

Proactive Personalized Mobile Multi-blogging Service on Smart-M3

Dmitry G. Korzun^{1,2}, Ivan V. Galov¹ and Sergey I. Balandin³

¹Department of Computer Science, Petrozavodsk State University – PetrSU, Republic of Karelia, Russia

²Helsinki Institute for Information Technology – HIIT, Aalto University, Finland

³FRUCT Oy, Helsinki, Finland

This paper discusses the project targeted at development of proactive, personalized and composite services for the blogosphere. By applying the smart spaces paradigm we have developed an approach for composition of blog spaces. We contribute formal technique for construction of proactive and personalized service for bloggers as well as for its integration with external applications. The proposed technique is implemented in SmartScribo application on top of the Smart-M3 platform. The developed prototype solution has confirmed that the approach results in services with the next quality level.

Keywords: smart spaces, Smart-M3, blogging, personalized, proactive, composite

1. Introduction

The giant data collections have been accumulated by Internet services and the growth continues. The data and the multitude of services can be used by a person independently of her/his location. This requires reconsidering approaches of service provision. A good reference case to evaluate new solutions is blogging, which certainly faces the size problem [3, 9].

A promising paradigm is smart spaces [4, 1]. It allows a multi-device system to use shared view of dynamic resources and context-aware services. Smart-M3 is an open source information sharing platform for prototyping smart space applications [8]. Smart-M3 application consists of distributed agents (Knowledge Processor, KP) sharing the smart information space.

Instead of information, the Smart-M3 platform operates with knowledge, as the provided storage allows representing habitual data, relations

between them and even such information as services and computations. Each smart space is assigned with an RDF triple store accessible via a Semantic Information Broker (SIB). The RDF representation of knowledge allows easy exchange and linkage, making cross-domain interoperability straightforward.

Based on [11], we can argue that the smart spaces applications with personalized, composite and proactive properties will be able to provide efficient and a user-friendly solution for blogging. The personalized property allows a user to access a service according to her/his personal interest and current needs. A good discussion on this can be found in [4]. The composite property allows temporal situational composition of semantically related knowledge or services, forming the knowledge relevant at the given moment. For more details, please refer to [5]. The proactive property allows a user to be provided with a service she/he is interested in at the given moment of time, see more in [12].

In this paper, we consider the design problem for Smart-M3 applications with personalized, composite and proactive properties. We propose space-based approach to built proactive and personalized services for blogging. We contribute formal technique based on dynamic smart spaces composition when application participants meet only with a fraction of huge data pool needed at a moment. Space sharing activity is synchronized within the application, yet it can be integrated to other smart space services.

SmartScribo is a Smart-M3 application for semantic mobile multi-blogging [13]. Mutli-blogging states that any user can interact in parallel with multiple blogs at many blog ser-

VICES. Users access the blogosphere using their personal mobile devices as clients. Personal and context information is semantically related with blog data and activity in personal blogger spaces.

The application smart space is fed from many heterogeneous sources and shared by many participants, which also dynamically join and leave. As a result, personal blogger spaces are composed into various subspaces. This process is proactive and each composition can initiate a personalized service.

The rest of the paper is organized as follows. Section 2 overviews the space-based multi-agent architecture of SmartScribo application. Section 3 introduces our technique of personal spaces present within the entire application smart space. Section 4 describes our space composition method that preserves the application only with a certain fraction of data from the blogosphere. Section 5 provides our notification model for agent synchronization, which supports proactive personalized services. Section 6 addresses the concern of application integration when the knowledge is kept in independent smart spaces. Section 7 summarizes our outcomes.

2. SmartScribo Smart Spaces Application

The blogosphere is made up of all blogs on all blog services and of all interconnections between blogs, forming a social network in which everyone can publish messages. A blog-service is a website with the content partitioned onto blogs. A blog consists of several discussions of a given author. Each discussion starts with

a post and then follows tree-like structure with comments, comments to comments and so on.

SmartScribo connects mobile bloggers with their blogs hosted at multiple blog-services [13]. We use term “user” for a blogger who uses SmartScribo. A user can perform the same blogging functions as a non-SmartScribo user and additionally benefits from semantic features, e.g., parallel access to multiple blog services and proactive blog recommendations based on user’s context and activity of other bloggers. For more details on semantic blogging, refer to [9].

