Role of Mobile OS and LBS Platform in Design of e-Tourism Smart Services

Ekaterina Balandina[1], Sergey Balandin[2,3], Yevgeni Koucheryavy[1], Mark Zaslavskiy[2,3]
[1] Tampere University of Technology, [2] ITMO University, [3] FRUCT Ltd
Tampere, Finland; Saint-Petersburg, Russia; Helsinki, Finland
e-mail: Ekaterina.Dashkova@student.tut.fi; yk@cs.tut.fi; Sergey.Balandin@fruct.org; Mark.Zaslavskiy@fruct.org

Abstract – **The paper discusses opportunities and challenges in development of the current ecosystem of digital services. Special attention is paid to analysis of the role of Location Based Services (LBS) platforms for service ecosystems in the Internet of Things (IoT) era. We study architectures of LBS-enabled smart systems and analyze factors that could enable faster adoption of new service paradigms by the industry. The paper discusses potential roles of the IoT infrastructure for addressing this problem. One of the supporting questions is the role of mobile operational systems in development of a future ecosystem of the services, which we study by reviewing two approaches implemented in two open source mobile operational systems: Sailfish OS and Tizen OS. One of the starting observations was that the "cold start" problem is one of the top factors that block services from successful development. The problem refers to the case when a new service lacks relevant content. We propose to address this problem by providing developers with a toolkit for accessing relevant content available in various open databases. Development of a method for efficient data importing from open databases and content management is one of the practical results of this study. We implemented the proposed method as an extension to the open source LBS platform Geo2Tag. Now, it is available for free use and illustrates really good performance. Results of our study were tested on the most typical use cases of services for tourists and hospitality industry. The practical results of projects are available for use by business and helped us formulate priorities for further research.**

Keywords: LBS; Internet of Things; Geo2Tag; Sailfish OS; import of open data.

I. INTRODUCTION

Nowadays, the service market is in the middle of a major transformation towards smart and proactive services. This transformation is supported by higher availability of powerful mobile devices with broadband network access, large memory and significant processing power, which allow to store and efficiently process large volume of data. Moreover, the development of the Internet of Things (IoT) ecosystem enables modern services to collect most relevant local content and manage the environment around the user. But, despite the fact that the opportunity is there, in practice, we still see only a few examples of services that take this opportunity to practice. So, let us analyze potential reasons for the slow adoption of new technologies and present our solutions that are targeted to improve the situation.

The absolute majority of available services are based on the interaction of a user with mobile devices. The user does not only consume deliverables, but is actively involved in content creating, knowledge management and decision making. However, we evaluate that this approach has reached its limits due to the explosive growth of volume and complexity of data. The services shall autonomously and efficiently search, filter and process the data and so become more intelligent and proactive.

Therefore, the motivation for this study is to take part in the definition of design for the new generation of mobile services ecosystem [1], which, in particular, fulfils the following criteria:

- provides innovative pro-active smart services that improve quality of life, are highly personalized, taking over the most difficult and boring work from the user, helping to save natural resources and wherever possible allow to use prophylaxis instead of curing a damage created by a problem;
- is stable, efficient and scalable to enable broad deployment of Internet of Things paradigm that will connect by two orders of magnitude more devices than the number of humans on the planet.

The development of such ecosystem for smart services demands to find solutions for the following two domains:

- mobile operational system (OS) that is functionally rich, flexible and efficient to provide required low-level support for the smart services;
- availability of a tool to define the accurate context of the user and collect the corresponding contextual information.

In complex ecosystems, such as smart spaces, mobile devices are considered primary as a tool for accessing services [2]. However, development of the smart services ecosystem demands more active role of mobile devices, i.e., they shall become center of the personal smart space and the personal management hub for the surrounding IoT devices [3]. Consequently the mobile devices shall become more powerful and the mobile OS be more efficient, functional-reach and flexible to provide the required support. In the next section of this paper we provide the corresponding analysis and comparison of the available mobile OS ecosystems.

