Active Control by a Mobile Client of Subscription Notifications in Smart Space

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

Abstract—**Smart space supports interaction of multiple participants though information sharing. This kind of interaction is indirect and networked. For detection of changes, which come in parallel from other participants, a participant applies the subscription operation. It allows receiving notifications passively on a regular basis. Absolutely dependable delivery of notifications cannot be guaranteed in modern computing environments. In particular, wireless networks are subject of frequent faults, making the subscription operation less reliable for mobile clients in smart spaces. In this paper, we study the problem of active control by a client in receiving subscription notifications. We introduce a simple mathematical model for control the check interval for subscription notifications, which ensures client adaptation to the notification loss rate.**

I. Introduction

The paradigm of smart spaces comes from ubiquitous computing and augments the emerging concept of Internet of Things (IoT). Smart space forms a service-oriented system with the following base functions [1], [2], [3], [4], [5].

- *Service construction* by dynamic participants based on their interaction through sharing available information and reasoning knowledge over it.
- *Service delivery* to the end-users through multimodal interfaces from the diversity of user interface devices in the computing environment.

Human (end-users) and non-human (machines, smart IoT objects) participants of a smart space are represented by software agents that run on devices of the IoT-aware computing environment. One of the application development platforms for smart spaces is Smart-M3 [6], which is open source[1] and oriented to research prototyping. A smart space agent is also called a knowledge processor (KP).

Let us mention the SmartRoom system [7], [8] as a reference example of a smart space application. The system aims at intelligent assistance (in a specifically equipped room) for such collaborative activity as conferences, meeting, or lectures. Software agents run on computing devices installed locally in the room or on computers accessible in the corporate network and Internet. This paper focuses on the class of agents responsible for personalized service delivery to the end-users. In the SmartRoom example, this class includes mobile clients [9], [10], [11], [12], [13]. They run on such personal devices as smartphones, tablets, or netbooks. Many Smart-M3 applications are now in development where the role of mobile clients become essential. In particular, e-Tourism applications are discussed in [14], [15], [16], [8].

A key mechanism for programming advanced interaction of Smart-M3 based agents is the subscription operation from publish/subscribe systems [17], [18], [19]. Let a mobile client be a subscriber, i.e., it specifies a persistent query to an interested part of the shared smart space content. Whenever the specified part is changed the client receives a subscription notification. Changes are due to parallel activity of other agents, which act as publishers in this interaction. In Smart-M3, each subscription requires its client to establish a network connection. Changes are controlled on the smart space side, and the corresponding notifications are sent to the client, which acts as a passive receiver. Let us call such subscription notifications as passive.

For mobile clients, the subscription operation is affected by losses of subscription notifications. This type of failures is especially common in wireless network settings [20], [21]. In this paper, we study the problem of active control by a mobile client itself for receiving subscription notifications. That is, each client can actively check changes, leading to active subscription notifications. This additional control allows mitigating the consequences of notification losses. On the other hand, frequent checks from the client lead to excessive resource consumption, and the check interval has to be adapted to the current fault rate.

The tradeoff of passive and active notifications can be described formally. We propose a simple mathematical model, which supports an adaptive strategy for a mobile client to determine reasonably the check interval for active notifications. Based on this model, we performed a simulation study. The experiment results indicate that the proposed model outperforms straightforward non-adaptive strategies.

The rest of the paper is organized as follows. Section II introduces the subscription operation for networked interaction of agents in smart spaces. Section III overviews the related work. Section IV considers the fault tolerance problem of subscription in the case of mobile clients and defines the

[1]Source code is available at http://sourceforge.net/projects/smart-m3 under the BSD license.

key parameters at the client side. Section V presents our mathematical model that describes how a client can adapt its check interval for active subscription notifications. Section VI describes our simulation study of the model efficiency and compares the model with straightforward non-adaptive strategies. Section VII concludes the paper.

II. Publish/subscribe in smart spaces

The communication model of publish/subscribe systems is widely used for organizing interactions of multiple networked participants [17], [18], [19]. A typical case is sparse-connected service-oriented systems where a lot of informational sources and destinations are used in parallel and asynchronously. Publish/subscribe systems have proven to be an effective way to deliver information of interest to many receivers, in a proactive style, and as soon as possible.

