Deployment of Smart Spaces in the Internet of Things: Overview of Design Challenges*

Dmitry G. Korzun[1,2], Sergey I. Balandin[3], and Andrei V. Gurtov[1]

[1] Helsinki Institute for Information Technology (HIIT), Aalto University
PO Box 19800, 00076 Aalto, Finland
{dkorzun, gurtov}@hiit.fi

[2] Department of Computer Science, Petrozavodsk State University (PetrSU)
33, Lenin Ave., Petrozavodsk, 185910, Russia
dkorzun@cs.karelia.ru

[3] FRUCT Oy, Helsinki, Finland
sergey.balandin@fruct.org

Abstract. The smart spaces paradigm and the M3 concept have already showed their potential for constructing advanced service infrastructures. The Internet of Things (IoT) provides the possibility to make any "thing" a user or component of such a service infrastructure. In this paper, we consider the crucial design challenges that smart spaces meet for deploying in IoT: (1) interoperability, (2) information processing, (3) security and privacy. The paper makes a step toward a systematized view on smart spaces as a computing paradigm for IoT applications. We summarize the groundwork from pilot M3 implementations and discuss solutions to cope with the challenges. The considered solutions can be already used in advanced service infrastructures.

Keywords: Interoperability, Semantic web, Security.

1 Introduction

The amount of information and services is growing so fast that users cannot efficiently utilize the existing Internet service infrastructure. Low communication between services results in high fragmentation of information. The huge opportunity is in analysis and efficient use of all information by all applications, involvement of many surrounding physical/digital objects into the service provision chain and enabling proactive delivery of the services.

The smart spaces paradigm aims at constructing advanced service infrastructures that follow the ubiquitous computing vision: smart objects are executed on a variety of digital devices and services are constructed as interaction of agents in information sharing environment [1, 2]—smart space. Its users connect new devices flexibly to the space and consume information from any of the services.

* This joint research is supported by Academy of Finland project SEMOHealth, TiViT IoT SHOK program and ENPI CBC Karelia grant KA-179.

The M3 concept further considers the Multidevice, Multidomain, and Multivendor properties of smart spaces [3, 4], resulting in M3 spaces. Smart-M3 [5–7] implements a pilot open-source interoperability platform of M3 spaces. The runtime information and majority of the underlying mechanisms are visible and manageable via a common knowledge base, which exploits Resource Description Framework (RDF) of the Semantic Web.

In contrast to Giant Global Graph of the Semantic Web, M3 spaces are of local and dynamic nature [3]. This property suits well for the Internet of Things (IoT) with its ubiquitous interconnections of highly heterogeneous networked entities and networks. IoT becomes a feasible internetworking substrate on top of which M3 spaces can be deployed. Autonomous everyday objects, being augmented with sensing, processing, and network capabilities, are transformed into smart objects that understand and react to their environment [8, 9]. It has led recently to revision of application programming techniques and met with new design challenges for development of IoT service infrastructures.

This paper sorts out the following design challenges, which smart space deployment and in particular M3-based service infrastructures meet in IoT.

Interoperability: How to manipulate with information in an open dynamic multi-device environment and to offer services to the users.

Information processing: How to reason over the information and to construct the services, despite of environment heterogeneity, volatility, and ad-hoc nature.

Security and Privacy: How to provide integrity and confidentiality of processed data and communication as well as authentication of services and users.

We expect that these challenges are most crucial on the recent phase of M3 concept realization. Other challenges are their instances to certain extent. That is, seamless device integration is connected to interoperability and security, knowledge exchange between services and understanding of the current situation are related to interoperability and information processing.

This overview continues our work [10] on the M3 concept. We analyze its IoT-related challenges and their impact on service infrastructure development and deployment. The analysis considers the latest achievements from recent pilot implementations of M3 spaces and services, including results from regular discussions on ruSmart and FRUCT conferences. We systematize potential responses to the challenges as well as existing M3-based solutions.