Also, in the last years, we have seen that the role of location-aware services is increasing and the corresponding ecosystem is growing fast and developing in all aspects. Most of mass-market mobile devices have embedded technologies for detecting location. This creates a huge opportunity to develop services and solutions that associate virtual tags to the real physical objects and processing most relevant content based on the geographical context. This domain is called Location Based Services (LBS) and it very natural approach for development of smart services of next generation, which can be used for various use cases, e.g., for the cultural heritage management systems [1][4].

The third section of the paper discusses the potential role of LBS platform for development of smart services. As a result of IoT ecosystem development, we expect significant rise of the demand for LBS support, as IoT solutions need to monitor the geographical position of the things in time and attach them with a set of relevant attributes. This is exactly what LBS platforms provide from the box. For this study we particularly discuss how the LBS platform can address the “cold-start” problem. It is well known problem for newly launched services that do not have critical mass of relevant content. Lack of relevant content makes services less attractive for the users and often lead to the complete fail of a project.

For this study, we took as an example the IoT-enabled open source LBS platform Geo2Tag [5]. The platform already provides most of required service primitives, so importing function could be delivered with reasonable efforts. The additional advantage of the platform is that it provides efficient toolkit for development of services based on the Smart Spaces principles.

As a source of content, we are offering the universal importer from largest databases of open data. Open data is fast growing model of content delivery, where content is generated by volunteers and in projects funded by government and public funding, which makes the generated content available for free use. Already now thus model is strong competitor to the classical schemes of paid access to the content and it is expected that its role will increase in the future [6]. So the third section provides general presentation of our solution and discusses benefits that it delivers to the services. The fourth section presents implementation of the corresponding solution and discusses its performance characteristics. The key findings and results of our study are summarized in the conclusion section, followed by acknowledgments and list of references.

II. Role of Mobile OS

The root of this study is based on observation that the mobile device shall not be any longer seen as a pure service consumption point. Nowadays, personal mobile devices are powerful enough to take role of a manager of IoT environment around the user. Smart spaces technology provides the best infrastructure for it [7]. However, for deployment of personal smart space its core part - Semantic Information Broker (SIB) shall be installed on the mobile device. Such architecture does not contradict to the smart spaces reference model [2], but so far this case has not been studied. The main reason is that previously mobile devices were not powerful enough and most mobile OS could not provide required low-level support and flexibility. Nowadays, technically smartphones are powerful enough, but the main question whether mobile OS could efficiently use the device hardware and provide all basic functional required for proper operation of smart spaces SIB.

We started this part of study by making review of the leading mobile OS. Analysis of iOS and Windows ecosystems discovered that both are not suitable, as do not provide the required access to the low-level interfaces, the provide privacy tools are insufficient and the overall performance is not insufficient. The first results for Android OS were much more promising. One can find a number of publications on using Android devices for smart spaces applications [8]. Moreover, our initial tests show that it is possible to make SIB working on Android devices. However, Android sets to many restrictions on use of the low-level functions. Consequently due to inefficient use of resources we end-up with situation that smartphone cannot provide sufficient processing power for proper management of the personalized smart spaces by SIB even on most powerful Android phones. Moreover, Android architecture is not good for implementation of the proper privacy solution. As a result Android OS was excluded from further consideration.