The roles of interacting participants are divided into publishers and subscribers. A publisher produces some informational content. In smart spaces, typical examples are information sources in the Internet, runtime-generated content from ongoing activity, or data sensed from the environment. If a subscriber is interested in certain content, then it specifies its interest as a persistent query—subscription. Whenever the specified content is changed (by some publishers) the system notifies the subscriber about. A subscriber is primarily interested in content, not in its publisher. A change can affect many subscribers. The specified content can be changed by different publishers. Participating in a smart space, every agent may combine both roles: publisher and subscriber.

From the subscriber point of view, passive subscription notifications are very convenient since they are delivered without explicit control from the subscriber itself. The common way to implement this subscriber-passive delivery is employing an information broker [18]. On one hand, the broker serves update requests from publishers, thus keeping a common information storage (e.g., in the blackboard style). On the other hand, the broker maintains all subscriptions and can map all incoming content changes to specified subscribers' interests.

In Smart-M3 [6], a broker is called semantic information broker (SIB). For each subscription, the SIB maintains a persistent network connection established by the agent's request [19], [22]. Knowing the set of all subscriptions, the SIB regularly checks that they are alive, removing any subscription if its network connection is lost. The Smart-M3 platform performs in the best effort style. A notification *should* be delivered to the subscriber if a related change in the content has happened. Some notifications can be unsent by SIB due to its overload or operability faults. SIB does not check delivery for already sent notifications. A notification can be sent although the underlying network connection is broken on the subscribers side. The above properties do not ensure the dependable delivery even if reliable network protocols are used, such as TCP.

For the considered class of mobile clients, the subscription fault tolerance is essentially affected due to the specifics of wireless network communication (Wi-Fi, 3G, etc.). A network connection between the client and SIB for subscription can be aborted on (i) the SIB side or (ii) the client side. In case (i), SIB sends no notification, and the client needs to reestablish

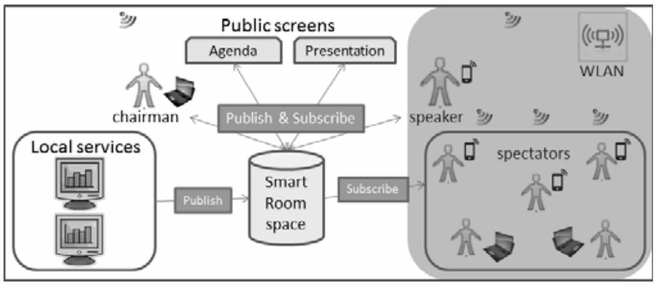

Fig. 1. End-users interact within the SmartRoom system using personal mobile devices. Subscription from them is subject of frequent failures

the network connection. The unsent notifications are lost since SIB does not use buffering. Case (ii) happens frequently due to the mobile device losses (temporarily) the network access or even switches to another wireless network. No notification the SIB is sending can reach the client, and it needs to reestablish the network connection. As in case (i), some notifications are not delivered.

Both cases we experimentally discovered in the SmartRoom system. The architectural view with emphasis on mobile clients is shown in Fig. 1. First, the fault tolerance of subscription requires solutions for manual and automated network connection recovery by a mobile client itself. For the Smart-M3 case, this problem was considered in [12]. Second, each client needs additional mechanisms for reducing the number of undelivered notifications. The obvious way to solve the second problem is augmenting the passive notification delivery the SIB provides with active control the mobile client performs individually on its own.

Recall that the considered case has certain specifics that must be taken into account in development of the active control solutions. In contrast to many publish/subscribe systems, SIB is a single instance. Subscription delivery is not affected by latent inter-broker communication. Subscribers are mobile and their participation is very dynamic. There can be sporadic periods of burst-like activity. The wireless network quality is also subject of frequent changes, including network connection losses and switching to other networks.