The rest of the paper is organized as follows. Section 2 introduces the smart spaces paradigm, M3 concept, and Smart-M3 platform. Section 3 shows an example of M3 space for illustrating the challenges. Sections 4, 5, and 6 sequentially consider the design challenges of smart spaces deployment in IoT and overview existing solutions and research directions. Section 7 summarizes the paper.

2 M3 Spaces

Smart space is an ecosystem of interacting computational objects on shared knowledge base. The key goal is seamless provision of users with information using the best available resources for all kinds of devices that the users can use in the ecosystem [1,3]. M3 spaces focus further on dynamic mash-up and integration of many users, devices, applications, where domains span from embedded digital equipment and consumer electronics to Web [4,5,10]. The fusion of physical and information worlds is not bound to any device type, device vendor, or application domain. Provision of end-users is localized within the situational environment and users' needs, including personalized and context-aware services.

Smart-M3 platform [6] can be used to deploy an M3 space, providing a space-based communication and synchronization substrate to independent agents—knowledge processors (KPs). They run on devices available in the environment and communicate by inserting information to the space and querying the information in the space. The space is represented by one or more semantic information brokers (SIBs); they maintain a knowledge base—a named search extent of information. Smart-M3 employs term "knowledge": a space keeps habitual data, relations between them, and even such information as computations. Existing SIB software implementations include original Smart-M3 SIB [6], RedSIB [7], OSGi SIB [11], RIBS [12], and ADK [13].

Instant content is stored as an RDF graph, adopting the low-level triple-based approach of the Semantic Web. The RDF model allows easy linking and semantic-level interoperability when there are many content producers and consumers. Each SIB performs RDF triples governance in possible cooperation with other SIBs of the same space. Transactions between SIB and KP follow Smart Space Access Protocol (SSAP), which supports the basic space primitives: join/leave, insert/update/remove, query, (un)subscribe.

M3 space application is an ad-hoc assembly of KPs implementing collaboratively a service scenario to meet users' goal. The high-level view is illustrated in Fig. 1. Scenario steps emerge from actions taken by the KPs and observable in the application M3 space. Access to global knowledge is possible via a gateway KP to the external world. A scenario can be composed from multiple applica-

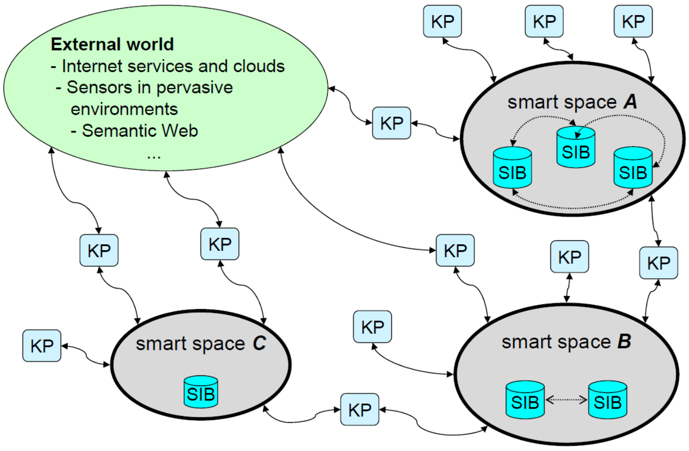

Fig. 1. Each space is maintained by own SIBs to host applications. An application is formed by KPs that publish and query shared content. Some KPs are wrappers for external services; some are used for knowledge exchange between spaces.

tions. The key point is the loose coupling between the participating KPs. The impact of each KP to others is limited by the knowledge the KP provides into the space. Within the same application its space can conform to a common ontology [14,15], though it can be modular, multi-domain, composed from multiple ontologies. Each KP applies own "sub-ontology" to interpret the accessible part of the shared content. The application is not fixed since its KPs may join and leave the space. Several demo pilots have been already developed and showed the feasibility of the M3 concept, see [2, 5, 10, 16–18] and references therein.

