages constructed and visualized for participants.

Note that if SmartRoom is used in the scenario for travel agency (tourist center) then there can be many tours (programs) with various POIs. The service maintains a JSON object with all suggested programs.

B. Search service

There are a lot of public data sources in the Internet with tourist-aware information: description of POIs and events, available tours, weather data and transport, accommodation, etc. In particular, let us mention huge collections in such image hosting web services as Flickr and Panoramio.

Search service allows discovering this information in open web services. Descriptive information is extracted taking into account the location of POI. Each found photo or picture is represented in individual Photo with relation to appropriate *Place* by object property placeHasPhoto.

C. Web Page service

Web Page service provides a way to effectively deliver service result (content) to the users. On the user side, a typical web browser function is needed only. The page construction is dynamical and accepts parameters in the construction request. The source content is taken from the SmartRoom space.

Every service, which needs construction of web pages, defines own templates for web pages. Basically, a template uses HTML and JavaScript. In advance, a template can use CSS styles for more attractive view. URLs of constructed pages are published in the SmartRoom space and can be accessed through properties hasClientUrl, hasAgendaServiceUrl, hasPresentationServiceUrl of individual Service.

There are global JSON objects, each keeps information about the service. Such objects are used for content visualization on the user side (JavaScript is used). They also can be passed as parameters for generation of subsequent pages.

D. Interaction

Notification model. Interaction between the considered services follows the notification model [17]. Whenever a service publishes data some other services are notified about the changes due to subscription. The notification model describes a finite set of possible variants for services to interact with each other. Each service subscribes to its own notification types, each corresponds to an interaction variant and represented as a set of RDF triples kept in the smart space.

Our instance of the notification model is shown in Table II. We use two types of notifications: simple and compound. The latter uses an additional entity as a parameter. All notifications act us one-to-one requests, excluding notification updateServiceInformation for one-to-many action.

Activity individual. We use a particular case of ontology-based subscription when it represents a persistent query to

TABLE II. NOTIFICATION MODEL FOR ETOURISM SCENARIOS

Name		Description
Service	Web Page	Called for updating web application content of Service
Notification	startUpdateServicePage	
Parameter	Service	
Service	Social Program	Activate recalculation of votes for Place
Notification	voteUpdate	
Parameter	Place	
Service	Presentation	Show Service on Presentation screen
Notification	startServiceMode	
Parameter	Service	
Service	Presentation	Stop showing Service on Presentation screen, switching to conference activity program
Notification	stopServiceMode	
Parameter	Service	
Service	Presentation	Update Service on Presentation screen
Notification	updateServicePage	
Parameter	Service	
Service	Agenda	Show Service on Agenda screen
Notification	startServiceMode	
Parameter	Service	
Service	Agenda	Stop showing Service on Agenda screen, switching to conference activity program
Notification	stopServiceMode	
Parameter	Service	
Service	Agenda	Update Service on Agenda screen
Notification	updateServicePage	
Parameter	Service	

detect changes in specified properties of a given individual. When subscription is initialized, all updates, which are related to the subscribed individual, are regularly sent to all the subscribers. It results in increased load of the subscribers. For reducing this load we introduce "activity individual", which virtually represents a particular activity in the SmartRoom space.

When a service publishes an activity individual in the smart space, this service also subscribes to the property that represents "completion of processing". When a service finishes handling responded activities, this service sets status of the activity to related property. In turn, sender receives only one subscription event related to its individual and after checking status of activity the service removes this individual. The performance is improved since services subscribe only to the information they need. Similar ideas for effective interaction can be found in [18].

An example is shown in Fig. 5. The interaction between Social Program service and Search service with an activity individual is specified in Algorithm 1. Social Program service and Search service interact using subscription to individual SearchActivity. The latter contains the search parameters (e.g., location of POI, type of search). Search service extracts the parameters and accesses appropriate web services (e.g., for photo search it uses Flickr and Panoramio).