III. Related work

Pongthawornkamol *et al.* [20] proposed a quantitative, analytical model to predict the effect of failures and commonly used recovery techniques to the quality of service each subscriber receives. The model estimates each subscriber's real-time reliability, which is the percentage of events that are successfully delivered to each subscriber on time. The model is applicable for publish/subscribe systems with a network of information brokers, where failures are due to routing in this network. In contrast to our Smart-M3 case, the fault-tolerant mechanisms are mostly intended for implementation on the broker side, e.g., keeping timestamps to detect undelivered notifications or buffering notifications with waiting for explicit delivery acknowledgement.

Shabangu [21] overviewed the problem of guaranteed timely delivery in publish/subscribe systems. A straightforward solution was proposed that supports real-time and guarantee delivery of notifications for latest updates. In contrast to our work, that class of solutions requires an additional component to monitor all system updates and to check that subscribers have received notifications.

Baldoni *et al.* [23] presented an analytical model for measuring the amount of losses on the subscriber side. They considered two delays: the subscription delay and the diffusion delay. Depending on the behavior of these delays, various models for predicting the notification loss can be constructed. Such a model cannot be applied for Smart-M3 mobile clients since the subscription notification loss is not due to inter-broker communication.

In our work, we consider the case when a subscriber (mobile client) spends individually its own resources to actively request the broker for new notifications. The resource consumption is controllable by every subscriber itself due to individual selection of the check interval. The latter is made adaptive to the observable notification loss rate.

IV. SUBSCRIPTION PARAMETERS AT THE CLIENT SIDE

Let a mobile client make a subscription: the content of interest is specified and a network connection between the client and its SIB is established (e.g., a TCP connection). Consider the parameters that describe the process of subscription notifications delivery on the client side.

Notifications arrive sequentially to the client. Let i be the sequence number of a notification the client successfully has received by the subscription. Let t_i be the time interval between delivered notifications $i-1$ and i. When the notification delivery is passive then t_i is determined by the update rate of specified content and increases when some notifications are lost.

Let k_i be the observed number of losses in the interval t_i, i.e., between successfully delivered notifications $i-1$ and i. Indeed, estimation of k_i by a client is a self-contained nontrivial problem. We assume that there are some mechanisms. In the simplest case, the client can assign $k_i = 1$ if it observes an evident failure, e.g., the TCP connection was broken and then repaired.

Denote by $\lambda = \lambda_i = k_i/t_i$ the estimation of instant rate for the notification loss. The client is interested in minimizing λ. It can be achieved by the client that explicitly queries the SIB and checks for updates. Active subscription notifications appear at the client side. This action produces i, leading to interleaving the passive and active notification delivery in sequence $i = 1, 2, \ldots$. If active notification detects no update then let $k_i = 0$.

With active notifications, t_i becomes a control variable for the client. Intuitively, selecting small values for t_i the client tries to receive all notifications, thus $k_i = 0$ is almost always and λ approaches zero. On the other hand, too small t_i leads to frequent requests and make the client overloaded (as well as the SIB and network).

Note that this strategy does not ensure full absence of notification loss. For instance, the SIB can be temporarily unavailable for certain time (e.g., network connection is lost), and t_i cannot be made smaller than the time of SIB unavailability. Moreover, when many agents operate with the SIB then the high workload due to small t_i from multiple clients may result in more losses.

