# Event Recording in Smart Room

Ivan Galov*, Rustam Kadirov*, Andrew Vasilev†, Dmitry Korzun*‡,
*Petrozavodsk State University (PetrSU), Russia
†Yaroslavl State University (YarSU), Russia
‡Helsinki Institute for Information Technology (HIIT), Finland
{galov, kadirov, dkorzun}@cs.karelia.ru, vamonster@gmail.com

**Abstract**

Activity tracking in smart space applications allows extending the applications with new features, such as automated generation of activity reports. In this paper we present a generic system – Event Recorder – that potentially can be used with any smart space application. We consider a particular case for Event Recorder: tracking events and producing summary reports of the activity held in Smart Room. The contribution of this study includes use case scenario and ontological representation model of Event Recorder, integration scheme of Event Recorder into the Smart Room system, and evaluation of appropriate visual formats for summary reports.

**Index Terms:** Smart spaces, Activity tracking, Smart Room.

## I. Introduction

A smart space application uses a common space to coordinate participating agents by sharing information [1], [2]. A change in the smart space of application indicates the change in both virtual and physical worlds. Recording of such changes in the smart space and their context would allow to analyze the intrinsic behavior of the application and its users, generate summary or progress reports, create developer support tools, and implement some other interesting features for existing applications.

In this paper we further develop Event Recorder system [3]. It captures arbitrary events in the application smart space and accumulates the change logs. The system uses ontological representation model to achieve a high level of information interoperability and to support integration with potentially any smart space application or a set of them. Basically, the only requirement for integration is that the application specifies in ontology terms which changes to track in the smart space. An important issue we consider in this development is visualization of summaries based on the accumulated change logs.

We apply this system for event recording in Smart Room – the system to automate holding conferences, meetings, and lectures [4]. The integration of Event Recorder extends the Smart Room service set by report generation features for events happening in the room. Smart Room makes particular requirements on visualization formats of summary and progress reports, which can be produced based on the change logs that Event Recorder accumulates. The prototype implementation is used for holding smart spaces sessions of the 13th Conference of the Open Innovations Association FRUCT (http://fruct.org/conference13).

The rest of the paper is organized as follows. Section II introduces the Smart Room system and identifies the need and point for integration of event recording features. Section III presents the Event Recording system and its use case scenario; then we describe the ontological representation model required for information interoperability in the integration. Section IV provides the scheme of integration and studies appropriate report visualization formats for the smart room. Section V concludes the paper and summarizes its main findings.

## II. Smart room

The Smart Room is a system aiming at automation of holding such events as conferences, meetings, and lectures. For the study of this paper, we limit ourselves on the conference automation features; most of them can be extrapolated to other event modes.

The room exploits the smart space concept [1], [5], [6] – a digital environment is constructed where all in-room devices (public screens, mikes, cameras, sensors, etc.), users (event participants with personal mobile devices), and external networked services interoperate by information sharing in a common space (smart room space). The system runs on top of Smart-M3 [7], [2], [8] and inherits the work on smart conference system [9], [10], [11].

The Smart Room applies the service set principle. Every service is implemented as a smart space agent that performs functions on client request and/or publishes to the smart room space information related to a specific component of the room. The service set principle implies that some subset of services is available at the moment; each is offered to participants, either as a public-purpose service or personalized one. The service set structure is shown in Fig. 1. The services are classified into five groups: conference, blogging, internal, external, and sensor.

Conference services automate holding a conference. The group consists of 1) management-service, 2) agenda-service, and 3) projector-service. The management-service handles the conference runtime: forming the conference agenda, starting/ending the conference, controlling speaker's time limits, and switching presentations in accordance with the agenda. The room is equipped with two public screens for visualizing information common to all participants.

The agenda-service implements a user interface of the conference agenda. The output is visualized on a public smart room screen – the agenda screen. In addition to basic information of the conference agenda, the screen displays information about speakers as well as highlights details of the current talk. Note that any ad hoc appearing information related to the talks and important for all participants can be shown on the agenda screen.