REST query. In interaction with Web Page service, we assume this service has low information about other services. In this case, the use of subscription becomes more complicated, since the service must has knowledge about all individuals. Web Page service acts as a mediator between the smart space and the client side (web application). The required interaction is based on the mechanism of parameterized REST query [19], [20]. For an example, when a participant votes, the client web application sends POST request with special URL.

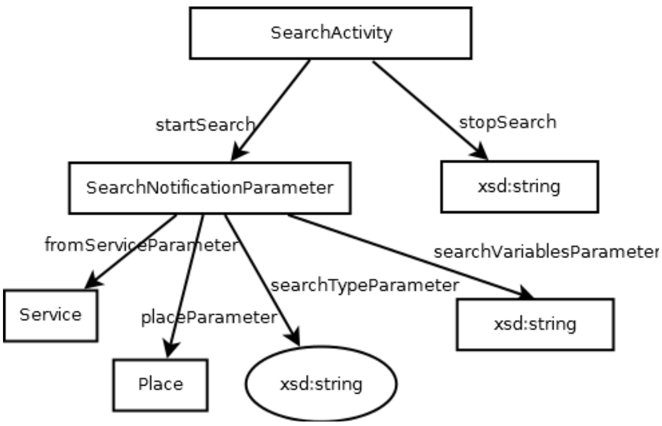


Fig. 5. Search Activity individual

Algorithm 1 Interaction between Social Program service s_{scl} and Search service s_{sch} using individual SearchActivity i shared in smart space S

s_{sch} subscribes to class of i;
s_{scl} publishes i in S;
s_{scl} subscribes to property stopSearch of i;
s_{sch} receives i by subscription;
s_{sch} extracts search type and parameters;
s_{sch} performs the search and publishes the result in S;
s_{sch} sets the search status in property stopSearch;
s_{scl} receives the search status from stopSearch property;
s_{scl} removes i from S;

The response has the following parameters: Placeslot UUID, Person UUID, and vote value. (UUID stands for Universally Unique Identifier.)

A particular case is how Social Program service interacts with Web Page service. Social Program service makes uses subscription based on notification VoteUpdate. When the notification is received by Social Program service, the latter creates a new JSON object is created, see Fig. 6. This object follows the ontology for social program as a root element. Individuals and their properties are stored in this object.

```
{"SocialProgram": {
  "Uuid": ...,
  "Title":"Test Program for Petrozavodsk",
  "Latitude":"61.784",
  "Longitude":"34.349",
  "Placeslots":[
     {"UUID": ...,
          "MinusRating":["Person":{...},...],
          "PlusRating": ["Person":{...},...],
          "Place":
                {"Uuid": ...,
                 "Description":"test",
                 "Latitude":"61.756",
                 "Longitude":"34.437",
                 "Photos":[
                        {"Uuid": ...,
                         "Description":"test",
                         "Latitude":"61.763",
                         "Longitude":"34.419",
                         "Title":...,
                         "Url": ...,
                         "UrlMedium": ...,
                         "UrlThumb": ...,
                        }, ...]
                }}, ...]
}}
```