Consider the M3 ontology-driven computing formalism based on the generic smart space model [19]: a space is a knowledge base $S = (n, I, \rho)$, where n is its name, I is information content, and ρ is a rule set to deduce knowledge. Content I is represented as an RDF graph. When ρ is OWL ontology O then I is further structured with classes and properties from O. Deduction in S can be performed on the ontology instance graph using techniques of Semantic Web [3, 19, 20]. Such a graph is formed by individuals (nodes) that are interlinked with object properties (links) and have data properties (attributes). That is, we can treat an application M3 space as $S = (I, O)$, where I is an RDF graph and O is an OWL ontology. A portion of knowledge $x \in S$ is an ontology instance graph that is a part of the deductive closure calculated from I according to O. We also refer to S as to the space unique name.

3 Explanatory Example: Smart Room

Let us consider an example M3 space—Smart Room [18] to explain the reasons and importance of challenges of smart spaces deployment in an IoT environment. Smart room system scope is shown in Fig. 2.

Communication in a smart room uses a wireless local area network (WLAN) attached to the Internet. Participants are chairman, active speaker (in turn relay manner), and spectators (including inactive speakers). Two public screens are available: (1) Agenda shows the event timetable and (2) Presentation shows material that each speaker presents.

The participants access services in the smart room using personal mobile computers (e.g., smartphones, tablets, laptops). The room is equipped with sen-

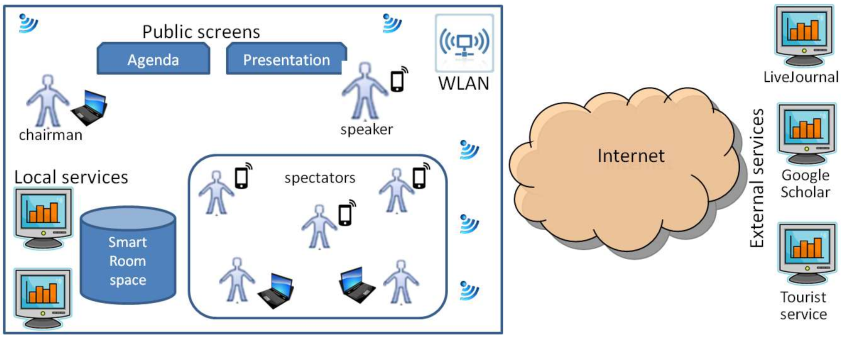

Fig. 2. Equipment and service environment for participants in Smart Room

sor devices that sense the physical parameters of the environment and participant activity. All knowledge is collected, organized, shared, and searched in a common smart space. Local services run on local computers or nearby servers. The system accesses the external world for appropriate Internet services. Service outcome is visible online on the public screens or personalized on mobile clients.

The heterogeneity of participating devices and information sources immediately faces with the interoperability challenge. Smart room service set is formed from multiple sources of heterogeneous information and requires intensive processing, including service discovery and the provision to a particular participant or a group of them. Personal information is essential for smart room operation, but it must support rigorous security and privacy defense mechanisms.

Examples of particular problems are the following. Integration of joining devices (e.g., personal devices) is seamless. Service management is adaptive, e.g., when some services become temporarily unavailable. Knowledge exchange is supported: one service utilizes knowledge deduced by another service, e.g., discussion of participants in the blog leads to updates in the agenda.

4 Interoperability

Information available on some devices may be interesting to other devices of the same environment. Furthermore, some devices should communicate with the external world. Currently, the standards for interoperability have been mostly created for single domains or are controlled by a single company. Such domain specific standards pose considerable challenges for IoT devices. The traditional standardization approach cannot achieve the basic IoT property: a device can interoperate with whatever devices accessible at the given time.

The smart spaces concept makes clear separation between device, service, and information level interoperability [5,21]. Device interoperability covers technologies for devices to discover and network with each other. Service interoperability covers technologies for space participants to discover services and use of them. Information interoperability covers technologies and processes for making information available without a need to know interfacing methods of the entity creating or consuming the information.