The generic smart space model was introduced in [11]. A space is $S = (n, I, \rho)$, where n is its name, I is an information set of tuples (content) and ρ is a set of rules to deduce new knowledge from I . In Smart-M3, I is represented using the RDF model, i.e., I is a set RDF triples. We consider a particular case when ρ is ontology O represented with the OWL model. Then I becomes structured with classes and properties from O , deduction in S uses the ontology graph of individuals. We also skip n from the notation and refer to the space by S . The simplified definition of a smart space is

$$S = (I, O),$$

where I is a set of RDF-triples and O is an OWL ontology.

SmartScribo represents the blogosphere in the application smart space, where user and blog data are kept dynamically and semantically related. The SmartScribo blogosphere space acts as a mediator infrastructure reflecting a part of the blogosphere that is actually used by the current set of users. Further, we denote the blogosphere smart space $S = (I, O)$, where I is the RDF triple store at SIB and O is the blogosphere ontology of SmartScribo (Figure 1).

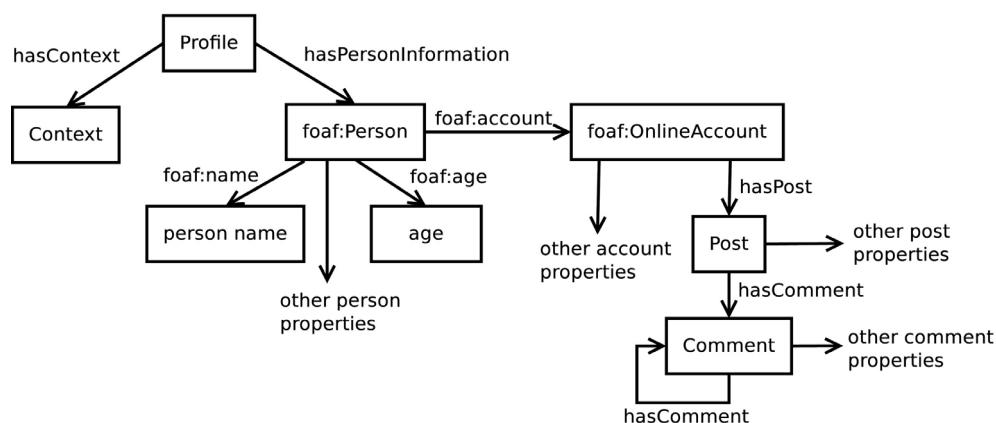


Figure 1. Basic OWL classes and their properties in the blogosphere ontology O .

The SmartScribo architecture is shown in Figure 2. The Smart-M3 application design principle states that the application space is shared by loosely coupled KPs, each of which implements a particular specialization. They communicate by modifying and querying the knowledge in the space.

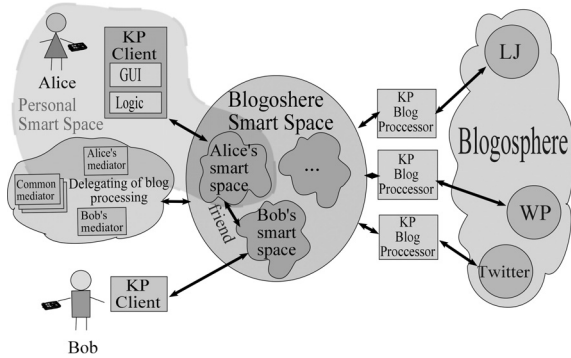


Figure 2. Architecture of SmartScribo.

There are three types of SmartScribo KPs: blog clients, blog processors and blog mediators. Users interact with multiple blogs at many blog services. Each user has her/his own smart space in S .

Clients run on user devices, operate with blogs locally, share personal data and context of the users and access the blogosphere smart space for operations at blog services.

Blog processors track blogger personal spaces relaying them with particular blog-services. Basically, a blog processor transfers new or updated blog messages to their blog service and

retrieve from the service those blog messages that are of user's current interest.

Mediators extend blogging with smart semantic features. One of the many possible examples is blog recommendation when a blog mediator tracks activity of the user and initiates search of semantically close blogs in the blogosphere.