Fig. 6. Example of JSON object for Social Program service

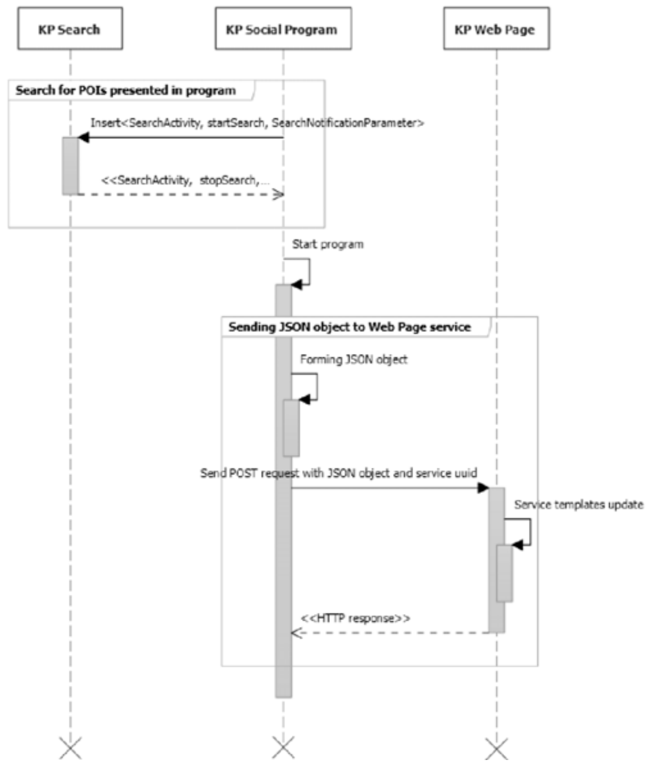


Fig. 7. Sequence for searching and start program processes

After formation of the object, it is sent to Web Page service with unique URL in a POST request. The service replaces the current object instance with a new JSON object. The new instance is then used for background updates by AJAX calls and for creating the page by the template engine. Notably that this proposed interaction mechanism supports enhancing SmartRoom with other services, which can interact with Web Page service.

Processing steps. Figure 7 shows the process of searching and start of program with use of the introduced mechanisms. First, the organizers, after configuring available POIs in the program, start search of photos and other media information. Second, the organizers start the selected program. It entails processing program in the JSON object and sending it to Web Page service. The latter updates template source parameters and hosts refreshed pages.

Figure 8 presents the process of accessing service pages by clients and the vote process. Any client can get URL from property hasClientUrl of individual SocialProgramService, when the client starts or due to subscription. Participants make their votes for some POIs, and corresponding properties appear in the smart space. Web Page service publishes a

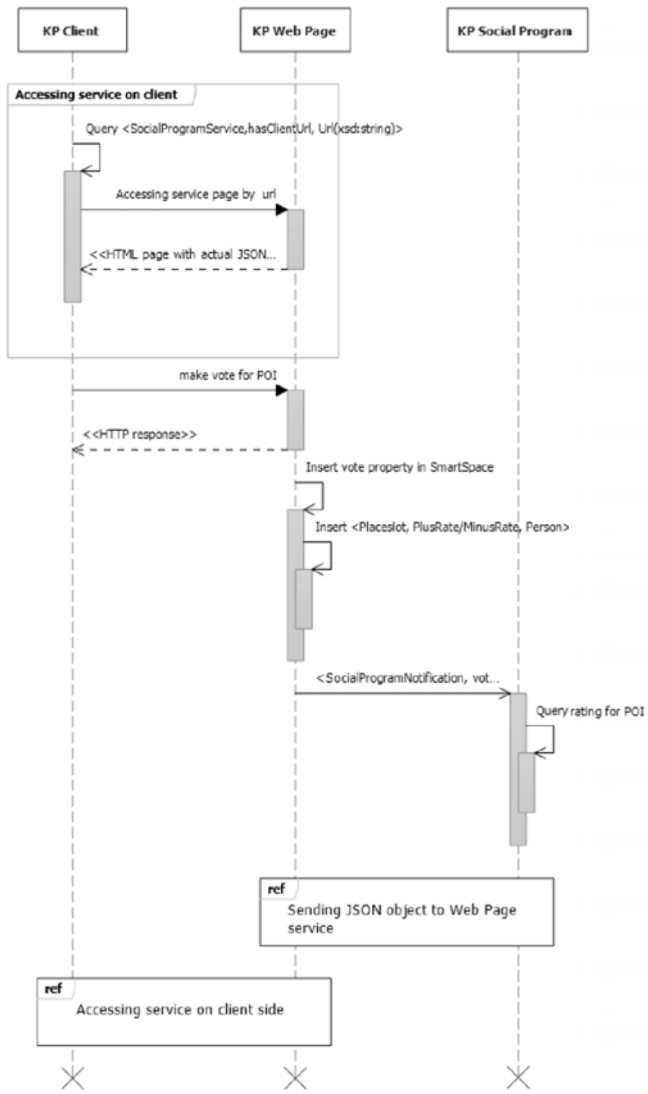


Fig. 8. Sequence for accessing by client and voting processes

notification for Social Program service. It recalculates votes and forms a new JSON object. The page for client is updated by AJAX calls.