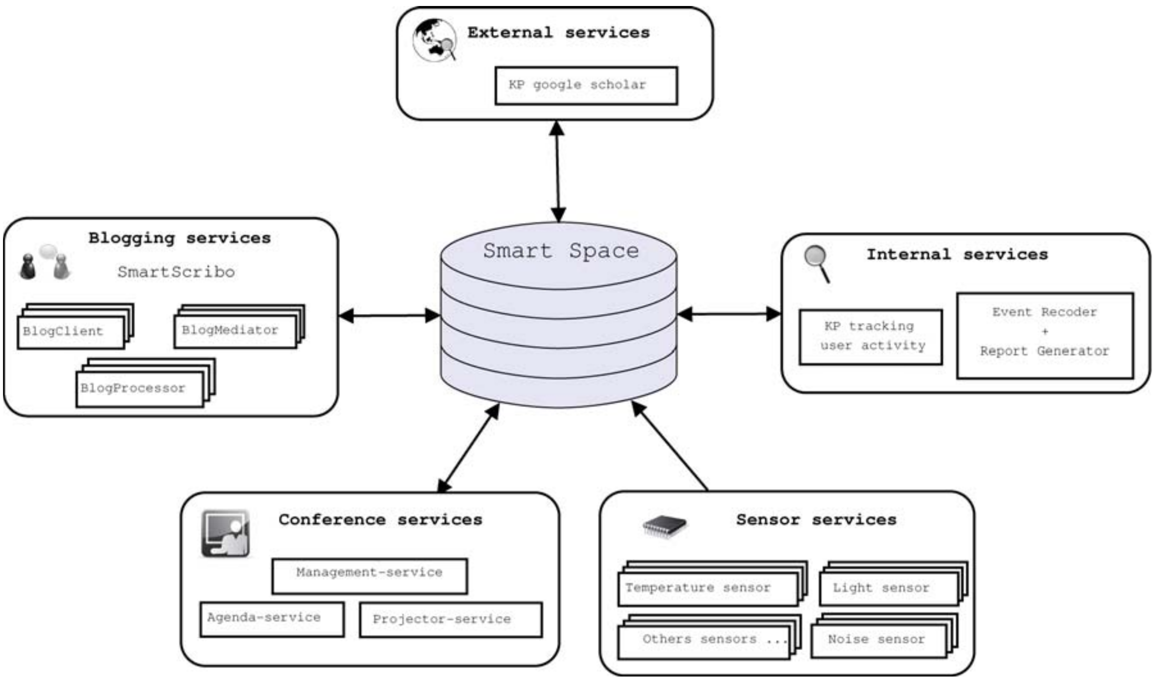



Fig. 1. Structure of the smart room service set, where the smart space is used for sharing all related information

The projector-service shows presentations and makes slide changing. Starting a talk, the management-service requests the projector-service to show the presentation on a public smart room screen – the presentation screen. Then the control is forwarded to the personal mobile device of the speaker and he/she manipulates with the slides (next/previous slides).

Other services groups provide additional features for holding a conference. Blogging services support online discussions during the conference time [9]. The conference blog is constructed at a local or public blog service (e.g., LiveJournal). The blog post structure reflects the agenda such that participants can directly comment every talk.

Sensor services feed the smart room space with sensed information about physical parameters of the room (temperature, noise/illumination levels, etc.). This information is further processed by other services. For instance, when the noise level exceeds a certain bound then the chairman is notified to perform appropriate control actions.

External services feed the smart room space with information contextually accessed from external sources (web services, public data bases, etc.). For instance, the Google Scholar provides a citation index of the current speaker or a list of recent publications related to his/her talk.

Internal services process information in the smart room space to derive additional knowledge about ongoing events. The focus of this paper is on making the Event Recorder system an internal service of Smart Room. The service keeps track of certain events happening during the conference and produces a summary of the track.

Every service is accessed from a smart room client agent installed on personal mobile end-user devices. For instance, the client shows the agenda and talk details, current presentation slide and presentations of other participants, smart room sensor information, conference blog. Current speaker controls his/her presentation from the client. Chairman client (an extension of the smart room client) controls and manipulates with the services, e.g., changing the order of talks in agenda.

Due to a variety of services and human participants, many events can occur in the smart room during the conference. If every service is considered as a source of information then a summary can be generated for this service, showing what happened in a certain period of time. Such summaries are useful for generating reports about the conference. For this problem we further develop the Event Recorder system, which allows tracking specified events in a given application smart space [3].

