Geo-Coded Environment for Integrated Smart Systems

Kirill Krinkin
Open Source and Linux Lab
Academic University RAS
Saint-Petersburg, Russia
kirill.krinkin@fruct.org

Kirill Yudenok
Department of Software Engineering
Saint-Petersburg Electrotechnical University
Saint-Petersburg, Russia
kirill.yudenok@gmail.com

***Abstract*—Smart Systems provide novel enabling functionalities and as such are currently a driving force behind product innovation. Smart Systems are, therefore, crucial for the competitiveness of companies and entire industry sectors. Geo-tagging and smart spaces are two promising directions in modern mobile market. Geo-tagging allows to markup any kind of data by geographical coordinates and time. This is the basis for defining geographical context which can be used in different types of applications e.g., semantic information search, machine-to-machine interactions. Smart spaces as the basis for seamless distributed communication field for software services provides semantic level for data processing. The paper is targeted to discuss opportunity of using geo-coded smart spaces in integrated Smart Systems.**

Keywords-geo-tagging; geo-coding; Smart Spaces; Smart System; LBS.

I. INTRODUCTION

Nowadays, we have two most promising software trends – location based services and pervasive smart environments (smart spaces). Both of them will be a base for user- and machine-oriented proactive services. Smart spaces should provide continuous distributed semantic data and communication field for software services, which is being run on personal devices and autonomous computers and robots. The most desired features of coming software is pro-activeness and context awareness, i.e., services will be able to adapt to the user's needs and situations and be able to manage decisions and behaviors on behalf of the user [1]. One of the important part of context is location-based data. These data are being used for two purposes: for clarifying semantic meaning of queries (when service retrieves the data from smart environment) and for limitation of space of search (usually, there is no point to make global search). Geo-coding (or geo-tagging) is the technique of markup real or virtual object by adding geographical coordinates and time. If we consider software, we have only virtual (or digital) objects like media, events, documents, etc. So far, smart spaces and geo-tagging systems are being developed mostly separately, there are only few works [2][3][4], where software design of smart spaces and geo-tagging integration are discussed.

This paper discusses the definition of a Smart System, based on the creation of an integrated platform as a part of the device for implementing the basic Smart System properties, Smart System use-cases, its architecture and criteria for the analysis of the constructed system.

This document proceeds as follows. Section II provides our definition for Geo Codes Smart System. Section III gives geo-coding problem for smart spaces. System Requirements are discussed in Section IV. In Section V, a high level design is considered. Section VI provides platform integration agent architecture. Smart-M3 and Geo2Tag data integration principles are presented in Section VII and the conclusion is presented in Section VIII.

II. GCSS SMART SYSTEM DEFINITION

A Smart System, called intellectual integrated system, has the following main features:

- System with a clear goal, which determines the directionality of the system;
- System receiving information from the outside world, a person, other systems (data acquisition);
- System responsible for processing information and making decisions to achieve the goals of the system.

Smart Systems must have the following properties:

- *Autonomy* – ability to operate without human intervention or other systems;
- *Openness* – the independent ability to interact with the physical and virtual worlds objects (systems, tools, people), the collection of information and influence to them. It also includes the ability to use and provide external interfaces;
- *Context-awareness* – the ability to independently collect contextual data and analyze the situation;
- *Self-organization* – the ability to maintain the autonomy, to control their own parameters and to select behavior strategies;
- *Purposefulness* – the presence of individual or collective goals and the ability of strategy synthesis and implementation;
- *Pro-activity* – the ability to predict the evolution of the situation in the future (see the decision tree), to determine the parameters of the desired impact and exercise influence.
- *Cooperativeness* – the ability to interact with other systems and/or the person to achieve a goal and assist in achieving the goals of other systems.

There are two possible views on a Smart System: interconnected devices and the device itself. Both of them rely on smart space middleware which provides semantic information sharing facility.