But, further study of this topic discovered that, nowadays, there are two open source mobile OS that could fulfill our requirements. The first candidate is Tizen OS that is the best of available mobile OS adopted for use in resource restricted devices and IoT environment. For the last two years, Tizen OS is clearly positioned to take the dominant role as operational system for Internet of Things [9]. Nowadays, one can find a lot of IoT devices on Tizen and the system ensures seamless device-to-device connectivity and address needs of the whole Internet of Things environment. Tizen OS is target to remove border between the personal mobile device and surrounding IoT devices, as all devices operate under the same OS and mobile device can be seen just as one of IoT devices. Such flat peer-to-peer architecture is very scalable and well fits to the general IoT scenarios. However, it is not optimal for deployment of smart spaces that are done on principles of client-server architecture. As a consequence Tizen smartphones are not so much optimized for heavy computations and the privacy model does not directly match to the smart spaces needs. Moreover, nowadays there are not many Tizen-enabled smartphones and none of them is powerful enough to run SIB of the personal smart space. So as a conclusion we admit that Tizen OS is very promising mobile OS with great potential for our needs, but at the moment it is not suitable and there are no Tizen smartphones that can be used even for prototyping purposes. We are going to keep eye on further development of Tizen OS ecosystem, but had to drop it for the purpose of this study.

Sailfish OS is the optimized development of principles defined in MeeGo OS [10]. Sailfish OS is focusing on smartphone as a primary type of target device. Nowadays, Sailfish OS is the most efficient and fast mobile OS where the additional key priorities are privacy and usability. The system is based on Linux kernel and provides most of required basic primitives for accessing low-level functional and interfaces. Smartphone under Sailfish OS can provide efficient hosting for SIB and at the same time perform role of a hub when accessing to the various IoT services. The open architecture allows developing and integrating the missing primitives to the system core. These are exactly the features we are looked from a mobile OS. Nowadays, there is more than half a dozen of high-end smartphones on Sailfish OS. So it was possible to find required device for our study.

As a result, we confirmed that modern mobile OS can enable the new organization of smart spaces, where the core of the smart space is hosted by the mobile device. Also, we can conclude that, currently, Sailfish OS provides the best ecosystem for building personalized M3-enabled smart spaces. Within scope of the study, we defined all basic elements of the smart spaces that dynamic personalize the virtual and physical environment based on users’ preferences. Based on results of this study, our next step will be to implement a fully functional IoT management architecture in personal smart space developed on top of the open source Smart-M3 platform [11].

III. Role of Location-based Context Information

The recent progress in volumes of available data storages and speed and availability of telecommunication technologies enable development of the new generation of services. Potentially this service in any place at any time can access any piece of knowledge created by the humankind. But, the volume of knowledge is so huge that it would flood any service. There are multiple technologies to address this need, starting from efficient data search algorithms to advanced methods of data mining and big data management. However, by applying even the best methods to the full set of data we end up in unacceptable delays and low relevancy of the results. The only way to significantly help this situation is by applying efficient filtering of the content. Context is the most natural filtering. Generally we took definition of the context as any information that can be used to characterize the situation of an entity, where an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves [12]. For this study we focus on a subset of the context information that can be derived based on the geographical location of an entity. As it was shown by the previous studies, in majority of practical use cases the location-based context gives the most relevant filtering of the content [13]. Moreover, according to the analytical forecast in the next 3 years the demand for LBS-context driven solutions among developers will increase by at least 22% [14].

As an illustration of the role of the location-based context in smart services, we decided to address well-known problems, typical for current services. We target to propose a solution for the lack-of-content (also known as “cold-start”) problem of new services. To make the story clearer for understanding and results more creditable, we illustrate the demand for such solution, the proposed idea and our implementation, on an example of services for the tourists and hospitality industry.

The recent reviews of tourist expectations done by tourist offices and agencies, as well as general analytic reports for the hospitality industry discovering clear trend that modern tourist is willing to see advanced e-tourism ecosystem that would enable personalization of service with minimal cost overhead. For example, one can easily see success of the simple trip planning web sites, which created a new market of online travel sales, with volume only in the USA of over $150B/year [15]. Generally there are many services for tourists and hospitality industry. But, basic services are not sufficient anymore. Tourists demand smart pro-active services for planning trips, onsite supporting, managing memories and sharing experience after the trip.