3. Personalization with Blogger Spaces

Denote $S_u = (I_u, O)$ the personal smart space of a user u . The high-level structure is shown in Figure 3. The space S_u keeps (i) blog data (from blog services), (ii) common personal information (from user profile) and (iii) current context state (from user's devices). Hence

$$I_u = [P_u \cup C_u \cup D_u]_O$$

where P_u is personal data, C_u is context, D_u is blog data. Information consistency and the relation between the subsets is ensured by the common ontology O .

A user u shares (partially) her/his personal profile in P_u . If u has several devices, then an aggregated profile is shared. It describes permanent or long-term data, i.e., name, age, e-mail, etc. They are not blog-specific and can be used in other domains. We use FOAF standard specification [7] for the representation; it supports an opportunity for integrating into other applications.

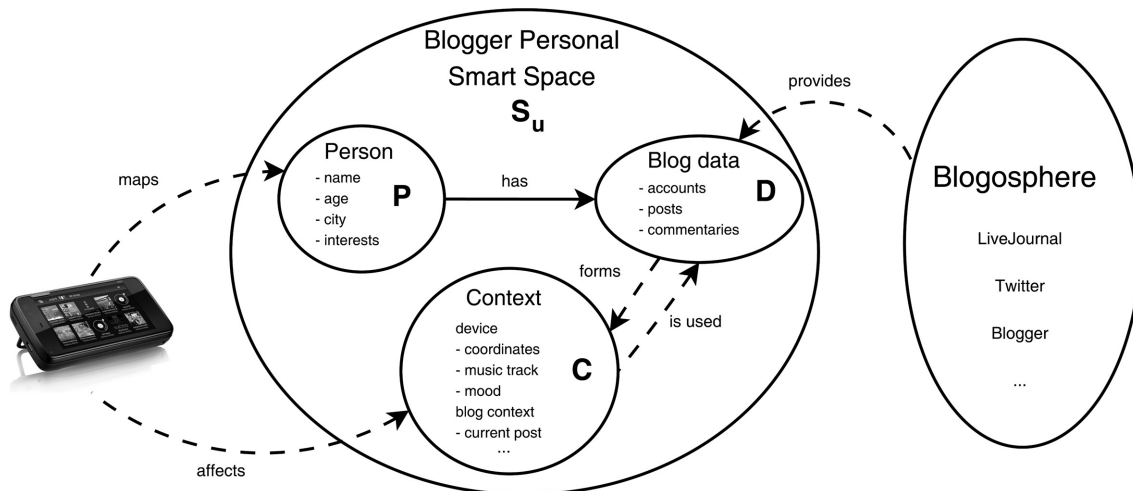


Figure 3. Structure of a blogger personal smart space in SmartScribo.

Context (C_u) represents current and mutable characteristics of a user, e.g., current position (mobile device, GPS), weather, music track, person's mood, the message she/he is reading now. For more discussions on possible context data and their use patterns, please read [2].

Blog data (D_u) include blog accounts and blog discussions. Each account has its own properties (login, password, picture, etc). Any account is associated with a set of discussions. A blog discussion is a tree of messages (root is a post, other nodes are comments). A message has such properties as title, text, tags, date.

Blog data are related with the personal data since the user may have accounts at several blog-services. It aims at multi-blogging when the same personal space is used for all blogs the users currently needs. SmartScribo becomes responsible for their proper association among multiple blog services on behalf of the user.

Access to S is session-based with explicit join and leave operations. The user u can choose where S is physically located (i.e., SIB selection). Also, u can select personal data to share in S , including the access function ϕ_u that limits access of other users to the content $\phi_u(I_u) \subseteq I_u$.

Based on knowledge available in S , other SmartScribo agents make personalized decisions. Blog processors understand which blogs u are interesting at the moment and apply to corresponding blog services. Blog mediators analyze the current needs of u and provide corresponding recommendations.

4. Spaces Composition for Collaboration of Bloggers

The blogosphere defines friendship relations between bloggers, e.g., a blogger at LiveJournal has easier access to blogs of her/his friends. SmartScribo reflects such relations composing personal spaces into interactive structures.

Space compositions provide information sharing between the spaces and can be used, e.g., for reading blogs of friends. We consider compositions that can be constructed with basic space operations [11] and applicable for semantic multi-blogging.

Basically, a blogger space keeps blog discussions owned by the user. Additionally, the space

can represent other blogs if she/he is interested in reading them.