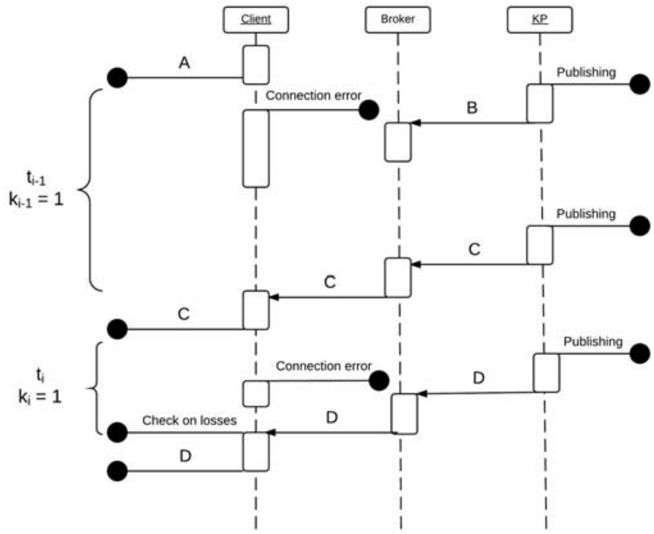

Fig. 2. Fragment of steps in subscription notification delivery

Figure 2 illustrates the parameterized subscription process. Informational content of the smart space is updated due to activity of some agents (KP). They publish some information using the SIB. The latter should notify the client whenever the update affects the specified part of content. Notification A is delivered successfully. When notification B is sent from the broker to the client, the network connection fails, and notification B is not delivered. The client successfully receives notification C, computes t_{i-1} and estimates $k_{i-1} = 1$. To diminish the observed loss of notifications the client makes an explicit check for updates. The check detects that the connection is lost, the client repairs it and then receives notification D and sets $k_i = 1$.

V. MATHEMATICAL MODEL FOR CONTROL OF NOTIFICATION CHECK INTERVAL

Let us introduce a simple model that describes adaptive client's control of check intervals t_i in terms of observable losses k_i. The model will be used then for constructing an adaptive strategy of the client in active subscription notification delivery. Assume that some initial t_0 is always defined. Consider the client's behavior evolving in time as $i = 1, 2, \ldots$.

Let the client have observed no losses in t_{i-1}, i.e., $k_{i-1} = 0$. Then let the client increase t_i in the following additive style. That is, the client observes indication that the system state becomes good. To save resources, the client makes a moderate increment with a fixed parameter $\delta > 0$.

$$t_i = t_{i-1} + \delta. \tag{1}$$

The increment is conservative since high increase of t_i is a clear risk for suffering a burst of losses.

Now let the client have observed certain losses in t_{i-1}, i.e., $k_{i-1} > 0$. The client has to reduce t_i to decrease the number of losses on the next interval. The reduction is multiplicative since the client is interested in fast achieving $k_i = 0$ in the next observation round. The reaction also should take into account the previous observations, thus we apply Multiplicative average with a fixed parameter $0 < \alpha < 1$.

$$t_i = \alpha t_{i-1} + (1-\alpha)\frac{t_{i-1}}{k_{i-1}+1}. \tag{2}$$

Combining (1) and (2) we construct the recurrent system

$$t_i = \begin{cases} t_{i-1} + \delta & \text{if } k_{i-1} = 0, \\ \dfrac{1+\alpha k_{i-1}}{k_{i-1}+1} t_{i-1} & \text{if } k_{i-1} > 0. \end{cases} \tag{3}$$

Note that (3) is valid for active subscription notifications only. When proactive notification i is delivered then t_i cannot be set by the client.

There are two simple cases of (3) that admit analytical solutions and show the extreme cases for adaptive behavior.

Case 1 (no loss): $k_i = 0$ for $i = 1, 2, \ldots$. Given fixed $t_0 > 0$ and $\delta > 0$ the solution to (3) is

$$t_i = t_0 + \delta(i-1), \quad i = 1, 2, \ldots. \tag{4}$$

Therefore, t_i grows linearly to infinity. This extreme case shows that when no losses then the client converges to the state without active subscription notifications.

Case 2 (constant loss): $k_i = k > 0$ for $i = 1, 2, \ldots$. Given fixed $t_0 > 0$ and $0 < \alpha < 1$ the solution to (3) is

$$t_i = t_0 \left(\frac{\alpha k + 1}{k+1}\right)^{i-1}, \quad i = 1, 2, \ldots. \tag{5}$$

Since $0 < (\alpha k + 1)/(k+1) < 1$ the check interval decreases Multiplicatively to zero. This extreme case shows that the client quickly converges to the case with domination of active subscription notifications. It is worth to note that in practical settings, periods of constant loss are non-scalable, and reduction of t_i eliminates them.

VI. Simulation experiments

For evaluation and comparison of the proposed model for adaptive strategy, we performed simulation experiments.