## III. Event recorder

Event Recorder is a system for event capturing and recording in a given application smart space. It is a generalization of event recording evaluated previously in the smart conference system and smart healthcare application [3], [12].

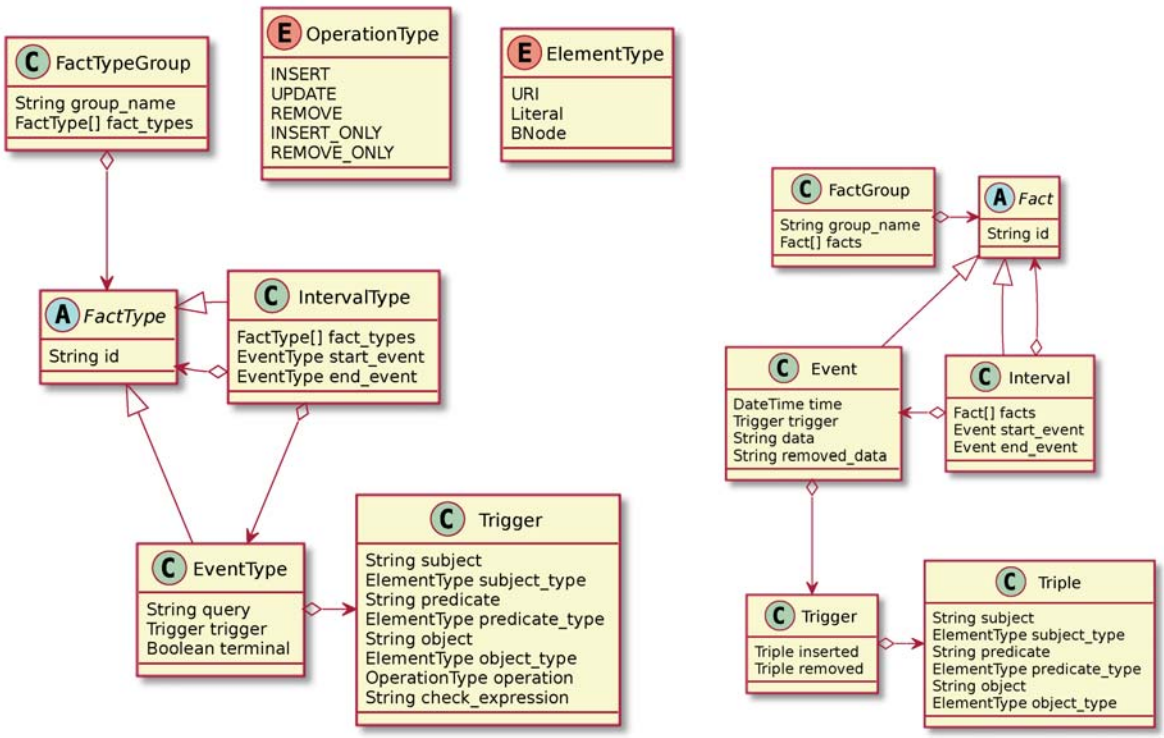
We define an event as anything occurring within the smart space application in a certain place and at given time instance. The location of the event is tied to the location of the application smart space. A change of particular data in the space determines the time instance. Along with events we identify time intervals that can be used to describe lasting occurrences. An interval begins when the particular event happens and lasts until the observation of another event. Therefore, the system tracks time-ordered, non-overlapping events.

The use case scenario for Event Recorder is the following. Initially, the type of events for tracking is identified (application-aware action). Those types are specified in the event recorder configuration file. Then the Event Recorder agent (implemented as a Smart-M3 knowledge processor) starts along with the smart application to be tracked for events. Event Recorder reads the configuration file, connects to the smart space, and continuously tracks appropriate data changes in the smart space. Whenever a change occurs Event Recorder registers the event and collects its description. The process stops when the specified termination event is detected or when Event Recorder is stopped by the user.

Event Recorder continues our previous efforts in context-aware capturing [3], [12]. Our recent contribution primary differs from the previous work by means of data accumulation. Event Recorder provides a snapshot of the application smart space. The user is not mandated to convert that knowledge into a particular visual representation. For the latter we provide a Report Generator tool, an additional application that visualizes the accumulated data. This architectural change results in new ontological models for interoperable data representation.