Figure 9 presents the process of accessing service pages for Agenda and Presentation screens. When organizers need to show the social program, they switch modes on Agenda and Presentation screens. For clients, the information access is very similar. The assigned properties are hasAgendaServiceUrl for Agenda service and hasPresentationServiceUrl for Presentation service.

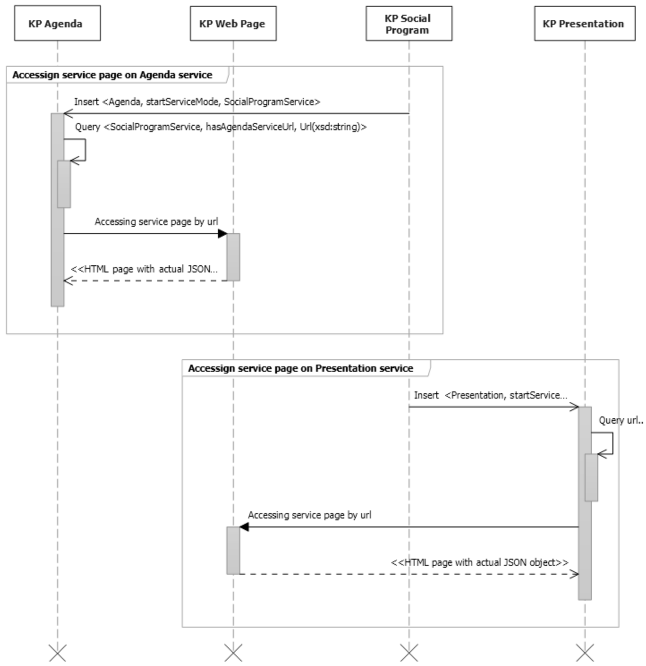


Fig. 9. Sequence for accessing by Agenda and Presentation services

VI. Case Study Implementation

Open source implementation of the considered services can be found in the SmartRoom project at https://sourceforge.net/projects/smartroom/. Table III presents the list of technologies our implementation utilizes in each service.

TABLE III. UTILIZED TECHNOLOGIES

Technology	Services	Description
Visual C# and .NET Framework	Social Program Search	Programming languages
Python	Web Page	
Windows Presentation Foundation (WPF)	Social Program	Graphical user interface for organizers
SmartSlog SDK (sourceforge.net/projects/smartslog/)	Social Program Search	Interaction with the SmartRoom space
Python KPI	Web Page	
World Around Me search engine (oss.fruct.org/projects/wam/)	Search	Search media information, including photos, POIs descriptions, etc.
CherryPy (www.cherrypy.org)	Web Page	Python web engine for REST query processing and accessing web pages with service content
Mako (www.makotemplates.org)	Web Page	Python template engine for constructing web pages
HTML, JavaScript, CSS jQuery (jquery.com) jQuery Mobile (jquerymobile.com)	Web Page	Web page programming tools