Application developer uses KP Interface (KPI) for programming KP logic and its interaction with the space by SSAP primitives [6,22]. The M3 concept requires that SIB supports a number of solutions for network connectivity, yielding multivendor device interoperability. For Internet communication, SIB supports HTTP and plain TCP/IP. Short-range wireless communications of mobile devices can use such connectivity solutions as Bluetooth or 6LoWPAN. Network on Terminal Architecture (NoTA) provides a possible solution for embedded devices. Reliable communication on top of IPv4 and IPv6 uses Host Identity Protocol [23], which supports mobility and multi-homing, see Sect. 6. Application code developer selects a connectivity mechanism for a device family.

The device heterogeneity introduces additional difficulty for the KP development. If the hosting device is a computer (i.e., relatively powerful OS and ability to run non-trivial programs), then KP can run directly on the device.

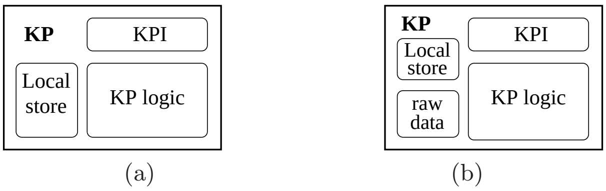

Fig. 3. Typical KP architecture: (a) KP running on a computer, (b) KP serving as a gateway for low-capacity devices.

The computational resources have to be allocated for KPI (SSAP operations, XML processing, networking), KP logic (written by the developer), and local store (knowledge that KP directly processes), see Fig. 3 (a).

Techniques for efficient KP programming for mid-capacity devices (laptops, smartphones, tablets, etc.) and certain embedded devices (with embedded Linux, Contiki, etc.) exist [14]. If a device is very primitive (even no operating system, like in a sensor), then hosting a KP on the device is unreasonable or even impossible. In this case, such a device can be attached to the space via a dedicated computer running a gateway KP, see Fig. 3 (b). The KP has to transform data between the given data format of satellite device and the ontological representation in the KP local store. For instance, this approach is used for constructing personal smart space in healthcare applicationss [24].

Many of primitive devices are pure data producers in the M3 space. Thus the required processing at the KP side is small due to the machine-oriented property of RDF. It includes transformation of the raw data into triples and construction of simple SSAP packets in XML or binary format [12,22]. The above processing can be implemented on the hardware level.

Information sharing in M3 spaces is based on the same mechanisms as in the Semantic Web, thus allowing multidomain applications, where the RDF representation allows easy exchange and linkage of data between different ontologies. It makes cross-domain interoperability straightforward [6]. Application domains are localized, limiting the search extent and ontology governance. That is, for each application its space $S = (I, O)$ is relatively small, allowing computationally reasonable knowledge maintenance at SIB and moderate performance expenses at KP. The interoperability is due to the locally agreed unification of semantics when accessing the same part of the space content I. That is, the space-wide ontology O is a virtual application-level component.

The space content I is organized into an RDF graph. Although explicit use of a specific ontology is not demanded, additional semantics are provided by an ontology O, usually defined in OWL. For example, a group of KPs can agree an aligned ontology for interpretation of a certain part of the space. The consistency of stored information is not guaranteed; KPs are free to interpret information in whatever way they want. This RDF-based low-level model requires KP code to operate with triples following the SSAP operations directly; the triples are basic exchange elements in communication with the M3 space.

For development efficiency, the high-level ontology-driven KP programming is supported, e.g., SmartSlog SDK [14]. The approach is based on an ontology library, which is automatically generated mapping the ontology to code in a given programming language. The KP logic then is written using high-level ontology entities (classes, relations, individuals). They are implemented with predefined data structures and methods. It essentially simplifies the KP code; the developer has the programming language-like tools to manipulate with the concepts defined in the ontology. The number of domain elements is reduced since an ontology entity consists of many triples. The library API is generic: its syntax does not depend on a particular ontology, ontology-related names do not appear in names of API methods, and ontology entities are used only as arguments.

Notably that ontology library is less machine-dependent than low-level KPI. The same high-level KP code is suitable for different devices since the ontology library can wraps the appropriate KPI.