The ontological model for the configuration file is shown in Fig. 2a. The top level elements are individuals of FactTypeGroup. They allow logical separation of event types of different applications from each other. This could be useful if the user wants to track a set of applications in parallel. Each FactTypeGroup may hold several EventType and IntervalType individuals; each has a unique identifier. It is required to distinguish recorded events from each other in the output log file for the latter processing.

EventType individuals define means to register an event and to gather context information about it. A Trigger individual describes the triple pattern that specifies a data set to track in the smart space in order to register the corresponding event. Any part of the pattern (subject, predicate or object) might be omitted identifying an arbitrary element. The user may also specify the type of modification operation and the check expression to refine the event registration moment. Whenever the trigger fires, Event Recorder retrieves data from the smart space using a SPARQL query [13] specified in the EventType individual field.




(a) Configuration file (b) Output file

Fig. 2. Ontological models of Event Recorder for data file structures

IntervalType individuals not only describe time intervals by providing references to start and end events, but also define the context in which internal events and intervals could be registered. Internally defined events or intervals will be registered by Event Recorder only during the interval lasting. The ability to define internal intervals allows to implement non-trivial logic for event capturing.

Since smart space applications do not operate with the static data sets, it would be troublesome to capture events and their context using a static configuration file. In order to solve this problem we made check expression and query properties adaptable to runtime information by providing means to access data received via subscription notifications and previous queries to the smart space. It is worth mentioning that only interval start events share collected data with the internally defined events. Using this information the user may adapt check expressions or queries to the monitored application context.

The part of Event Recorder configuration file describing a slide change event in Smart Room environment is shown in Fig. 3. The trigger for the event is defined as a modification of the currentSlideNum property in projector-service description. It is sufficient to indicate only predicate to capture the required change in the smart space. The descriptive query for the event requests identifiers of individuals describing the projector-service and the presentation being shown. The expression "%(object)" inside the query is substituted with the object part of triple that fired the trigger. Upon receiving identifiers Event Recorder requests all properties of the individuals from the smart space and adds retrieved data to the event description.

```
id: slide-change-event
trigger:
  predicate: http://www.cs.karelia.ru/smartroom#currentSlideNum
query: >
  Prefix sr: <http://www.cs.karelia.ru/smartroom#>
  SELECT ?projector ?presentation
  WHERE {
    ?projector sr:currentSlideNum %(object) .
    ?projector sr:currentPresentation ?presentation
  }
```

Fig. 3. Slide change event type description

The ontological model of the output format is depicted in Fig. 2b. The model resembles the structure of the configuration file. The topmost elements are individuals of FactGroup that hold all facts registered for the corresponding application. Interval individuals contain fact subgroups and provide description for start and end events. Event individuals describe single events including the time, when the event have been registered, data retrieved from the smart space and data modification that triggered the event. The latter information resides inside Trigger individuals in form of first added and first removed triple from subscription notification.

## IV. Smart room events tracking and reporting

For the smart room purpose the feature of producing detailed reports about a conference and its events is essential. Reports allow to examine operation of the services, track user activity and resemble events that took place. We developed a scheme of the integration of Event Recorder with the Smart Room service set.

We consider two scenarios for integration: summary visualization and online visualization. The first scenario implies that all events are captured during the conference and its summary report is generated at the end of the conference. The second scenario is more complicated for implementation since the event interpretation is formed on-the-fly during the conference. The online visualization is useful to show dynamic information along with the agenda on the corresponding public screen.

At this phase we elaborate only the summary visualization scenario. It includes Report Generator—an autonomous module that creates visual reports based on the change logs of Event Recorder and given visualization templates. The latter are specified by the application (Smart Room in our case).

The integration scheme is shown in Fig. 4. The process of report generation consists of the following steps.

1) The conference organizers create the configuration file for Event Recorder, choose visualization formats, and specify required report types in visualization template file.
2) The configuration file and visualization template file are provided to the Event Recorder agent, which is running on a dedicated computer of the Smart Room system.
3) The Smart Room system forms its smart space and starts all services.
4) Event Recorder connects to the smart room space and initializes the registration of specified events.
5) The chairman opens the conference session; speakers make their talks in accordance with the agenda.
6) During the conference the Smart Room services and clients manipulate with information in the smart room space. Event Recorder tracks the events and collects records in its change log file.
7) The chairman finishes the conference session.