At the moment, there are not effective approaches for markup semantic data by temporal and spatial context for

 ISBN: 978-1-61208-318-6

using in integrated Smart Systems. For instance, if we have presented coordinates and time as traditional Resource Description Framework (RDF) triples, the system performance will be not acceptable, due the big amount of data for processing. On the other hand, there are number of systems for fast temporal and spatial search and filtration. For most of integrated Smart Systems next functions are absent, but required:

- Search objects (usually RDF triples) by given time intervals;
- Defining set of objects which are enclosed inside geographical region or defined spatial structure (buildings, squares, etc.).

The main goal of this work is to suggest an approach for building integrated Smart Systems with using such data model and program interfaces, when advantages of using semantic and geographical markup are available at the same time. In the first instance, integrating Smart Systems are considered.

In other words, we need to design a system, which includes components to perform main and missing Smart System functions. Each system component is responsible for the execution own functions and also provides an Application Programming Interface (API). The integrated system should have a common communication interfaces, protocols and programming interfaces for interaction with other Smart Systems.

As initial system components, we have chosen Smart Spaces [5], Smart-M3 [6] platform, and the Internet of Things LBS Geo2Tag [7] platform, as ones of the fastest growing platforms of these areas. Smart-M3 platform provides a common communication field for cooperation and allows processing and storing semantic information (knowledge). LBS Geo2Tag platform is responsible for the provision of geospatial data from a variety sources.

To build a new Smart System based on selected technologies, we must consider the following aspects:

- Integrated system architecture development;
- Unification of general platforms levels (Smart-M3 and Geo2Tag);
- Common protocols and communication interfaces between device;
- Behavior model.

The main feature of the Smart System discussed in this project is the ability to connect the location data to any object in smart space.

An area such as Smart Systems can be used in various spheres of human activity:

- Space industry;
- Automotive industry;
- Information and telecommunication area;
- Internet of Things;
- Energetic industry;
- Medical area;
- Privacy and security.

In the solutions proposed in this paper, the Smart System allows determining the location of each space object (thing, entity) in time. It can be used for spatial and geographical context clarification in order to increase context awareness of user-oriented services.

As one can be seen from the further evolution of the Internet of Things direction in 2015-2016 [8], it will be possible to identify the location of all the people and of the objects of everyday use. That being said, the fact that every object around us is endowed with information and a variety of sensors. This enables obtaining the necessary information in real-time mode about its conditions and the surrounding objects state.

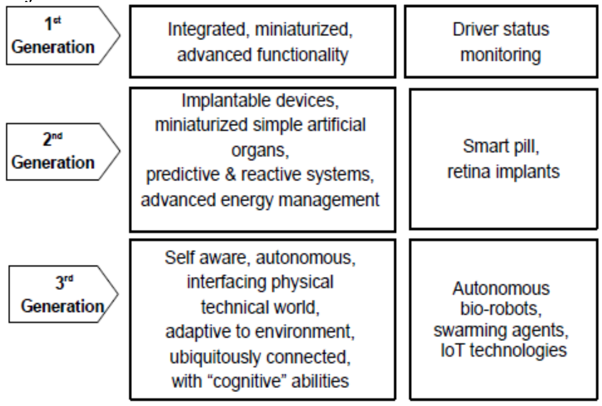

Figure 1. Continuing revolution of Smart System Integration

Today, there are prototypes of Smart Systems that have reached the state of commercial products. In Fig. 1, three generations of Smart Systems integration are presented [9].

III. SMART SYSTEM GEO-CODING PROBLEM

To solve the problem of smart space and its subspaces subjects search, two options can be considered.

For the search world subjects in any space and their subspaces, there are two solutions, namely, (i) geo-coding, and (ii) coordinates determination. Geo2Tag platform supports only subjects geo-markup in a certain area (map); this makes it possible only to search subjects in a given space. In order to be able to markup and find subjects not only in a given space, but also in its subspaces, e.g., in the space of a building, city, street, etc., it should be possible to determine the coordinates of subjects in the predetermined space or their markup by a predefined plan of space subjects, such as subspace ontology, map ontology, etc. This method can also be used to determine the coordinates of the moving objects.

