Geo-coding in Smart Environment: Integration Principles of Smart-M3 and Geo2Tag

Kirill Krinkin[1] and Kirill Yudenok[2]

[1] FRUCT LLC,
Saint-Petersburg, Russia
kirill.krinkin@fruct.org
[2] Saint-Petersburg Electrotechnical University,
Saint-Petersburg, Russia
kirill.yudenok@gmail.com

Abstract. 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 (M2M) interactions. Smart spaces as the basis for seamless distributed communication field for software services provides semantic level for data processing. Most desired feature 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. The paper is dedicated discussion of integration most popular open platforms for smart spaces and geo-tagging (Smart-M3 and Geo2Tag) as possible solution for creation context aware proactive location based (LBS) services.

Keywords: geo-tagging, geo-coding, smart-m3, Geo2Tag, LBS.

1 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. Most desired feature 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. This data is 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

S. Balandin et al. (Eds.): NEW2AN/ruSMART 2013, LNCS 8121, pp. 107–116, 2013.

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 is discussed. In this paper requirements and high level design for Integrated Geo-Coded Smart-Space (GCSS) middleware are discussed. Rest of paper is constructed next way: in second part we analyze related works, in third part requirements and architecture are proposed.

From practical point of view we use Smart-m3[1] and Geo2Tag Platform[2] as most developed open source middleware for smart spaces and geo-tagging. In the last part of paper we conclude proposed architecture and define directions for development.

2 Overview of Smart-M3 and Geo2Tag Platforms

2.1 Smart-M3 Platform

Smart-M3 is an open source software platform [5, 6] that aims to provide Semantic Web information sharing infrastructure between software entities and various types of devices. The platform combines ideas of distributed, networked systems and Semantic Web [7, 8]. The major application area for Smart-M3 is the development of smart spaces solutions, where a number of devices can use a shared view of resources and services [9]. Smart spaces can provide better user experience by allowing users to easily bring-in and take-out various electronic devices and seamlessly access all user information in the multi-device system from any of the devices [10].

The simplified version of the Smart-M3 smart spaces reference model is shown in Fig. 1. The Knowledge Processors (KPs) represent different applications that use the smart space. The smart space core is implemented by one or several Semantic

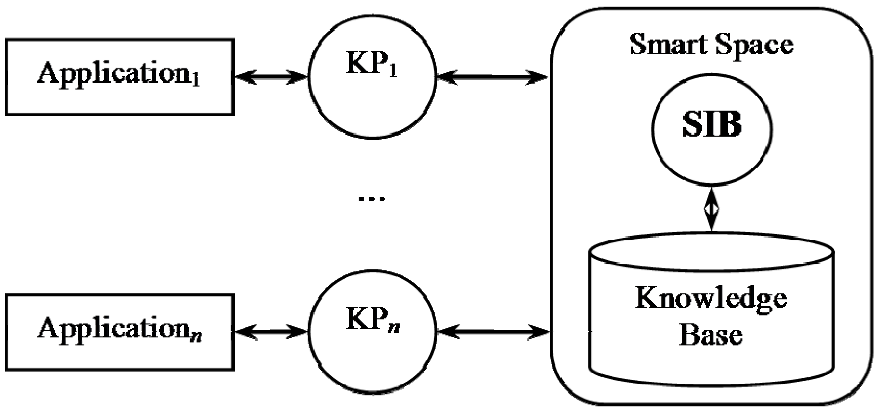

Fig. 1. Smart space based on Smart-M3: simplified reference model

[1] http://en.wikipedia.org/wiki/Smart-M3
[2] http://geo2tag.org

Information Brokers (SIBs) interconnected into the common space. The information exchange is organized through transfer of information units (represented by RDF triples) from KPs to the smart space and back. The information submitted to the smart space becomes available to all KPs participating in the smart space. The KPs can also transfer references to the appropriate files/services into the smart space, since not all information can be presented by RDF triples (e.g., a photo or a PowerPoint presentation). As a result the information is not really transferred but shared between KPs by using smart space as a common ground [11].

2.2 Geo2Tag Platform

Geo2Tag is centralized system with server, that storage all information and provide access to it by REST API. It has following main advantages – it is open source and it doesn't depend on concrete web-server or DB type. Server consists of two main parts – server application and database (DB). Server application is a web application written by FastCGI framework. It main task is processing clients requests to REST API. Database contains information about users and their geodata.