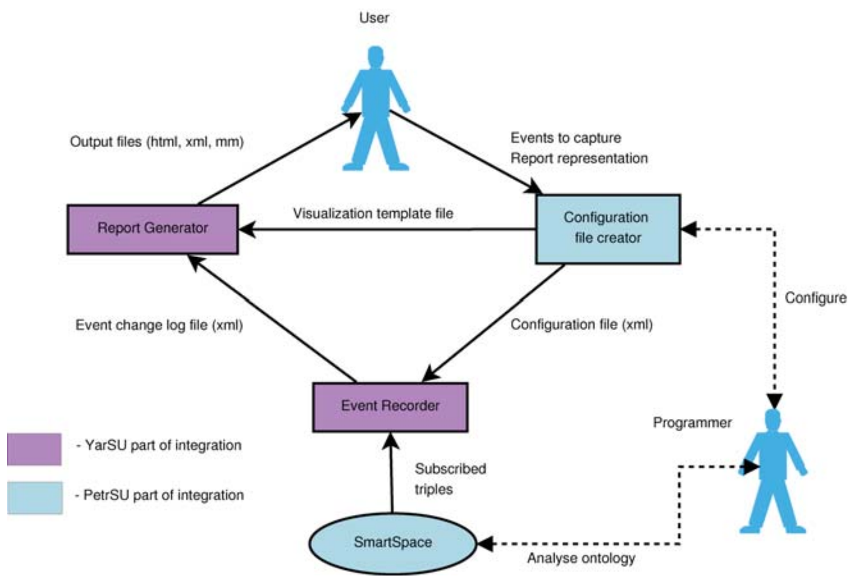



Fig. 4. Scheme of integration of Event Recorder into Smart Room

8) Event Recorder detects the conference termination event and stops.
9) Event Recorder provides the event change log file to Report Generator for generating visual reports about the conference.

For the Smart Room settings, Report Generator creates visual reports in two formats. The first one is a mind map – an associative hierarchical map, consisting of nodes and links between them [11]. A simple example is shown in Fig. 5, where the tree-like structure represents a part of the radial hierarchy of conference presentations. The expressiveness and the comprehensibility of a generated mind map heavily depends on the overall map structure


Fig. 5. Example of the visualization in form of mind map

that is described by the visualization template. For example, the presented mind map can be extended with more associated details describing the events happened during the presentations and the relations between the events. The mind map format is convenient for representing a structure of large events that consist of many other events.

The mind map format is less adequate for displaying a lot of parallel or long-lasting events. In this case the timeline format suits better. Events are displayed in the chronological order, see an example in Fig. 6. In practice, the timeline format is typically used in dynamic application environments with GUI implemented in HTML and JavaScript. A long timeline needs scrolling, so this format is less suitable for static PDF-like reports.

Report Generator tool uses the visualization template file to determine the format of the resulting report. The template also describes how to convert event records into appropriate visual representation. For the mind map format an event template describes a set of nodes, their relations and contents that will be added to the mind map. For the timeline format an event template describes the means to form a brief and a full descriptive strings; the first one will be shown on the timeline itself, the second one will be displayed in an event review window if user selects the event. Event templates are defined for concrete event types, therefore it is possible to have a visualization template relying only on a subset of the recorded events. Thus distinct reports can be created in a various forms using the single event log file.

## V. Conclusion

This paper presented the recent development phase of the Event Recorder system from FRUCT Lab of Yarslavl State University. The key contribution is generalization of event recording features to any smart space application. This property is achieved with ontological representation model that specifies the intra-application events to be recorded.

Our other result is integration of the event recording features into the Smart Room system, developed at FRUCT Lab of Petrozavodsk State University. The integration allows producing summary reports on held events in the smart room space. We studied two formats for visualizing such reports: mind map and timeline.

For the primary steps of further development we plan to support online visualization scenario when reports are formed on-the-fly and immediately provided for public observation.