At the same time, there is demand from developers to reduce the complexity and the amount of resources required for development and maintenance of such services. For example, because of the development complexity, most services are missing on-site support and are isolated from each other. In this study, we target to come up with the toolkit for efficient development of such services. As a result, the development cycle shall be shortened and on-site support of the tourists and data exchange between the services provided from the box will take place. In the following, we start summarizing the main identified problems for this domain:

1) “cold start” as most services cannot collect the critical mass of relevant content;
2) update delays, which might results in significant decrease of relevance of the content and sometimes be even misleading;
3) user’s content and settings cannot easily be shared between the apps;
4) development of smart services related to very high implementation and maintenance complexity.

Due to the above listed problems current e-Tourism ecosystem is highly fragmented and slow in adoption ideas of advanced smart services. As a solution we propose to use LBS platform that provides common ground for tourist services and services of the hospitality industry.

Most of available LBS platforms have high entrance threshold for developers, i.e., license fee, high complexity of development or both. This stops developers from broad adoption of LBS-enabled architecture in the services. Also most of available LBS platforms are lacking efficient mechanisms for keeping data up-to-date and only a few LBS platform are IoT-enabled, which is strict requirement for the future-proved services.

Based on earlier performed analysis of various LBS ecosystems [16], we selected Geo2Tag LBS Platform [5]. Nowadays, Geo2Tag is the most popular open source LBS platform in the world [17]. Geo2Tag is recommended by IEEE Internet of Things technical community for prototyping IoT solutions in City tagging scenarios [18]. Moreover, Geo2Tag is available for deployment in clouds, can be provided as PaaS, or configured on standalone server or even on the network of regular PCs. This provides developers and users with widest possible choice of architectures and development approaches for their solutions.

A number of services and applications are implemented on top of Geo2Tag, including a few cultural heritage systems and tourist services, e.g., Open Karelia network [19] and New Moscow cultural heritage system [20]. What is important is that these solutions are not just stand alone museum systems, but a service overlay on top of LBS platform. Such design is very scalable and provides complete infrastructure for the development of LBS services for networks of regional museums. So far, a number of regional museum networks and wide variety of supporting services were created on top of Geo2Tag cultural overlay. A few local SME and independent developers use the system and are able to exchange relevant content between the services and share functional primitives published in community-developed open source libraries.

Content is the key asset for all tourist services and its availability has core meaning for the popularity of a service. Building critical mass of content for new services is currently a complex task that requires a lot of time and resources. At the same time there is a lot of open data on culture and history (as well as in many other application domains). In this study we target to make content of open databases available for services in LBS platforms.

The goal of this project is to provide developers with the enhanced set of tools for importing cultural heritage information to the Geo2Tag database. We developed an efficient content importing tool that was tested with popular European open databases and Wikipedia. Further details of the developed toolkit are discussed in the next section.

 ISBN: 978-1-61208-505-0

We are going to continue this work by expanding the list of supported open databases. Finally, we target to provide import access to major open data on cultural heritage [21][22][23], plus add support of data import from Europeana and make in-depth analysis of additional sources of relevant content. Moreover, to help with content management we are going to provide developers with special Geo2Tag modules (agents) for automatic search and filling of the missing fields (with unspecified/null content). Plus, we target to release a special content kick-off admin tool to help create an initial set of content for a service based on most relevant open data, and these new results will be included to our next publication.

IV. Implementation of the Proposed Extensions

Implementation of the open data importer for Geo2Tag is done based on definition of import API on top of user's plug-in subsystem available in the platform. The interface includes module that defines general import algorithm, library of abstract classes for all steps of import process and the set of REST-interfaces for controlling and managing the import procedure. Implementation of the import procedure has total length of 1113 lines of code and is available for free download as open source extension included to the default main package of the Geo2Tag platform [5]. The algorithm developed in the import API use the following source data:

- Channel ID (the named set of data);
- Link to the original set of open data;
- Name of the Geo2Tag service - destination of the imported data.