A straightforward solution is duplication. Consider an example in Figure 4. If Alice is interested in discussion B1 from Bob's blog, then her blog processor reads B1 from the blog service and makes a copy B1'' into Alice's personal space. If Alice makes comments to B1, then her blog processor publishes them on the blog service. Bob has his own copy B1' in his smart space if he is a SmartScribo user.

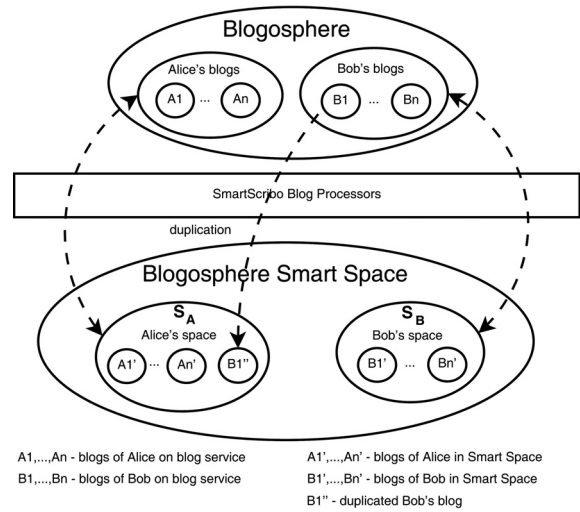


Figure 4. Bob's blog duplication in Alice's personal smart space.

Let Alice's personal space be

$$S_A = (I_A, O); I_A = [P_A \cup C_A \cup D_A]_O.$$

First, Bob's virtual personal space is constructed:

$$S_B^{(A)} = (I_B, O); I_B = [P_B \cup C_B \cup D_B]_O,$$

where data retrieved from blog services are stored, without access to actual Bob's space in S . Typically $P_B = \emptyset$ and $C_B = \emptyset$ as the blog service does not provide access to personal information and context. Secondly, Bob's blog data are added to S_A with *merge* operation [11]:

$$\text{merge}(S_A, S_B^{(A)}, O) = (I_A \cup D_B, O) \rightarrow S_A,$$

where ontology O agrees on merging friend's blog data with user's blog data.

The duplication is appropriate when another blogger is not present in the same blogosphere

space. It is important that it makes SmartScribo independent of the means by which other bloggers access the blogosphere. Moreover, the same blogger can act as SmartScribo or non-SmartScribo user.

If several users are presented in the same blogosphere space, then their spaces can be composed using data sharing between them. Consider an example in Figure 5. If Alice is interested in discussion B1 from Bob’s blog, then Bob shares a part of his data which is consumed by Alice’s space.

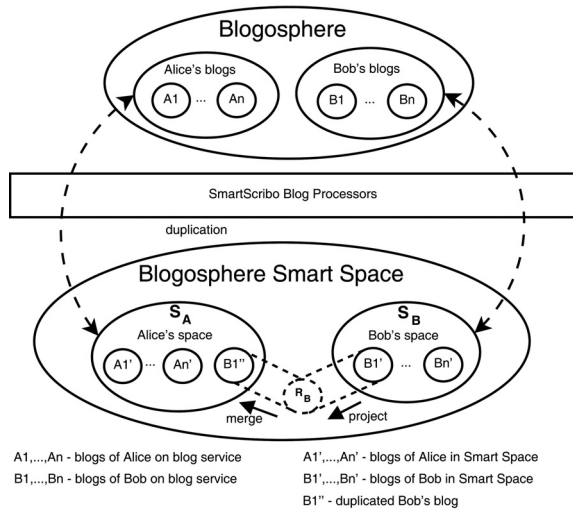


Figure 5. Ontological composition of Alice and Bob’s spaces.

This composition method uses *project* operation [11]. Let S_A and S_B be personal spaces of Alice and Bob. First, the virtual space

$$R_B = \text{project}(S_B, \phi_B, O) = (\phi_B(I_B), O)$$

is formed, where ϕ_B is Bob’s access function. Secondly S_A is updated using the merge operation:

$$\text{merge}(S_A, R_B, O) = (I_A \cup I_{R_B}, O) \rightarrow S_A.$$

The composition uses social relations between personal spaces (e.g., friendship) and defines interest-based relations between a blogger and blogs. It provides a base for semantic blogging where methods of the Semantic Web and social network analysis are applicable, e.g., such as in [9, 3]. For instance, a temporal group formation can be identified in the space to recommend popular blogs within this group to others.