5 Information Processing

The SIB side of M3 space provides mechanisms for knowledge discovery and first-order logic reasoning. Each space may contain its own set of reasoning capabilities. The most important mechanisms are semantic queries and subscription.

To find appropriate knowledge in S the KP constructs a query using semantic query languages as SPARQL. The SIB resolves the query [7] and returns an ontology instance graph $x \in S$. The KP interprets the result locally and then can insert new knowledge to S or update some previous instances. The appropriate deduction (e.g., deductive closure) is performed by SIB dynamically—at query-time (also at insert-time in some spaces).

A subscription operation is a special case of query—a persistent query, realizing the publish/subscribe communication model in smart spaces [25]. Changes in the space content trigger actions from participating KPs. Subscription is used (i) for synchronizing KP's local knowledge storage with the shared space, as well as (ii) for receiving notifications about recent changes. The latter is a way for a KP to detect events happening in the system.

The RDF-based semi-structured knowledge representation with no strict ontology conformance shifts the responsibility of knowledge interpretation and truth maintenance to the agents. Each KP u manages a non-exclusive part I_u of knowledge and applies own expertise for reasoning over I_u. A KP u uses own ontology O_u as an assistance tool that helps to achieve a common understanding with other KPs, see Fig. 4. Each KP publishing its shared knowledge provides meta-information to indicate intention for interpretation. Thus, an application is aware of the unification of semantics, which can be done in a localized manner (between a group of KPs) and even runtime. It may result in information inconsistency in the space and misinterpretation on the reader side. The supporting mechanisms to deal with this problem are under development [14, 26, 27].

The M3 concept supports multi-space applications when a KP needs the information from several spaces (Fig. 1 in Sect. 2). It provides an opportunity for applications integration in an ad-hoc manner. Notably that the coupling between the participating KPs is loose and KP granularity can be extremely fine (e.g., a KP can implement a function outside of the hosting UI concept).

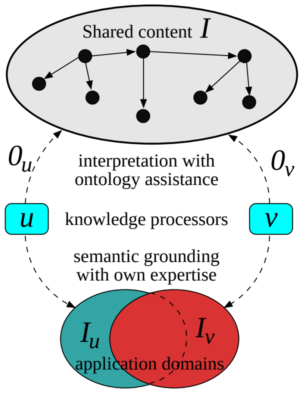

Fig. 4. KPs u and v make own interpretations of the content. The ontologies and semantic grounding are agreed to ensure a common interpretation and aligned domains.

When an application needs a service from another Smart-M3 application, a KP mediator can be used to connect these spaces [16]. The mediator should properly interpret corresponding knowledge in the source and target spaces, construct a mapping between them, and execute the exchange. Ontology accompanied with logic programming rules can define constraints to which exchanged instances satisfy as well as specify mappings between the spaces.

6 Security

The security challenge includes traditional issues of open distributed systems, such as key exchange and resource restrictions, and specific problems caused by the dynamicity and heterogeneity of smart spaces [28]. We classify smart space security components onto (a) share level, (b) space access control, and (c) communication. Let u and v be KPs in space S.

Security of the share level is based on a sharing function $\sigma_u(S) \subset I$ that defines which locally available knowledge to publish in S for sharing with others. Each KP makes own decisions on its share level, keeping essentially private knowledge at the local storage only. It does not prevent u to combine private and shared knowledge in local reasoning.

In the space access control, an access function ϕ_u limits other KPs in access u's shared content; $\phi_u \subset I$ is the knowledge that u allows for v. Hence u and v collaborate in the content $\phi_u \cup \phi_v$ (Fig. 5). Since SIB enforces access control over brokered information, application-specific policies need additional support.

Access control benefits from meta-information published in the space. Exclusive access to the content can be on RDF level. The method of [29] allows restricting the access for an arbitrary set of triples. Meta-information is additional triples that specify which data are protected and which KP is their owner. The method may be embedded in middleware data access primitives of a standard KPI, so becoming hidden to the KP developer. Although the method allows extension for more security attributes beside synchronization, any KP is able to see what has been protected in the space.