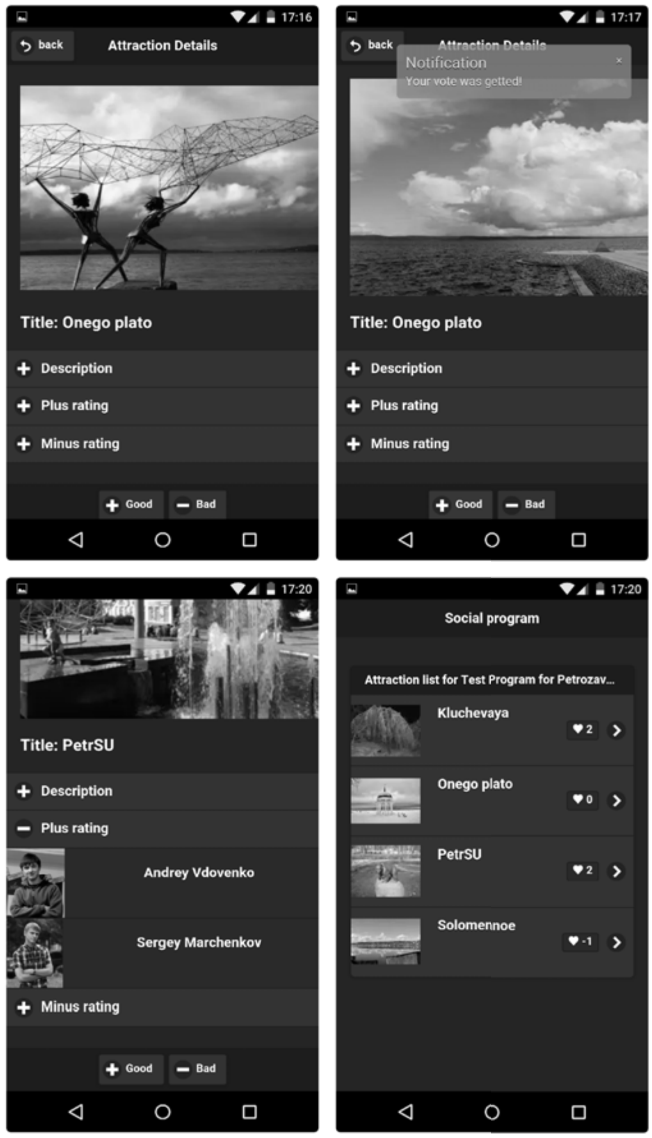
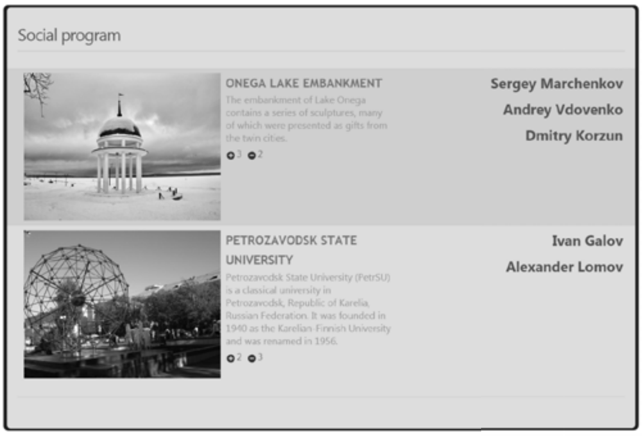


Fig. 10. Screenshots of Social Program service on personal mobile devices (left-to-right, top-to-bottom): POI details with descriptions, voting event, voting result for a POI, a POI list with voting results

Social Program service is written in Visual C# language using .Net Framework. Windows Presentation Foundation (WPF) is used for graphical user interface. SmartSlog SDK is used for interaction with Smart-M3 SIB. Every individual of the ontology is represented in an appropriate C# class with the same properties (called a model in C#). This model is used in building graphical user interface and for constructing JSON objects for Web Page service.



a)

b)

Fig. 11. Screenshots of Social Program service on public screens: a) Agenda service, b) Presentation service

Figure 10 shows screenshots of Social Program service interface. The first three screenshots demonstrates the service view on personal mobile devices of the participants: 1) an information page about a POI including a slide show of found photos, 2) a page with incoming notification from the voting process for this POI and 3) a page of recent voting results for this POI. The last screenshot introduces the full list of all POIs and voting result for each of them. Figure 11 shows similar screenshots for Agenda and Presentation screens.