5. Blog Notifications for Synchronization

Activity of SmartScribo agents is synchronized by sharing specific knowledge in the blogosphere smart space. The pub/sub system abstraction is used [6], which is implemented in Smart-M3 as the subscription operation.

The basic synchronization is between a blog client and a blog processor. If the client updates blog data in the blogosphere space, then the processor reflects the change at the blog service. If a blog service has discussions interesting to the user, then an appropriate blog processor shares them in the space. In the straightforward solution, KPs subscribe to blog data in the blogosphere space, leading to a large number of subscriptions and hence to high inefficiency.

We developed a more efficient mechanism using an ontological notification model. A notification initiates KPs to execute actions with blog data (request notification) and to share the result (response notification). For instance, a blog processor tracks notifications from its clients to read/write blog messages from/to the blog service, see Figure 6.

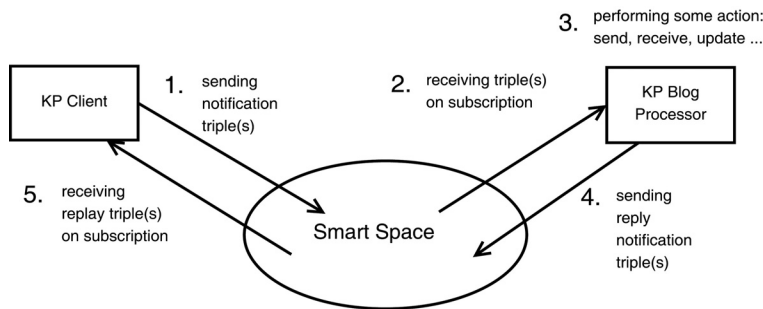


Figure 6. Basic synchronization between a blog client and its blog processor.

A notification is represented as an RDF triple or a set of them. The latter case is for a notification with several parameters and they are represented as a chain of triples. Those KPs that are interested in actions from some other KPs publish notifications. Depending on its role and needs each KP subscribes to the notifications. Whenever a notification is changed, the KP performs the action. If the action returns a result, then it is shared in the space.

Notifications form a separate space (N, O) in S , which is ontologically linked with personal spaces $\{S_u\}$ of all users u . A request notification triple has the object that is a URI of an ontological individual from D_u (e.g., account or post). A response notification triple has the object that is a string with the latest result the operation has returned.

Blog notifications are divided into the following groups: accounts, posts and comments. An account notification requests a blog processor to perform session-based operations using account login and password, e.g., the KP logs in the service on behalf of the blog client. The account notification triple is “ServiceType-Operation-AccountInfo”, where the subject is a blog service (LiveJournal, Twitter, WordPress, etc.), the predicate is a required operation (login, refresh, etc.) and the object refers to an ontology individual with account information. Post and comment notifications are similar (send, receive, update and delete operations).

Different KPs can use the same notification. For example, a client publishes a new blog message in the blogosphere space. The event is recognized by several blog processors, each redirects the message to its own blog service. In addition, a blog mediator tracks this activity, e.g., for accounting blog popularity.

The notifications in the examples above are reactive: an explicit action by a user leads to appropriate actions by blog processors and mediators. Our notification model supports the proactive property, e.g., when a blog client publishes notifications invisibly for the user based on the current user context. For example, a blog mediator is notified whenever the user is starting to read a post with a new topic; the mediator activates searching thematically similar blogs and the recommendations are delivered to the user.

6. Integration with Other Smart Spaces Applications

In the context of this paper, integration means that one application can benefit from knowledge from another application. A reference case we have already implemented is blogging in smart conference [10], where SmartScribo is used to support discussions between conference participants, including online chatting for talks presented at the moment.

Smart conference system maintains the visual content available for participants: a current presentation slide from the conference projector and up-to-date session program from the conference whiteboard. The presenter changes the slides directly from her/his mobile device. SCS supports automatic time management functions and some other useful online control functions.

The integration scenario extends the smart conference functionality as follows. There is a conference blog at some blog service where a post is created for each talk. A post starts a discussion (questions, opinions, answers from the participants including the authors). Consequently, SmartScribo can access this blog. The key problem is coordination between the conference space, blog service and blogosphere space.