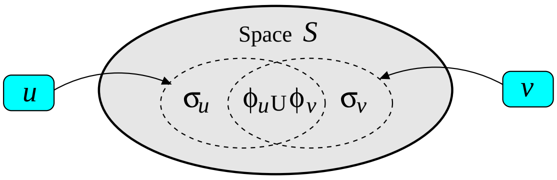

Fig. 5. KPs u and v provide restricted content to share in S. They collaborate accessing only allowed parts of content of each other.

IoT applications need context-dependent and fine-grained access control [28, 30]. Smart space access control policies define which KPs are allowed to access which objects. Security level of joined devices is measured. An access control ontology allows representing meta-information about the context and granularity. SIB utilizes this information to authorize the access to the space content. The approach enables devices to share knowledge with the same security level even when these devices do not have interoperable security protocols for direct confidential communication. Note that combining security policy rules with reasoning allows exploiting further the advantages of logic programming and description logic [29, 31]. Unfortunately, many reasoning problems in general form require exponential time in the worst case.

Accessing the space is session-based by join/leave operations [32]. It forms a base for mechanisms of access control and secure communication. For instance, KP identity and cryptographic keys can be implemented with Host Identity Protocol (HIP) [23], which is standardized by IETF. The HIP exchange authenticates a KP-to-SIB communication session based on robust identities. They can be used for access control to different parts of the space, implementing the access function ϕ. All transferred data are encrypted, so providing confidentiality and message integrity in communication with SIB. The same approach was discussed in [30] for Transport Layer Security (TLS) protocol. However, TLS does not support mobility and multi-homing as well as causes significant overhead.

Many IoT devices are of low capacity (memory, CPU, battery, etc.), and they cannot use the full scale of security capabilities that the basic HIP or other Internet protocols provide [9]. A HIP-based extension for secure transfer of private data in M3 spaces is proposed in [24] for healthcare applications with wearable and implantable medical devices. The proposal employs HIP Diet Exchange (DEX) to establish secure associations between KP and SIB. HIP DEX requires rather limited computation capabilities from the devices since it uses elliptic curve cryptography to distribute the shared secret. Although HIP DEX is designed for resource-restricted devices it still provides possibility to control performance level by adjusting cryptographic computation difficulty.

7 Conclusion

This paper considered the smart spaces paradigm and its potential for service infrastructure development in IoT environment. The discussion focused on the

Table 1. Summary of challenges and their solutions

Challenge	Provided solutions and feasible directions
Interoperability: device, service, information	Many network protocols. RDF-based operation of SSAP. Multiplatform KPIs and reusable code. Ontology libraries and code generation. Development tools for mid- and low- capacity devices.
Information processing	SPARQL queries and first-order reasoning. Subscription and proactive services. Ontology-driven development and runtime mechanisms. Multi-space operation and mediator-based synchronization.
Security and Privacy	RDF-based knowledge access control and mutual exclusion. HIP-based network communication. Ontology-based control policies and context-aware security.

design challenges of smart spaces deployment: (1) interoperability, (2) information processing and (3) security and privacy. Table 1 lists the corresponding solutions. Although full-valued solutions are still under development, the presented summary shows the overall feasibility and applicability of the smart spaces computing paradigm for IoT settings.

Knowledge processors running on IoT devices and cooperating in various service scenarios are loosely coupled. The shared content conforms an ontological knowledge representation, supporting also localized agreements and personalization. These properties provide a base to tackle the interoperability challenge.

The semantic reasoning mechanisms and their distributed nature support effective processing within huge multi-source information collections. The M3 concept states localized ad-hoc spaces and integrates the Semantic Web with other information on surrounding electronic devices. This groundwork feeds and catalyzes solutions to the information processing challenge.

Progress in Internet security protocols provides promising solutions for confidential communications and authentication of the participants with strong cryptographic identities. The computation overhead can be made low, and even low-capacity devices are involved into the service infrastructure. Additionally, advanced semantic models equipped with logic programming techniques support fine-grained context-dependent access control to the shared content.