The import algorithm implements the following four steps:

1. Downloading data. This step contains all actions required for establishing connection, authorization and downloading data from an open data database.
2. Fragmentation of the downloaded open data set to the level of individual elements. At this step the algorithm performs de-serialization of the initial set of data.
3. One by one translates each individual element from the set of open data to Geo2Tag compatible format. At this step, the algorithm extracts geo-location and time-stamps attributes from the original open data element and saves them to the newly created geo-tag element in Geo2Tag database. In addition, content of the new geo-tag is associated with metadata on current import session, i.e., link to the original data set and the time-stamp of the import procedure.
4. Saves created set of geo-tags in the Geo2Tag database available for direct access by the service.

The set of abstract classes of the import API contains logic of each step of the import algorithm plus templates for implementation of REST-interfaces for controlling import procedure. Further details on the design of this part of the algorithm are discussed by Zaslavskiy and Mouromtsev [24].

The proposed method includes procedure for geo-context layout of the open data. Implementation of this procedure shall be supported by the set of corresponding basic primitives and tools provided by Geo2Tag platform. Moreover, due to specifics of organization of open data the location information could be stored in various formats, starting from geo-coordinates and up to street address or even textual description of a place. This creates the new challenge of building universal solution for extracting geographical coordinates for the street address (this is relatively easy as the corresponding libs are available) and the text description. To address this demand (which could be also useful as independent service for some application) we have developed special Geo2Tag plug-in, that solving the problem of direct geo-coding. This plug-in extends functional of Geo2Tag platform by the following two REST-interfaces:

- /instance/plugin/geocoding/service/<string:serviceName>/job – creates task on packet geo-coding of data for the specified service channel serviceName;
- /instance/plugin/geocoding/service/<string:serviceName>/job/<string:jobId> - provides control of the created task.

As an example of the real use case created on top of the developed API we developed a plug-in to import data from the Open Karelia public museum overlay to make it available for use by any services in Geo2Tag. This source of open data has been selected for the first pilot as its objects contain information about the location in the form of geographical coordinates and time presented in an interval-based format of dates with "B.C." trigger. This let us to confirm that the developed algorithm works and could successfully process complex data structures of the open data defined in formats that are unfriendly for machine processing.

The next step was to confirm that the proposed method is efficient enough to be executed even on the mobile devices, i.e., so that this import procedure could be directly used by smart services on smartphones. We tested implementation of the method on the low-performance home PC with the following formal characteristics:

- Dual-core CPU processor @ 2.10GHz – this corresponds to the top clock frequency for smartphones;
- RAM: 3Gb – this volume was selected as it is maximum volume of RAM on modern smartphones.

We choose to use this PC instead of direct testing on the smartphone as it helped us to save time on implementation of the test environment plus let us use most well known and verified monitoring tools. And the PC parameters were selected in a way to formally match to the top smartphones on the market. But, as we know smartphones are able to more efficiently use memory and processing power, so the obtained results could be safely referred as a "bottom" performance estimate for modern smartphones.

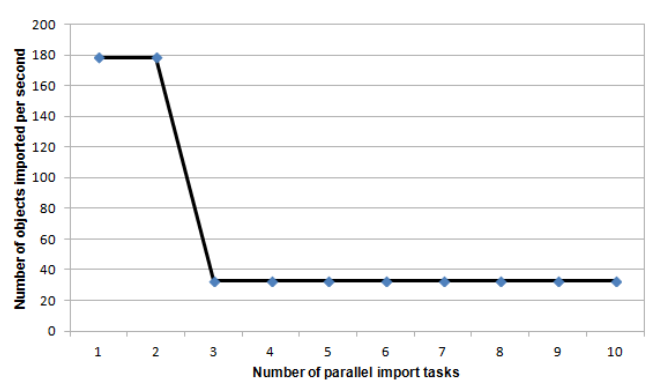

Figure 1. Dependency of the number of imported objects per second on the number of parallel import tasks.