The client's adaptive behavior depends on distribution of subscription notification losses. We analyze the following two distributions, where values for k_i are generated as non-negative integers.

1) Value for k_i is selected from $[at_i, bt_i]$ uniformly at random.
2) Value for k_i follows the Poisson distribution with mean and variance λt_i for $\lambda > 0$.

In our simulation we take $a = 0$ and $b = 0.1$ for random losses and $\lambda = 0.05$ for losses following the Poisson distribution. The dependence of k_i on t_i assumes that the number of losses is proportional to the time interval length. That is, both cases are unified such that one notification is lost in every 20 s on average.

Table I summarizes the client strategies we experimented with. The initial value is $t_0 = 20$ s, which confirms the intuition that one loss happens on this interval on average.

TABLE I. Parameters of the experimented strategies

Strategy		Description
Parameter	Value	
Adaptive strategy		The strategy follows (3). $\alpha = 0.5$ trades off previous and recent observations equally. $\delta = 20$ s is equal to the interval for one loss on average.
α	0.5	
δ	20	
Multiplicative decrease		When $k_{i-1} > 0$ the check interval t_i is reduced by two. If $k_{i-1} = 0$ then set $t_i = t_0$.
factor	0,5	
Random selection		Random strategy when t_i is selected from interval (a, b) at random.
a	10	
b	30	
Constant check interval		The check interval is always set $t_i = t_0$.

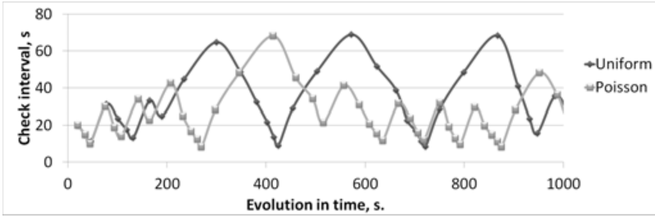

Fig. 3. Behavior of the adaptive strategy for uniform and Poisson distributions of notifications losses

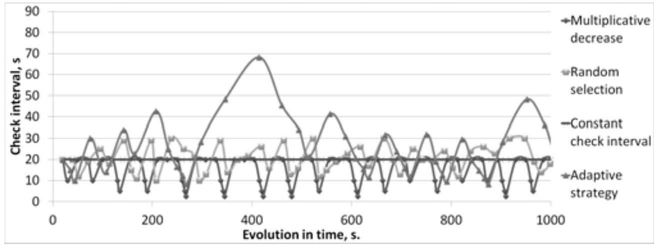

Fig. 4. Comparison of strategies for control of the check interval

For each distribution of losses we consider the evolution of the client for $0 < i \leq 100$.

Figure 3 shows the client's control of t_i with the adaptive strategy of model (3) and for the two distributions of notification losses. The time is shown in seconds, instead of sequence numbers i. The check interval t_i decreases linearly, although (5) describes multiplicative reduction if time units are i. In random losses, non-zero k_i appears very likely; it leads to "sawtooth" behavior with big peaks. In Poisson losses, the case $k_i = 0$ has high probability, leading to many peaks of smaller size.

The second type of experiments compares different strategies for a mobile client. The experimented strategies are summarized in Table I. All strategies are studied for the Poisson distribution of losses. The distribution parameters are the same as in the previous experiments.

Figure 4 compares the behavior of the strategies. The all non-adaptive strategies result in low t_i, so requiring more resources from the client. The multiplicative decrease strategy leads to high oscillation. We expect that the proposed adaptive strategy behaves even better (compared with the others) if the distribution of losses is time-dependent.

Now consider some efficiency metrics. Let k_{avg} be the average number of losses over the entire evolution. Similarly,

t_{avg} is average length of the check interval.

$$k_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} k_i, \quad t_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} t_i.$$

Intuitively, the client is interested in $k_{\text{avg}} \to \min$ and $t_{\text{avg}} \to \max$. Table II shows a comparison for different strategies. The multiplicative decrease strategy has the worst result in time and the best one in the losses count. It is a result of its semi-adaptive character, when the reduction is fast and interval increase is conservative. Reasonable efficiency shows the random selection and constant check interval. Note that their average efficiency can be essentially degraded if the distribution of losses becomes time-dependent. We can conclude that our adaptive algorithm outperforms the other strategies.