References

1. Cook, D.J., Das, S.K.: How smart are our environments? an updated look at the state of the art. Pervasive and Mobile Computing **3**(2) (2007) 53–73
2. Smirnov, A., Kashnevik, A., Shilov, N., Oliver, I., Balandin, S., Boldyrev, S.: 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) 42–51
3. Oliver, I.: Information spaces as a basis for personalising the semantic web. In: Proc. 11th Int'l Conf. Enterprise Information Systems (ICEIS 2009). (May 2009) 179–184
4. Balandin, S., Waris, H.: Key properties in the development of smart spaces. In: Proc. 5th Int'l Conf. Universal Access in Human-Computer Interaction. Part II:

Intelligent and Ubiquitous Interaction Environments (UAHCI '09), Springer-Verlag (2009) 3–12

5. Liuha, P., Lappeteläinen, A., Soininen, J.P.: Smart objects for intelligent applications - first results made open. ARTEMIS Magazine (5) (October 2009) 27–29
6. Honkola, J., Laine, H., Brown, R., Tyrkkö, O.: Smart-M3 information sharing platform. In: Proc. IEEE Symp. Computers and Communications (ISCC'10), IEEE Computer Society (June 2010) 1041–1046
7. Morandi, F., Roffia, L., D'Elia, A., Vergari, F., Cinotti, T.S.: RedSib: a Smart-M3 semantic information broker implementation. In Balandin, S., Ovchinnikov, A., eds.: Proc. 12th Conf. of Open Innovations Association FRUCT and Seminar on e-Tourism, SUAI (November 2012) 86–98
8. Kortuem, G., Kawsar, F., Sundramoorthy, V., Fitton, D.: Smart objects as building blocks for the internet of things. IEEE Internet Computing **14**(1) (January 2010) 44–51
9. Bonetto, R., Bui, N., Lakkundi, V., Olivereau, A., Serbanati, A., Rossi, M.: Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples. In: Proc. IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2012), IEEE Computer Society (2012) 1–7
10. Korzun, D.G., Balandin, S.I., Luukkala, V., Liuha, P., Gurtov, A.V.: 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). Volume 4., Moscow: Physmathlit (September 2011) 64–71
11. Manzaroli, D., Roffia, L., Cinotti, T.S., Ovaska, E., Azzoni, P., Nannini, V., Mattarozzi, S.: Smart-M3 and OSGi: The interoperability platform. In: Proc. IEEE Symp. Computers and Communications (ISCC'10), IEEE Computer Society (2010) 1053–1058
12. Suomalainen, J., Hyttinen, P., Tarvainen, P.: Secure information sharing between heterogeneous embedded devices. In: Proc. 4th European Conf. Software Architecture (ECSA '10): Companion Volume, ACM (2010) 205–212
13. Gómez-Pimpollo, J.F., Otaolea, R.: Smart objects for intelligent applications – ADK. In: Proc. 2010 IEEE Symp. Visual Languages and Human-Centric Computing (VL/HCC). (Sep 2010) 267–268
14. Korzun, D.G., Lomov, A.A., Vanag, P.I., Honkola, J., Balandin, S.I.: Multilingual ontology library generator for Smart-M3 information sharing platform. International Journal on Advances in Intelligent Systems **4**(3&4) (2011) 68–81
15. Lomov, A.A.: Ontology-based KP development for Smart-M3 applications. In Balandin, S., Trifonova, U., eds.: Proc. 13th Conf. of Open Innovations Association FRUCT and 2nd Seminar on e-Tourism for Karelia and Oulu Region, SUAI (April 2013) 94–10
16. Korolev, Y., Korzun, D., Galov, I.: Smart space applications integration: A mediation formalism and design for Smart-M3. In: Proc. 12th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN 2012) and 5th Conf. Internet of Things and Smart Spaces (ruSMART 2012). LNCS 7469, Springer-Verlag (August 2012) 128–139
17. Kiljander, J., Ylisaukko-oja, A., Takalo-Mattila, J., Eteläperä, M., Soininen, J.P.: Enabling semantic technology empowered smart spaces. Journal of Computer Networks and Communications **2012** (2012)
18. Galov, I., Korzun, D.: Smart room service set at Petrozavodsk State University: Initial state. In Balandin, S., Ovchinnikov, A., eds.: Proc. 12th Conf. of Open Innovations Association FRUCT and Seminar on e-Tourism, SUAI (November 2012) 239–240