In general, each smart space application uses its own smart space with its own ontological knowledge representation, though several applications can construct their spaces at the same physical location (i.e., using the same SIB). In this case, knowledge integration from different smart spaces can employ a spaces mediator that synchronizes data of different sources.

Let a SmartScribo user u also participate in smart conference. The latter has smart space $S_{SC} = (I_{SC}, O_{SC})$, which can be physically located at different SIB. To synchronize data between spaces, the mediator tracks for changes of certain subsets: $J \subseteq I$ and $J_{SC} \subseteq I_{SC}$. For any change in J or J_{SC} the mediator updates data in the other space. For tracking changes the mediator uses the notification model.

In terms of SmartScribo, the integration can be implemented using a mediator KP, which typically acts as a blog processor. The KP can

access the smart spaces of all applications involved into integration. For instance, for Smart Conference, it is a conference blog processor that reads conference program from the conference space, creates and maintains the conference blog at the service and make the blog accessible to SmartScribo clients by sharing it in the blogosphere space.

7. Conclusion

The paper presents our experience and an example of proactive and personalized Internet service, which applies the smart space paradigm. We have discussed: (1) the technique for building personal spaces as a part of the application smart space; (2) the method for efficient space composition that allows the application to operate only with a fraction of huge data; (3) the notification model that enables synchronized provision of proactive and personalized services; and (4) the integration scheme that illustrates how one application can benefit from the knowledge produced by another application. The presented solution has been released for testing and use. The SmartScribo code is available for free download at <http://sourceforge.net/projects/smartscribo/>.

8. Acknowledgments

This research is a part of grant KA179 of Karelia ENPI programme, which is co-funded by the European Union, the Russian Federation and the Republic of Finland.

References

- [1] S. BALANDIN, H. WARIS, Key properties in the development of smart spaces. *Proc. of 5th Int'l Conf. Universal Access in Human-Computer Interaction. Part II: Intelligent and Ubiquitous Interaction Environments (UAHCI'09)*, (2009) pp. 3–12. Springer-Verlag.
- [2] M. BALDAUF, S. DUSTDAR, F. ROSENBERG, A survey on context-aware systems. *International Journal Ad Hoc Ubiquitous Computing*, 2(4) (2007), 263–277.
- [3] M. D. CHOUDHURY, H. SUNDARAM, A. JOHN, D. D. SELIGMANN, Extraction, characterization and utility of prototypical communication groups in the blogosphere. *ACM Transactions on Information Systems*, 29(1) (2010), 6:1–6:53.
- [4] D. J. COOK, S. K. DAS How smart are our environments? An updated look at the state of the art. *Pervasive and Mobile Computing*, 3(2) (2007), 53–73.
- [5] E. G. DA SILVA, L. F. PIRES, M. VAN SINDEREN, Towards runtime discovery, selection and composition of semantic services. *Comput. Commun.*, 34(2) (2011), 159–168.
- [6] P. T. EUGSTER, P. A. FELBER, R. GUERRAOU, A.-M. KERMARREC, The many faces of publish/subscribe. *ACM Comput. Surv.*, 35 (2003), 114–131.
- [7] THE FRIEND OF A FRIEND (FOAF) PROJECT. [http://www.foaf-project.org/\[01/16/2012\]](http://www.foaf-project.org/[01/16/2012])
- [8] J. HONKOLA, H. LAINE, R. BROWN, O. TYRKKO, Smart-M3 information sharing platform. *Proc. of IEEE Symp. Computers and Communications (ISCC'10). IEEE Comp. Soc.*, Jun. 2010 pp. 1041–1046.
- [9] D. R. KARGER, D. QUAN, What would it mean to blog on the semantic web? *Web Semant.*, 3(2–3) (2005), 147–157.
- [10] D. KORZUN, I. GALOV, A. KASHEVNIK, N. SHILOV, K. KRINKIN, Y. KOROLEV, Integration of Smart-M3 applications: blogging in smart conference. *Proc. of 11th Int'l Conf. Smart spaces (ruSMART'11) and of 4th Int'l Conf. Next generation wired/wireless networking (NEW2AN'11)*. (Aug. 2011), pp. 51–62, Springer-Verlag.
- [11] I. OLIVER, S. BOLDYREV, Operations on spaces of information. *Proc. of IEEE Int'l Conf. Semantic Computing (ICSC'09). IEEE Comp. Soc.*, (Sep. 2009) pp. 267–274.
- [12] D. I. TAPIA, A. ABRAHAM, J. M. CORCHADO, R. S. ALONSO, Agents and ambient intelligence case studies. *J. Ambient Intelligence and Humanized Computing*, 1(2) (2010), 85–93.
- [13] D. V. ZAICEVA, I. V. GALOV, A. SANNIKOV, A. MEZHENIN, D. G. KORZUN, Mobile Multi-Blogging in Smart-M3: Architecture and Scenarios. *Acta Universitatis Lappeenrantaensis*, 440 (2011), 15–22.