The Search service is written in Visual C# language using .Net Framework. The serialization mechanism is used for web services responses. Search service inherits the mechanisms of WorldAroundMe search engine [21]. The latter consists of Core and Drivers modules, aiming at the extensibility in adding new web services and functions. The Core module redirects search requests from e-Tourism services to appropriate drivers. In turn, a driver queries its service and receives data on POIs, e.g., links to images.

Web Page service is written in Python language. CherryPy is used as a web engine and Mako as a template engine. There are three main modules. The first one is a request dispatcher, which is implemented with CherryPy. The dispatcher module makes redirection to an appropriate module and invokes actions based on the given URL. The second module is a KP module, which uses Python KPI. The KP module is responsible for operation with the SmartRoom space. The third module is for templates; it is written using HTML, JavaScript and CSS. Additionally, we use jQuery and jQuery Mobile to simplify the code. The templates module constructs web pages with information for users and provides them with ability to perform feedback actions.

We performed a test deployment to measure the performance of implemented services. The deployment involves two personal computers (Linux/Windows) and five smartphones (Windows Phone). Each PC has two core processors of 2.4 GHz, RAM 4 Gb, HDD 2.5 Tb with standard i/o speed. A test program has 10 different places. We measured the execution time of the following operations.

- Search of 20 photos for each place, 10 s on average.
- Transition of a JSON object from Social Program service to Web Page service, 500 ms on average.
- Transition of updated votes from Web Page service to Social Program service, 600 ms on average.

The high search time is due to operation with external web services (Flickr and Panoramio), when the services are requested sequentially. Indeed, parallel requests would lead to better performance. The transition performance of the implemented web-based interaction of services can be concluded as suitable for practical use.

VII. Conclusion

This paper continued our study on development of collaborative work environments. Our approach employs the SmartRoom system to implement various services of different domains within this base system. We showed that such an application domain as tourism can be effectively used for enhancing services of the SmartRoom system. The implemented service for social program, supporting its collaborative construction by conference participants, demonstrates the feasibility of our approach.

ACKNOWLEDGMENT

This research is financially supported by the Ministry of Education and Science of the Russian Federation: project # 14.574.21.0060 (RFMEFI57414X0060) of Federal Target Program "Research and development on priority directions of scientific-technological complex of Russia for 2014–2020" and project # 2.2336.2014/K from the project part of state research assignment. The authors are grateful to all anonymous reviewers for their valuable comments.

REFERENCES

[1] L. Sebastia, I. Garcia, E. Onaindia, and C. Guzman, "e-Tourism: a tourist recommendation and planning application," *International Journal on Artificial Intelligence Tools*, vol. 18, no. 5, pp. 717–738, 2009.

[2] A. Smirnov, A. Kashevnik, S. I. Balandin, and S. Laizane, "Intelligent mobile tourist guide," in *Proc. 13th Int'l Conf. Next Generation Wired/Wireless Networking and 6th Conf. on Internet of Things and Smart Spaces (NEW2AN/ruSMART 2013)*, ser. LNCS 8121, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer-Verlag, Aug. 2013, pp. 94–106.

[3] J. Borrás, A. Moreno, and A. Valls, "Intelligent tourism recommender systems: A survey," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7370–7389, Nov. 2014.

[4] S. Balandin and H. Waris, "Key properties in the development of smart spaces," in *Proc. 5th Int'l Conf. Universal Access in Human-Computer Interaction (UAHCI '09). Part II: Intelligent and Ubiquitous Interaction Environments, LNCS 5615*, C. Stephanidis, Ed. Springer-Verlag, 2009, pp. 3–12.

[5] E. Ovaska, T. S. Cinotti, and A. Toninelli, "The design principles and practices of interoperable smart spaces," in *Advanced Design Approaches to Emerging Software Systems: Principles, Methodology and Tools*, X. Liu and Y. Li, Eds. IGI Global, 2012, pp. 18–47.