To resolve this problem within a smart space it is required to develop a special knowledge processor (KP); by composing a subspace ontology, it creates a semantic representation model. Coordinates of each subspace subject are obtained from Geo2Tag platform after its labeling on pre-created visualization subspace.

In other words, the subspace is created using a special visualization technology [10]; then, all subjects are placed into this subspace. These subjects are marked within a special representation subspace map (all markup subjects are assigned the coordinates), leading to a representation ontology of the given subspace. KP ontology is used for processing and obtaining information from the given subspace.

Fig. 2 shows the geo-location approach on the example of the Smart-M3 platform. The main Geo-location KP is designed to handle all the information from the geo-space. Ontology Space Creation KP will be used for the building subspaces from its ontology.

Context management system KP server for convenient context management processed by all (sub)spaces. Also, the approach includes various space sensors, KPs for processing coming information from the sensors, and also the representation and description space ontologies with its domains.

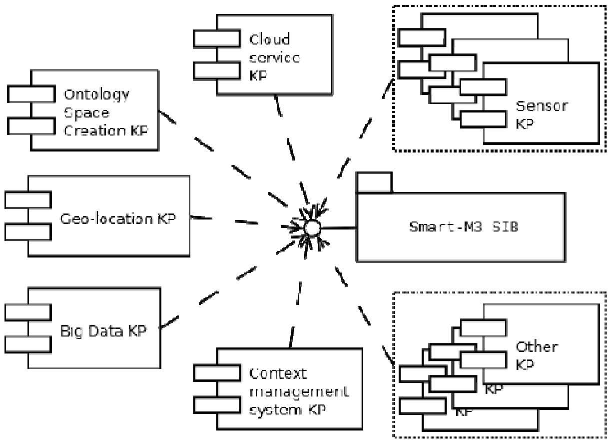

Figure 2. Geo-location Smart-M3 approach

This scheme could be extended by adding knowledge processors for off-line data processing.

Further on, we will present the geo-location agent used to integrate Smart-M3 and Geo2Tag platforms for processing subjects' coordinates in the space.

IV. GCSS SYSTEM REQUIREMENTS

The first main task of the Smart System platform is the integration of Smart-M3 and Geo2Tag platforms, and also expanding the smart space with new data, e.g., geo-data [10]. There are several promising use-cases of Geo-Coded Smart Space (GCSS):

- Geographical markup of smart space data;
- Search set reduction;
- Search context rectification.

GCSS should implement main features from both types of platforms, which are:

- Providing interfaces for semantic data and access;
- Smart-M3 API – Qt [11], Python [12], Java [13];
- Distributed storage for semantic information;
- Interfaces for association semantic objects with geo-tags;
- Spatial and temporal filtration.

Also non-functional requirements should be taken in account:

- Performance – ability to work with big amount of semantic objects geo-tags like cloud based massive offline processing and local context indexing/caching.
- Compatibility – the GCSS should be accessible by legacy interfaces (i.e., SSAP or REST), which is required for seamless integration with existing systems.

Below is the list of the main functional use-cases of the integration platforms agent:

- Smart space Smart-M3 platform management (leave, join, query, insert, delete, update, subscribe, unsubscribe);
- Geo2Tag platform management (connect, disconnect, obtain platform data, search, filtration);
- Geo-tags conversion mechanism to space data (triples) and vice versa;
- Smart space searching and filtering algorithms by means of Geo2Tag platform;
- Ranking mechanism of space data (the algorithm of selection the latest objects by location, optional);

The last three use-cases are fulfilled by the main features of the agent to increase the space with new information, i.e., geo-data; that will be used to determine the location and search for objects in space. The first two are available on Smart-M3 and Geo2Tag platforms.