The first set of performance tests is measuring performance of the import plug-in when serving a number of parallel import tasks. This test case simulated situation when a few services have to perform parallel importing of the content from open data databases.

In fact, the most typical scenarios for smartphones is that there is one or two parallel tasks that might need to simultaneously import content. But, we decided to makes the corresponding simulations for up to 10 parallel import tasks. The result dependency of the number of imported objects per second on the number of parallel import tasks is presented in Fig. 1. The algorithm shows really good performance of approximately 180 imported objects per second for the cases of one and two parallel tasks. When the number of parallel tasks exceeds the number processor cores we see step degradation of performance, which is direct result of multi-flow scheduling that has to be used in such cases. But, good news is that all modern smartphones are multi-core so we can expect similar level of per task performance, but for a larger number of parallel tasks, which is very important as with development of smart proactive services we can expect to see up to 10 parallel tasks on importing additional content to support efficient decision making.

The next analyzed key indicator was the time expectancy for the complete import procedure. The corresponding experimental results are shown in Fig. 2, where one can see the dependency of the maximum, minimum and average import time on the number of parallel tasks.

The results presented in Fig. 2 show that despite the restriction of the maximum number of concurrent executed threads and the Global Interpreter Lock (GIL) technology in Python [25], the average import time is growing slowly.

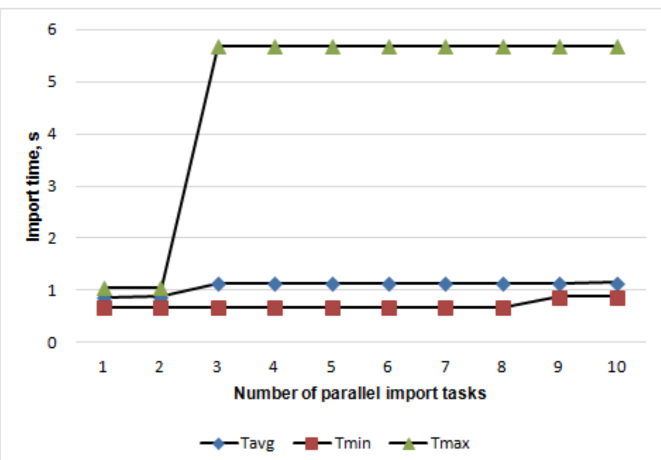

Figure 2. Dependency of the maximum, minimum and average import time on the number of parallel tasks.

When the number of parallel tasks exceeds the number of processor cores we observe sharp step increase for the maximum time of import. As discussed above, it is caused by the fact that in this experiment we used only built-in mechanism for managing connections to the web server Apache and the GIL and have not apply advanced scheduling tools. This leads to inefficient performance when one core has to process multiple tasks. At the same time one can notice that after this step change the value of maximum import time remained virtually unchanged with the growth of the number of parallel import tasks. This can be explained by the operation of the MongoDb cache [26].

Based on the received good performance levels and stable results without strict dependency on the number of parallel tasks, we can conclude that the proposed import method can be used for on-fly support of multiple smart services when executed directly on a smartphone with Geo2Tag support. As a result the developed solution can significantly simplify and speeds-up development and testing of services and enable “hot start” of new services. The proposed import extension has been committed to the open source LBS platform Geo2Tag and currently it is included to the default installation set of the platform.

V. Conclusion and Future Work

This paper proposes a new reference architecture for personal smart spaces, where the center of smart space is hosted by the smartphone. This idea is not new, but such architecture could not be implemented in the past. The main reasons were lack of efficient mobile platform to host SIB and infrastructure for efficient management of context data based on geographical location of the user. Within the scope of this study, we developed all basic elements to enable dynamic personalization of the virtual and physical environment based on the preferences of the user. As a result, we can conclude that this study gives a valuable scientific contribution to the development of architectures for various use cases of smart services in the Internet of Things environment. There is still the unsolved problem of collaborative personalization as the environment has to adapt itself to multiple users. The size of gravity field of each personal smart space shall be dynamically defined at any moment in time. This is an important topic for study that we are planning to address in the future.