TABLE II. EFFICIENCY METRICS OF THE EXPERIMENTED STRATEGIES

Metric	Multiplicative decrease	Random selection	Constant check interval	Adaptive strategy
k_{avg}	0,59	1,19	0,89	1,23
t_{avg}	14,23	19,87	20	28,8
$\lambda = k_{\text{avg}}/t_{\text{avg}}$	0,042	0,06	0,045	0,041
$\lambda_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n}\frac{k_i}{t_i}$	0,078	0,06	0,045	0,043

Figure 5 shows the count of losses (k_i) and the response behavior of the adaptive strategy. The distribution of losses is Poisson (cf. Fig. 3). It illustrates how the check interval is adapted in accordance with the number of losses.

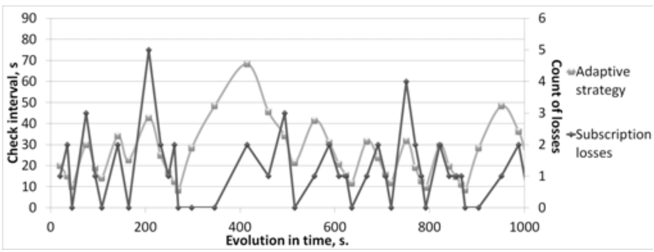

Fig. 5. Count of losses (k_i) and the response of adaptive strategy (t_i)

Let us analyze the dependence of the adaptive strategy on δ, see model (3). Table III shows experiments where δ is varied. Smaller values for δ leads to less losses. Bigger values reduce the load the client shifts to the SIB.

TABLE III. VARIATION OF δ IN THE ADAPTIVE STRATEGY

Parameters	Variation			
δ	25	35	45	55
Metric	**Values**			
k_{avg}	1,41	1,57	1,96	1,98
t_{avg}	34,6	35,2	37,4	39,5
$\lambda = k_{\text{avg}}/t_{\text{avg}}$	0,041	0,045	0,052	0,05
$\lambda_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n}\frac{k_i}{t_i}$	0,041	0,045	0,048	0,051

VII. CONCLUSION

This paper has studied the problem of subscription fault tolerance for the case of a mobile client interacts with other participant in a smart space. Such a client subscribes to certain part of shared information and should receive notification on their changes on a regular basis. On one hand, if the client performs none or rare active checks then it can suffer from losses of some notifications. On the other hand, high pooling from the client leads to excessive load. We proposed a simple mathematical model that aims at achieving a tradeoff at the client side when the client adapts its check interval to the current observations of the notification loss. Our simulation experiments show that the solution is reasonability compared with non-adaptive client strategies. Now we start to apply the model in real settings: as a testbed we use the SmartRoom system and its mobile clients (sourceforge.net/projects/smartroom/files/clients/).

ACKNOWLEDGMENT

This research is financially supported by project # 1481 from the basic part of state research assignment # 2014/154 and by project # 2.2336.2014/K from the project part of state research assignment of the Ministry of Education and Science of the Russian Federation. The article was published with financial support from the Strategic Development Program of Petrozavodsk State University. The authors would like to thank all anonymous reviewers for their valuable comments.

REFERENCES

[1] 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.

[2] A. Smirnov, A. Kashnevik, N. Shilov, I. Oliver, S. Balandin, and S. Boldyrev, "Anonymous agent coordination in smart spaces: State-of-the-art," in *Proc. 9th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN'09) and 2nd Conf. Smart Spaces (ruSMART'09), LNCS 5764*. Springer-Verlag, 2009, pp. 42–51.

[3] 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, pp. 1–14, 2012.

[4] 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.

[5] D. Korzun, S. Balandin, and A. Gurtov, "Deployment of Smart Spaces in Internet of Things: Overview of the design challenges," in *Proc. 13th Int'l Conf. Next Generation Wired/Wireless Networking and 6th Conf. on Internet of Things and Smart Spaces (NEW2AN/ruSMART 2013), LNCS 8121*, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer-Verlag, Aug. 2013, pp. 48–59.