V. GCSS HIGH-LEVEL DESIGN

High-level layered design for GCSS is presented in Fig. 3. Each level of the system is responsible for the functions and includes its own interface. The following are the layers of the system GCSS:

Interfaces level is responsible for data representation and processing for applications and services;

Integration level contains components for translating geographical data from Geo2Tag format to Smart-Space format and vice versa;

Domain engines level contains particular implementations of smart-space and geo-coding middleware;

Data cloud backend – optional components, which is being used for providing advanced services like off-line data pre-processing, storage for Binary Large Objects, indexing, caching, etc.

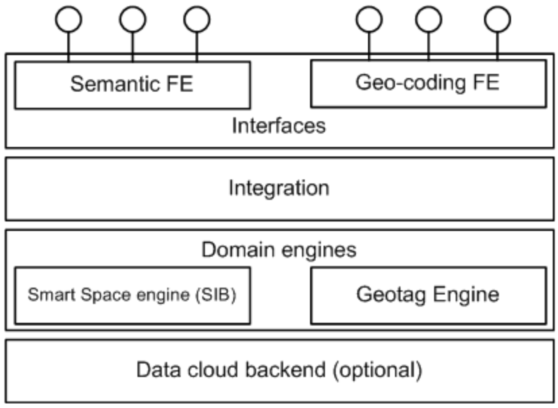

Figure 3. High-level layered design of GCSS

There are five basic components (levels) [14] that provide basic functioning contour of the system (system life cycle):

Data acquisition level – presented by sensors and other receiving information interfaces from the outside world, a person, other systems;

Data pre-processing level – data storage and transformation of the primary form to a form suitable for analysis and decision–making;

Decision-making level – module responsible for information processing and making decisions to achieve the goals of the system, and support tasks related to self-diagnosis and self-organization;

Command level – responsible for making the transformation into control signals own functional components and external systems for the environmental impact implementation;

Action level – implementation of information and physical control of external systems, including the task of encoding and transmitting control signals to run-time systems and control command execution.

The main object of the platforms integration is the integration agent or mediator. Its primary task is to provide interaction between Smart-M3 and Geo2Tag platforms and the platforms data conversion into one common format (triplets). Each platform has the necessary programming interface (API).

VI. PLATFORM INTEGRATION AGENT

The integration agent (GCSS) is responsible for the platforms integration and fills the Smart-M3 space with geo-data by conversion mechanism. Next, the agent will combine the functionality of both platforms (Smart-M3 and Geo2Tag) and will become a sort of common platform within the device to control and manage data between all smart space devices [15].

One can create a agent ontology by using special Smart-M3 ontology generator, i.e., Smart Slog [16][17].

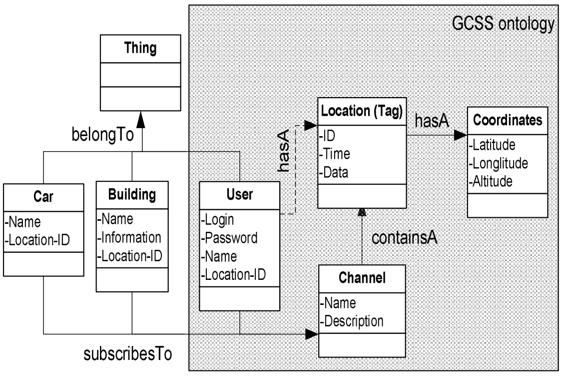

Figure 4. Overlay ontology used by GCSS

GCSS ontology consists of four classes – User class, Channels class, the Tag itself, and its Coordinates. Class User is responsible for a user's of the Geo2Tag platform in space, the tags channel describes a set of tags for a given criterion, the Tag class describes itself data. It should be noted that users can subscribe to an unlimited number of channels, as well as a channel can contain unlimited number of tags. Class User can directly communicate with the tag through the property hasA. Coordinates are allocated in a separate class for more convenient their representations in an agent ontology. More details on GCSS ontology are described in [18].