Another direct conclusion of this study is that, currently, Sailfish OS provides the best ecosystem for implementing personalized M3-enabled smart spaces. Based on analysis and received practical results, our next step will be to implement a fully functional IoT management architecture in personal smart space deployed on top of the Smart-M3 platform.

Another aspect that we are going to address in the follow up study is the development of business model that takes into account interests of smartphone and IoT-devices producers. Our preliminary analysis shows that the discussed technical architecture provides a very reasonable solution, as it does not demand reallocation of business niches. In this case, one can see the personal smart spaces architecture on Sailfish OS as the business peacemaker for mobile device producers and manufactures of IoT devices. But, in-depth study of these issues is required and we also target to develop a solid proposal on the monetization scheme for personal smart spaces services. Moreover, we would like to come up with an analysis of possible roadmaps that combine smartphones and IoT ecosystems into a solid personalized service provision infrastructure that will surround users 24/7.

Finally, the paper makes an important step forward in the development of the LBS-ecosystem by offering an universal

 ISBN: 978-1-61208-505-0

solution for importing most relevant content using open data databases and defining relevance based on geo-context. As a result, this helps to address the "cold start" problems as well as a few other problems highly relevant for the service developers. The developed content importing algorithm was implemented on the Geo2Tag LBS platform. The performance tests show that the proposed import method can be used for on-fly support of multiple smart services, when executed directly on a smartphone. Our import extension has been committed to Geo2Tag and currently it is included to the default installation set of the platform. A task for further study is to analyze the dependency of import tasks performance characteristics on the number of parallel tasks when executed on Sailfish OS operated multi-core mobile device.

ACKNOWLEDGMENT

Authors of ITMO University are thankful for the financial support provided by Government of Russian Federation, Grant 074-U01. Ekaterina Balandina is grateful for support provided by the Graduate School DELTA via Tampere University of Technology. The authors thank Geo2Tag community for great work on development and maintenance of the open source LBS platform Geo2Tag.

REFERENCES

[1] K. Kulakov, O. Petrina, D. Korzun, and A. Varfolomeyev, "Towards an Understanding of Smart Service: The Case Study for Cultural Heritage e-Tourism", in Proc. of the 18th FRUCT & ISPIT Conference, 18-22 April 2016, Saint-Petersburg, Russia. IEEE, pp. 145-152. 2016.

[2] 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, pp. 3–12. 2009.

[3] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the Internet of Things," IEEE Internet Computing, vol. 14, no. 1, pp. 44–51, Jan. 2010.

[4] E. Balandina, S. Balandin, Y. Koucheryavy, and D. Mouromtsev, "Innovative e-Tourism Services on top of Geo2Tag LBS Platform", the 11th International Conference on Signal Image Technology & Internet Systems (SITIS 2015), Bangkok, Thailand. pp. 752-759. 2015.

[5] "Web portal of Geo2Tag developers community", FRUCT Ltd, URI: www.geo2tag.org. retrieved: August, 2016.

[6] "Market value Open Data to reach 286 billion by 2020", European data portal, URI: http://www.consultancy.uk/news/3019/market-value-open-data-to-reach-286-billion-by-2020, 9 December 2015. retrieved: August, 2016.

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

[8] N. Lebedev, I. Timofeev, and I. Zavialova, "Design and Implementation of the First Aid Assistance Service Based on Smart-M3 Platform", in Proc. of the 18th FRUCT & ISPIT Conference (FRUCT18), IEEE, pp. 174-180. Apr. 2016.