[6] K. Kulakov and A. Shabaev, "An approach to creation of smart space-based trip planning service," in *Proc. 16th Conf. of Open Innovations Association FRUCT*, Oct. 2014, pp. 38–44.

[7] A. Varfolomeyev, D. Korzun, A. Ivanovs, and O. Petrina, "Smart personal assistant for historical tourism," in *Recent Advances in Environmental Sciences and Financial Development. Proc. 2nd Int'l Conf. on Environment, Energy, Ecosystems and Development (EEEAD 2014)*, C. Arapatsakos, M. Razeghi, and V. Gekas, Eds., Nov. 2014, pp. 9–15.

[8] D. Korzun, I. Galov, A. Kashevnik, and S. Balandin, "Virtual shared workspace for smart spaces and M3-based case study," in *Proc. 15th Conf. of Open Innovations Association FRUCT*, S. Balandin and U. Trifonova, Eds. ITMO Univeristy, Apr. 2014, pp. 60–68.

[9] 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.

[10] 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.

[11] D. G. Korzun, I. V. Galov, A. M. Kashevnik, N. G. Shilov, 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.

[12] S. Marchenkov and D. Korzun, "User presence detection based on tracking network activity in smartroom," in *Proc. 16th Conf. of Open Innovations Association FRUCT*, Oct. 2014, pp. 45–50.

[13] T. Nguyen, S. W. Loke, T. Torabi, and H. Lu, "On the practicalities of place-based virtual communities: Ontology-based querying, application architecture, and performance," *Expert Systems with Applications*, vol. 41, no. 6, pp. 2859–2873, 2014.

[14] K.-L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, W. Burns, and I. Solheim, "Ontological user modelling and semantic rule-based reasoning for personalisation of help-on-demand services in pervasive environments," *Future Generation Computer Systems*, vol. 34, pp. 97–109, 2014.

[15] M. I. Martín-Vicente, A. Gil-Solla, M. Ramos-Cabrer, J. J. Pazos-Arias, Y. Blanco-Fernández, and M. López-Nores, "A semantic approach to improve neighborhood formation in collaborative recommender systems," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7776–7788, 2014.

[16] C. Evers, R. Kniewel, K. Geihs, and L. Schmidt, "The user in the loop: Enabling user participation for self-adaptive applications," *Future Generation Computer Systems*, vol. 34, pp. 110–123, 2014.

[17] I. Galov and D. Korzun, "A notification model for Smart-M3 applications," in *Proc. 14th Int'l Conf. Next Generation Wired/Wireless Networking and 7th Conf. on Internet of Things and Smart Spaces (NEW2AN/ruSMART 2014), LNCS 8638*, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer-Verlag, Aug. 2014, pp. 121–132.

[18] D. Bernini, F. Fiamberti, D. Micucci, and F. Tisato, "Architectural abstractions for spaces-based communication in smart environments," *J. Ambient Intell. Smart Environ.*, vol. 4, no. 3, pp. 253–277, Aug. 2012.

[19] C. Riva and M. Laitkorpi, "Designing web-based mobile services with rest," in *Service-Oriented Computing-ICSOC 2007 Workshops*. Springer, 2009, pp. 439–450.

[20] C. Ivan and R. Popa, "Cloud based cross platform mobile applications building and integrating cloud services with mobile client applications," *Advances in Computer Science: an International Journal*, vol. 3, no. 2, pp. 69–77, 2014.

[21] A. Vdovenko and D. G. Korzun, "Programmnyy cliyent World Around Me dlya mobil'nykh ustroystv OS Windows Phone," in *Vserossiyskiy Konkurs Nauchno-Issledovatel'skikh Rabot Studentov i Aspirantov*, A. Nemcev, A. . Shtifanov, V. Belenko, and S. Nemcev, Eds. Belgorod, 2012, pp. 555–559, in Russian.