Each Geo2Tag platform user, if it exists, will be associated with its own user in the Smart-M3 space; if not, then, a new space user will be created; this will be automatically attached to the tag location and to the channels which it subscribes to. Location may be attached to any space object after adding new property (e.g., Location-ID) in the object class of the space ontology. Class Tag property Data is mainly used for searching and filtering space objects, but it can be also used for its association with the object.

It should be noted that the user location or other space object (not static) can change location with time and in order to remain relevant data necessary to provide handling this situation. Smart-M3 platform provides a publisher-subscriber mechanism; by subscribing to specific triplets, the object will automatically receive new data after each change. In our case, these data are the properties of the Coordinates class.

The agent will use the object model of the ontology representation, i.e., have clearly documented ontology classes names and their properties, as well as certain triplets (subscription). Thus, the space agent ontology will look like a list of properties that are linked by a predicate. In the first version of the integration agent, the space will be filled only with geo-data, which will be linked with their space objects (a person, object, etc.). In the future, we plan to expand the space by the addition of the users and channels tags information.

All Geo2Tag platform data are stored in a database on a dedicated server. Geo2Tag platform allows recording and retrieving data using Representational State Transfer (REST) specific queries [19] in Java Script Object Notation (JSON) format [20]. There is also a variety of clients to work with a Geo2Tag platform, mainly for mobile platforms.

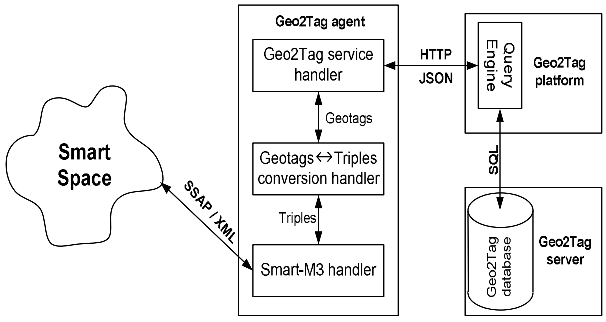

Figure 5. GCSS architecture

The integration with the Smart-M3 platform will be implemented through a special mediator (agent). Its main task is to convert data from one platform format (Geo2Tag, JSON) to another format (Smart-M3, XML). As mentioned above, the Geo2Tag platform transmits data in JSON format; this is a text format, but in a more readable form for humans.

The agent consists of three main components:

- Geo2Tag service handler;
- Geotags – Triples conversion handler;
- Smart-M3 handler.

Geo2Tag service handler is responsible for obtaining geo-data, it connects to the server database and requests data using a special class LoadTagsQuery.

Geotags–Triples conversion handler is required to bring data to a convenient form for the triplets creation. Since the data are returned in JSON format, they need to be parsed and pulling the necessary data, namely, time, location and description of the geo-tag by saving them for later processing.

At the last stage, it is a connection to the space; then, triplets are created, according to the ontology. Finally they placed into the Smart-M3 space. Below, a list of the location triplets that are created during conversion mechanism is presented:

- < User, "hasA", Location-ID >
- < Location, "hasID", ID >
- < Location, "hasTime", Time >
- < Location, "hasData", Data >

- < Location, “hasLatitude”, Latitude >
- < Location, “hasLongitude”, Longitude >
- < Location, “hasAltitude”, Altitude >

The main evaluation criteria of the Smart System platform will serve for its performance, the ability to integrate into embedded devices, the amount of transmitted traffic, and the response speed. The analysis should show how the platform behaves in the real conditions and only then takes steps to improve its operability.

VII. GEO2TAG AND SMART-M3 INTEGRATION

One of the main action of the integration agent use-case is the geo-tags conversion mechanism to the space triplets; below the pseudo-algorithm is presented;

Connect to the Geo2Tag platform by using a *Login()* query;

Point service (database) by *setDB()* query, where data will be obtained;

Sampling nearest tags with a *LoadTags()* query or *Filter()* inside the defined geometry figure;

Obtaining the necessary tags parameters from received data (JSON format);

Formation of the initial triplets for space objects representation by class Triple(S, P, O): a triplet for linking space object with its location, a triplet for location time, coordinates and data. In general, six triplets describe *Location (Tag)* of space object.