[6] 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.

[7] D. Korzun, I. Galov, and S. Balandin, "Development of smart room services on top of smart-m3," in *Proc. 14th Conf. of Open Innovations Association FRUCT*, S. Balandin and U. Trifonova, Eds. SUAI, Nov. 2013, pp. 37–44.

[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 and D. Korzun, "Access to smart room service set from end-user mobile devices," in *Proc. 12th Conf. of Open Innovations Association FRUCT and Seminar on e-Tourism*, S. Balandin and A. Ovchinnikov, Eds. SUAI, Nov. 2012, pp. 300–302.

[10] 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.

[11] P. Kovyrshin and D. Korzun, "Android smartphone as a microphone in SmartRoom system," in *Proc. 15th Conf. Open Innovations Framework Program FRUCT*, S. Balandin and U. Trifonova, Eds. ITMO Univeristy, Apr. 2014, pp. 198–199.

[12] A. Vdovenko and D. Korzun, "Delivery of smartroom services using mobile clients," in *Proc. 14th Conf. Open Innovations Framework Program FRUCT*, S. Balandin and U. Trifonova, Eds. SUAI, Nov. 2013, pp. 215–216.

[13] P. Kovyrshin and D. Korzun, "Programming Android client for Smart-M3 applications: SmartRoom case study," in *Proc. 14th Conf. Open Innovations Framework Program FRUCT*, S. Balandin and U. Trifonova, Eds. SUAI, Nov. 2013, pp. 197–198.

[14] A. Smirnov, N. Shilov, A. Kashevnik, N. Teslya, and S. Laizane, "Smart space-based ridesharing service in e-Tourism application for Karelia region accessibility: Ontology-based approach and implementation," in *Proc. 8th Int'l Joint Conf. on Software Technologies (ICSOFT 2013)*, J. Cordeiro, D. A. Marca, and M. van Sinderen, Eds. SciTePress, Jul. 2013, pp. 591–598.

[15] 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.

[16] A. Smirnov, A. Kashevnik, A. Ponomarev, N. Shilov, M. Schekotov, and N. Teslya, "Recommendation system for tourist attraction information service," in *Proc. 14th Conf. of Open Innovations Association FRUCT*, S. Balandin and U. Trifonova, Eds. SUAI, Nov. 2014, pp. 148–155.

[17] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surv.*, vol. 35, pp. 114–131, June 2003.

[18] R. Baldoni, R. Beraldi, S. Tucci Piergiovanni, and A. Virgillito, "On the modelling of publish/subscribe communication systems," *Concurr. Comput.: Pract. Exper.*, vol. 17, no. 12, pp. 1471–1495, 2005.

[19] A. A. Lomov and D. G. Korzun, "Subscription operation in Smart-M3," in *Proc. 10th Conf. of Open Innovations Association FRUCT and 2nd Finnish–Russian Mobile Linux Summit*, S. Balandin and A. Ovchinnikov, Eds. SUAI, Nov. 2011, pp. 83–94.

[20] T. Pongthawornkamol, K. Nahrstedt, and G. Wang, "Reliability and timeliness analysis of fault-tolerant distributed publish / subscribe systems," in *Proc. the 10th Int'l Conf. on Autonomic Computing (ICAC 13)*. USENIX, 2013, pp. 247–257.

[21] P. S. Shabangu, "Guaranteed real-time delivery of context-aware messages in publish/subscribe systems," Master's thesis, Department of Computer Science, University of Zululand, The South African Republic, 2007.

[22] 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*, S. Balandin and A. Ovchinnikov, Eds. SUAI, Nov. 2012, pp. 86–98.

[23] R. Baldoni, R. Beraldi, S. Tucci Piergiovanni, and A. Virgillito, "Measuring notification loss in publish/subscribe communication systems," in *Proc. the 10th IEEE Pacific Rim Int'l Symp. on Dependable Computing*, 2004, pp. 84–93.