[9] C. Weinschenk, "Tizen: We Can Be the OS of the IoT", ITBusinessEdge. URI: http://www.itbusinessedge.com/blogs/data-and-telecom/tizen-we-can-be-the-os-of-the-iot.html. Jun 2014. retrieved: August, 2016.

[10] D. Luis, "Sailfish OS discovers its MeeGo roots on the Nokia N9", phoneArena.com, URI: http://www.phonearena.com/news/Sailfish-OS-discovers-its-MeeGo-roots-on-the-Nokia-N9_id50240. 11 December 2013. retrieved: August, 2016.

[11] D. Korzun, A. Kashevnik, S. Balandin, and A. Smirnov, "The Smart-M3 platform: Experience of smart space application development for Internet of Things", in Proc. of the 15th Conf. Next Generation Wired/Wireless Networking and 8th Conf. on Internet of Things and Smart Spaces (NEW2AN/ruSMART 2015). LNCS 9247. Springer. pp. 56–67. Aug. 2015.

[12] A. K. Dey, "Understanding and using context," Personal and ubiquitous computing, Vol. 5, No. 1, pp. 4–7, 2001.

[13] I. Paramonov, A. Vasilyev, and E. Mamedov, "A conceptual framework for development of context-aware location-based services on smart-M3 platform," in Proc. of 17th conference of Open Innovations Association FRUCT. IEEE, pp. 142-150. Apr 2015.

[14] J. Ellacott, "Global LBS Platform Market 2015-2019 - Increased Demand for Location-based Services", BusinessWire. URI: http://www.businesswire.com/news/home/20150716005694/en/Increased-Demand-Location-Based-Services-Improve-Global-LBS, 16 July 2015. retrieved: August, 2016.

[15] M.C. Rodriguez-Sanchez, J. Martinez-Romo, S. Borromeo, and J.A. Hernandez-Tamames, "GAT: Platform for automatic context-aware mobile services for m-tourism", Expert Systems with Applications, Vol. 40, pp. 4154–4163. 2013.

[16] E. Balandina, S. Balandin, Y. Koucheryavy, and D. Mouromtsev, "IoT Use Cases in Healthcare and Tourism", in Proc. of the 17th IEEE Conference on Business Informatics (CBI 2015), Lisbon, Portugal. pp. 37-44. 2015.

[17] "Open Source LBS Platforms", MetricsKey rating, URI: http://metricskey.com/open/open-source-location-based-services/, retrieved: August, 2016. & and "LBS Platform" keyword search, Google, URI: https://www.google.fi/search?q=LBS+Platform. retrieved: August, 2016.

[18] "Recommended LBS platforms for prototyping IoT solutions in City tagging scenarios", the reference is in the bottom of the first page of PDF, IEEE Internet of Things, URI: http://iot.ieee.org/iot-scenarios.html?prp=6. retrieved: August, 2016.

[19] "Open Karelia cross-border museum network", Hypertext ENPI CBC project, URI: www.openkarelia.org. retrieved: August, 2016.

[20] "New Moscow cultural heritage system", Mosart, URI: http://www.novmosdata.ru/. retrieved: August, 2016.

[21] Set of Open Culture Data in Netherlands databases, URI: http://www.opencultuurdata.nl/datasets. retrieved: August, 2016.

[22] Database of open data in Italy, URI: http://www.iccd.beniculturali.it. retrieved: August, 2016.

[23] Set of Open Culture Data in Russia databases, URI: http://mkrf.ru/opendata/. retrieved: August, 2016.

[24] M. Zaslavskiy and D. Mouromtsev, "Implementation of the new REST API for open source LBS-platform Geo2Tag", in Proc of Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), IEEE. pp. 125-130. Nov. 2015.

[25] D. Beazley, "Understanding the python GIL", PyCON Python Conference. Atlanta, Georgia. 2010.

[26] K. Banker, "MongoDB in action", Manning Publications Co, 2011.

 ISBN: 978-1-61208-505-0