Connection and insertion triplets in space are done with the help of Smart-M3 API.

After the execution of the algorithm, the space will be filled with latest tags from the Geo2Tag server database. The inverse transform mechanism (triplets to the geo-tags) are quite similar, but only performed at the Smart-M3 platform.

Now, the integration agent responsible for the platforms integration fills the space (Smart-M3) with geo-data by conversion mechanism. Next, the agent will combine the functionality of both platforms (Smart-M3 and Geo2Tag) and will become a type of common platform within the device to control and manage data between all smart space devices.

The next main platforms integration agent use cases are smart space data searching and filtering algorithms.

A filtering mechanism of space data is required to obtain relevant information at the moment when the system works; therefore, by filtering the objects by location, we will have a list of the most relevant data at a given time. We consider a filtering data mechanism based on their metadata obtained by SparQL queries [21].

Each ontology object has a set of metadata, for example, *Id*, *Description*, *Type*, *Time*, *Position*, *Status* (e.g., *Offline, Online, Connecting*). Object metadata is used in the filtering process to retrieve only those objects that satisfy the consumer (client) requirements.

Searching and filtering algorithms are based on the Geo2Tag platform is filtering queries; as a result, as SparQL queries require significantly more Smart-M3 resources, it might affect to the performance of the whole system.

A general Smart Space data filtering algorithm scheme based on the Geo2Tag platform via Smart-M3 is presented in Fig. 6.

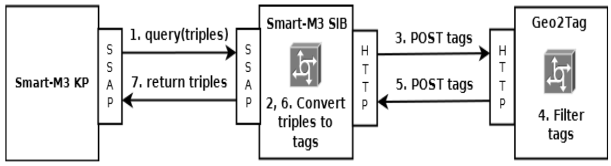

Figure 6. General Smart Space data filtering algorithm scheme

This filtering method operates as follows: KP sends a request to the Smart-M3 for sample required triplets; then, Smart-M3 makes a request to the Geo2Tag for retrieval necessary data, for example, by Radius [22]; Smart-M3 converts the triplets to tags and sends them back to KP.

After the development of the Smart System platform within the device, each of its mechanisms (algorithm) will be subject to thorough analysis by the following criteria: universality, performance, resources, the ability to integrate into embedded devices, memory size, the amount of transmitted traffic, response speed [23]. We expect that the analysis will show how the platform behaves in the real conditions, to improve its operability.

VIII. CONCLUSION

In this paper, we proposed a description of a Smart System based on common device platforms; we discussed the requirements and use-case of platform systems, its high level design and architecture for smart-space and geo-coding middleware integration. This integration could be made by using special Smart-M3 Knowledge Processor, which monitors both spaces and translates data from one to another and vice versa.

The current results of the project:

- Integration platforms agent prototype;
- Geo-tags conversion mechanism;
- Filtering mechanism based on the platform Smart-M3.

The next step in the development of Smart Systems device platform is the complete platform components integration, common protocols and interfaces for communicating between all devices. There are still open questions for future development: overall system performance, effective object monitoring, temporal and spatial filtration, integration with media objects.

ACKNOWLEDGMENT

The authors would like to thank Finnish Russian University Cooperation in Telecommunication Program for provided equipment and JetBrains Company for financial support.

REFERENCES

[1] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context Aware Computing for The Internet of Things: A Survey”, Communications Surveys Tutorials, IEEE, 2013 pp. 1–44.

[2] N. Nabian, C. Ratti, A. Biderman, and G. Grise, “MIT GEOblog: A platform for digital annotation of space for collective community based digital story telling,” 3rd IEEE International Conference on Digital Ecosystems and Technologies, Piscataway, N.J., IEEE, 2009, pp. 353-358.