Received: June, 2012
Accepted: August, 2012

Contact addresses:

Dmitry G. Korzun
 Department of Computer Science
 Petrozavodsk State University – PetrSU
 33, Lenin Ave.
 185910 Petrozavodsk
 Republic of Karelia, Russia
 e-mail: dkorzun@cs.karelia.ru

or

Helsinki Institute for Information Technology - HIIT
 Aalto University
 PO Box 19800
 00076 Aalto
 Finland

Ivan V. Galov
 Department of Computer Science
 Petrozavodsk State University – PetrSU
 33, Lenin Ave.
 185910 Petrozavodsk
 Republic of Karelia, Russia
 e-mail: galov@cs.karelia.ru

Sergey I. Balandin
 FRUCT Oy
 Helsinki
 Finland
 e-mail: sergey.balandin@fruct.org

DMITRY KORZUN received his B.Sc. (1997) and M.Sc. (1999) degrees in applied mathematics and computer science from the Petrozavodsk State University (Russia). He received a Ph.D. degree in physics and mathematics from the St.-Petersburg State University (Russia) in 2002. He is an Associate Professor at the Department of Computer Science of Petrozavodsk State University PetrSU, Russia (since 2003) and a part-time Research Scientist at the Helsinki Institute for Information Technology HIIT, Aalto University, Finland (since 2005). Dmitry Korzun serves on TPC and editorial boards of a number of international conferences and journals. His research interests include analysis and evaluation of distributed systems, discrete modeling, ubiquitous computing in smart spaces, Internet of Things, software engineering, algorithm design and complexity, linear Diophantine analysis and its applications, theory of formal languages and parsing. His educational activity started in 1997 at the Faculty of Mathematics of PetrSU. Since that time he has taught more than 20 study courses on hot topics in computer science, applied mathematics, information and communication technology. He is an author and co-author of about 100 research and educational publications. His recent book has been published by Springer (Structured Peer-to-Peer Systems: Fundamentals of Hierarchical Organization, Routing, Scaling, and Security. ISBN 978-1-4614-5482-3, October 2012).

IVAN GALOV received his B.Sc. degree in information systems and technologies from the Petrozavodsk State University – PetrSU (Russia) in 2011. Now he is an MSc student and programmer at the Department of Computer Science of Petrozavodsk State University. His research interests include smart spaces, Internet of Things, ontology modeling and software engineering. His educational activity started in 2007 at the Faculty of Mathematics of PetrSU. He is an author and co-author of more than 10 research publications.

SERGEY BALANDIN is a President of Open Innovations Association FRUCT and Adjunct Professor in the Department of Communications Engineering at the Tampere University of Technology, Finland. He received the M.Sc. degree in computer science from St.-Petersburg Electro-Technical University “LETT” (Russia) and M.Sc. degree in telecommunications from Lappeenranta University of Technology (Finland). In 2003, Sergey graduated from PhD School of Nokia Research Center and St.-Petersburg Electro-Technical University “LETT” and got his PhD degree in telecommunications and control theory. From 1999 till 2011, Sergey Balandin worked for Nokia Research Center. His last position was Principal Scientist of Ubiquities Architectures team and head of University Cooperation Program in Russian and CIS. Sergey Balandin has authored or co-authored over 70 papers in the field of smart spaces and advanced wired and wireless communications. He co-edited several proceedings books published in LNCS, Springer. Sergey Balandin serves on TPC and steering committees of a number of international conferences and workshops. His current research interests include various aspects of smart spaces solutions, network and services performance evaluation, and Internet of Things.