19. Oliver, I., Boldyrev, S.: Operations on spaces of information. In: Proc. IEEE Int'l Conf. Semantic Computing (ICSC '09), IEEE Computer Society (September 2009) 267–274
20. Gutierrez, C., Hurtado, C.A., Mendelzon, A.O., Pérez, J.: Foundations of semantic web databases. J. Comput. Syst. Sci. **77**(3) (May 2011) 520–541
21. Ovaska, E., Cinotti, T.S., Toninelli, A.: The design principles and practices of interoperable smart spaces. In Liu, X., Li, Y., eds.: Advanced Design Approaches to Emerging Software Systems: Principles, Methodology and Tools. IGI Global (2011) 18–47
22. Kiljander, J., Morandi, F., Soininen, J.P.: Knowledge sharing protocol for smart spaces. International Journal of Advanced Computer Science and Applications (IJACSA) **3** (2012) 100–110
23. Gurtov, A., Komu, M., Moskowitz, R.: Host Identity Protocol (HIP): Identifier/locator split for host mobility and multihoming. Internet Protocol Journal **12**(1) (March 2009) 27–32
24. Gurtov, A., Nikolaevskiy, I., Lukyanenko, A.: Using HIP DEX for key management and access control in smart objects. In: Proc. of Workshop on Smart Object Security. (March 2012) Position paper.
25. Lomov, A.A., Korzun, D.G.: Subscription operation in Smart-M3. In Balandin, S., Ovchinnikov, A., eds.: Proc. 10th Conf. of Open Innovations Association FRUCT and 2nd Finnish–Russian Mobile Linux Summit, SUAI (November 2011) 83–94
26. Smirnov, A., Kashevnik, A., Shilov, N., Balandin, S., Oliver, I., Boldyrev, S.: On-the-fly ontology matching in smart spaces: a multi-model approach. In: Proc. 3rd Conf. Smart Spaces (ruSMART'10) and 10th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN'10), Springer-Verlag (2010) 72–83
27. Janhunen, T., Luukkala, V.: Meta programming with answer sets for smart spaces. In: Proc. 6th Int'l Conf. Web Reasoning and Rule Systems (RR 2012), Springer (2012) 106–121
28. Evesti, A., Suomalainen, J., Ovaska, E.: Architecture and knowledge-driven self-adaptive security in smart space. Computers **2**(1) (2013) 34–66
29. D'Elia, A., Manzaroli, D., Honkola, J., Cinotti, T.S.: Access control at triple level: Specification and enforcement of a simple RDF model to support concurrent applications in smart environments. In: Proc. 11th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN'11) and 4th Conf. Smart Spaces (ruSMART'11), Springer-Verlag (2011) 63–74
30. Suomalainen, J., Hyttinen, P.: Security solutions for smart spaces. In: Proc. 2011 IEEE/IPSJ Int'l Symposium on Applications and the Internet (SAINT '11), Washington, DC, USA, IEEE Computer Society (2011) 297–302
31. Aziz, R.A., Janhunen, T., Luukkala, V.: Distributed deadlock handling for resource allocation in smart spaces. In: Proc. 11th Int'l Conf. Next Generation Wired/Wireless Networking (NEW2AN'11) and 4th Conf. Smart Spaces (ruSMART'11), Springer-Verlag (2011) 87–98
32. Lomov, A.A.: SmartSlog session in Smart-M3. In Balandin, S., Ovchinnikov, A., eds.: Proc. 12th Conf. of Open Innovations Association FRUCT and Seminar on e-Tourism, SUAI (November 2012) 66–71