[3] J. Rishede, T. Man, and L. Yiu, “Effective Caching of Shortest Paths for Location-Based Services”, SIGMOD ’12, Scottsdale, Arizona, USA, 2012, pp. 313-324.

[4] K. Kolomvatsos, V. Papataxiarhis, and V. Tsetsos, “Semantic Location Based Services for Smart Spaces,” 2nd International

Conference on Metadata and Semantics Research (MTSR), 2007, Corfu, Greece, pp. 515-525.

[5] D. J. Cook and S. K. Das, "How smart are our enviroments?" an updated book look at the state of the art, Pervasive and Mobile Compution 3(2), 2007, pp. 53-73.

[6] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö, "Smart-M3 Information Sharing Platform", 1st Workshop on Semantic Interoperability in Smart Spaces, 2010, pp. 1041-1046.

[7] I. Bezyazychnyy, K. Krinkin, M. Zaslavskiy, S. Balandin, and Y. Koucheravy, "Geo2Tag Implementation for MAEMO", 7th Conference of Open Innovations Framework Program FRUCT, 2010, Saint-Petersburg, Russia, pp. 7-11.

[8] http://www.compression.org/energy-productivity-of-systems/, [retrieved: Jan, 2014].

[9] EPoSS Strategic Research Agenda 2009 – http://www.smart-systems-integration.org/public/documents/publications/, [retrieved: Jan, 2014].

[10] M. Nollenburg, "Geographical visualization", Human-Centered Visualization Environments, Lecture Notes in Computer Science, vol. 4417, 2007, pp. 257-294.

[11] http://qt-project.org/, [retrieved: Jan, 2014].

[12] http://www.python.org/, [retrieved: Jan, 2014].

[13] http://www.java.com, [retrieved: Jan, 2014].

[14] G. Akhras, "Smart Materials and Smart Systems for the future", Canadian Military. Journal 2000, pp. 25-31.

[15] D. Korzun, I. Galov, A. Kashevnik, N. Shilov, K. Krinkin, and Y. Korolev, "Integration of Smart-M3 Applications: Blogging in Smart Conference," Proc. 4th Conf. Smart Spaces (ruSMART 2011), Saint-Petersburg, Russia, 22-23 August 2011, pp.51-62.

[16] D. Korzun, A. Lomov, P. Vanag, J. Honkola, and S. Balandin, "Generating Modest High-Level Ontology Libraries for Smart-M3", Proc. 4th Int'l Conf. Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM, 2010, pp. 103–109.

[17] D. Korzun, A. Lomov, P. Vanag, J. Honkola, and S. Balandin, "Multilingual ontology library generator for Smart-M3 information sharing platform", International journal on Advances of Intelligent System 4 (3&4), 2011, pp. 68-81.

[18] K. Krinkin and K. Yudenok, "Geo-coding in Smart Environments: Integration Principles of Smart-M3 and Geo2Tag," In Proceedings of the 13th International Conference, NEW2AN 2013 and 6th Conference, ruSMART 2013, St. Petersburg, Russia, August 28-30, 2013, Proceedings. Springer 2013 Lecture Notes in Computer Science, pp. 107-116, ISBN 978-3-642-40315-6.

[19] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, [retrieved: Jan, 2014].

[20] http://www.json.org/, [retrieved: Jan, 2014].

[21] E. Nageba, P. Rubel, and J. Fayn, "Semantic agent system for automatic mobilization of distributed and heterogeneous resources," In Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics, WIMS '13,ACM, New York, NY, USA, 2013, Article 28, pp. 9-17.

[22] http://geo2tag.org/index.php/Exchange_protocol, [retrieved: Jan, 2014].

[23] M. Zaslavsky and K. Krinkin, "Geo2tag Performance Evaluation," Proceedings of the 12th Conference of Open Innovations Association FRUCT and Seminar on e-Travel, Oulu, Finland, 2012, pp. 185-193.