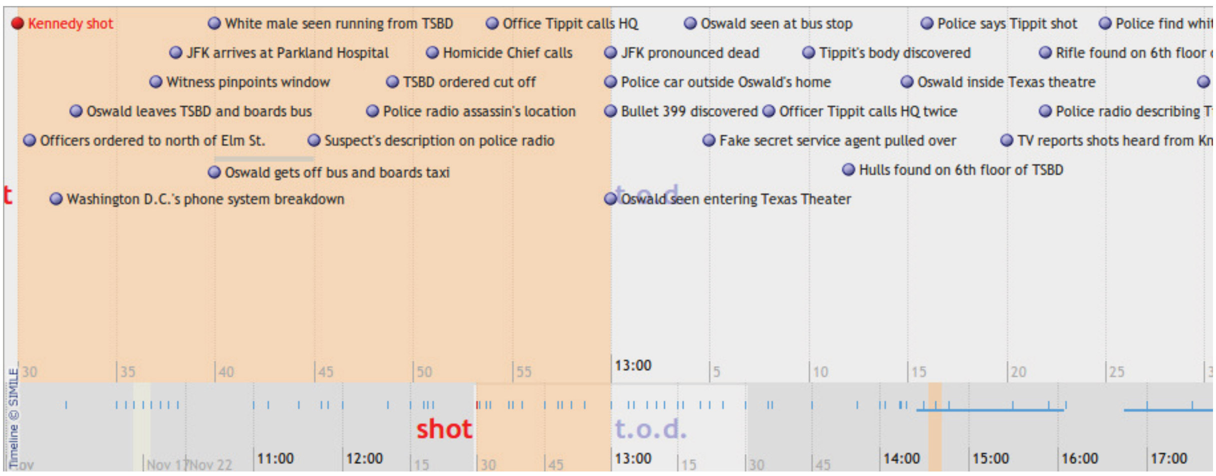



Fig. 6. Example of the timeline format

Also, certain work is needed for the visualization; we plan to study other formats for summary and progress reports applicable in the Smart Room settings.

## Acknowledgment

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. We are grateful to the Open Innovations Association FRUCT for its support and R&D infrastructure. We would also like to thank Sergey I. Balandin, Alexey M. Kashevnik and Ilya V. Paramonov for their feedback and expertise.

## References

[1] D. J. Cook and S. K. Das, "How smart are our environments? an updated look at the state of the art," *Pervasive and Mobile Computing*, vol. 3, no. 2, pp. 53–73, 2007.

[2] 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.

[3] A. Vasilev, E. Krylov, A. Subbotkin, and I. Paramonov, "Event recording system for smart space applications," in *Proc. 12th Conf. of Open Innovations Association FRUCT and Seminar on e-Tourism*. SUAI, Nov. 2012, pp. 147–153.

[4] 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*, Nov. 2012, pp. 239–240.

[5] P. Liuha, A. Lappeteläinen, and J.-P. Soininen, "Smart objects for intelligent applications - first results made open," *ARTEMIS Magazine*, no. 5, pp. 27–29, Oct. 2009.

[6] J. Kiljander, A. Ylisaukko-oja, J. Takalo-Mattila, M. Eteläperä, and J.-P. Soininen, "Enabling semantic technology empowered smart spaces," *Journal of Computer Networks and Communications*, vol. 2012, 2012.

[7] 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.

[8] F. Morandi, L. Roffia, A. D'Elia, F. Vergari, and T. S. Cinotti, "RedSib: a Smart-M3 semantic information broker implementation," in *Proc. 12th Conf. of Open Innovations Association FRUCT and Seminar on e-Tourism*. SUAI, Nov. 2012, pp. 86–98.

[9] 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.

[10] 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.

[11] A. Vasilev, I. Paramonov, E. Dashkova, Y. Koucheryavy, and S. Balandin, "Interaction of HiveMind application with Smart Conference system," in *Proc. IEEE Int'l Conf. on Communications (ICC 2012)*. IEEE, Jun. 2012, pp. 6658–6661.

[12] E. Stankevich, I. Paramonov, and I. Timofeev, "Mobile phone sensors in health applications," in *Proc. 12th Conf. of Open Innovations Association FRUCT and Seminar on e-Tourism*. SUAI, Nov. 2012, pp. 136–141.

[13] E. Prud'hommeaux and A. Seaborne, "SPARQL query language for RDF," W3C Recommendation, Jan. 2008. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/