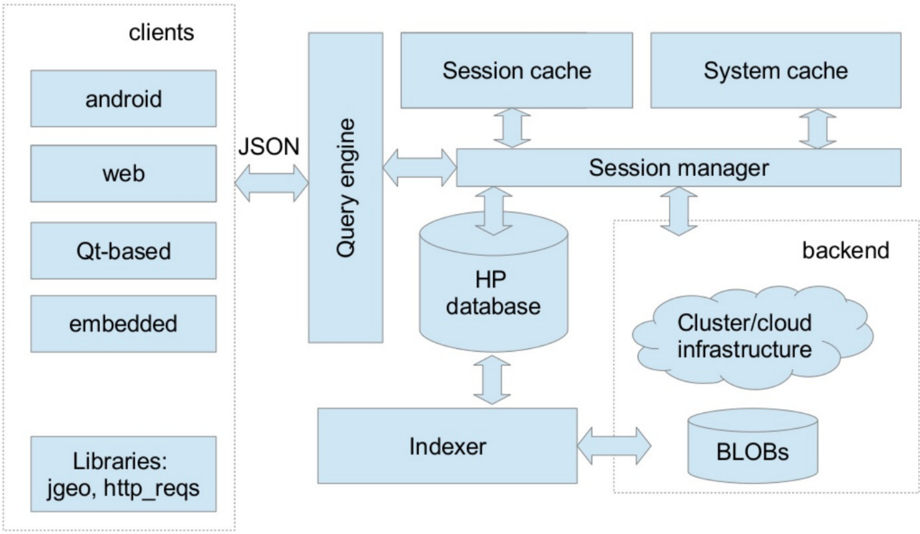

Fig. 2. Geo2Tag architecture

Platform use following basic data types:

- User – data that represent human user of Geo2Tag – login and password;
- Channel – object that contain name, description and set of tags. User subscription for the channel means that user can read tags from it;

- Tag – object that contain URL of multimedia object, name, description, time of creation and coordinates(latitude and longitude);

Main Features of Geo2Tag platform:

- user management: registration, login, log-off, session management;
- data retrieval about users and matching personal geographical spaces to the personal smart spaces;
- channel management: subscription/unsibscription;
- sending geographical data from smart-space to the geo-tagging system;
- getting data from geo-tagging system;
- spatial filtration;

More information about platform and design details can be found at related work – [12].

3 Related Work

The most overall point at the integration smart spaces facilities and geo-coding has been developed by Pervasive Computing Research Group. They focused on indoor Location Based Services (LBS) and coding real world objects. In [4] next basic approaches and components for integration are suggested: spacial ontology, ontology-driven map annotation, GIS-based ontology population and navigation algorithms. This approach is very justified from application point of view but has several limitation if we consider context free integration (without any assumptions about application domain).

In [13] we can find an idea about tree-based region distribution of semantic information in global space. It looks like a geographical fractal structure, which is providing the same structure of smart space data for application in any geographical position. All geographical information distribution is organized as a tree with orthogonal algorithm for navigation and search.

In this paper is spoken about smart system creation by combining the work of two platforms (Smart-M3 and Geo2Tag) using Smart-M3 agent (KP). Main difference of this work from previous ones is to use common platform for knowledge processing of Smart-M3 space.

The platform and its knowledge processors (agents) take on the whole job of collecting, analyzing and processing of space knowledge described by ontologies and deductions. When developing the platform agents we can use existing GIS-based or spatial ontology according to the Geo2Tag LBS-system or use your own, such as Geo2Tag system representation ontology.

The use and presentation of geo-data in the Smart-M3 platform adds a new property to the data as location in space and time that will allow to place objects in the real world as well will also give them the ability to search in a space such as a room or house space.

4 System Requirements and High-Level Design

There are several promising use-cases of GCSS:

- geographical markup of smart space data;
- search set reduction;
- search context rectification.

In next subsections high requirements and architecture of GCSS are discussed.

The main task of the agent - the Smart-M3 and Geo2Tag platforms union. One can say that the agent is an extension of the Smart-M3 platform, since the fully interoperable with Geo2Tag LBS platform, expanding the space with new data – geo-data.

The main user interaction with agent is to run it and specify the connecting settings for the Smart-M3 and Geo2Tag platforms and also monitoring and control of its operations. The agent then works independently checking receipt of new geo-data, producing a conversion in triplets and publish them into space.

4.1 High-Level Requirements

GCSS should implement effectively main features from both (smart space and geo-coding) type of systems, which are:

- providing interfaces for semantic data and access;
  Providing an interface for connecting to the platform and access to its semantic data. This feature is implemented in the Smart-M3 API (Qt, Python).
- distributed storage for semantic information;
  Smart-M3 platform has its own data storage, this storage may be replaced with a more stable and productive. Current stable Smart-M3 0.4.1 version uses Berkeley DB storage.
- interfaces for association semantic objects with geo-tags;
  Development of two-way geo-data conversion mechanism in the semantic objects (space triplets) and the creation of Geo2Tag platform ontology.
- spacial and temporal filtration.
  Development of the necessary algorithms for searching and filtering of smart space semantic data.

Also non-functional requirements should be taken in account:

- Performance – ability to work with big amount of semantic objects geo-tags; for some purposes we need to implement features 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.

At the moment, there are all functionality to work with the Smart-M3 platform and for development of its agents (KP), there is no mechanism for geo-data converting to the space triplets as well as algorithms for their search and filtering.

The latest versions of Smart-M3 and Geo2Tag platforms fairly stable, but have some defects such as those associated with security or performance of the Smart-M3 platform.

4.2 Layered Architecture

In our work we rely on existing middleware for smart-space environment and geo-tagging. As smart space infrastructure Smart-M3 is being used; for geo-coding we use Geo2Tag Platform. Selection of those platforms is caused the availability all source codes and solution maturity.

On Fig 1. High-level layered design for GCSS is presented. There are four levels of system.

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** – it contains smart-spaces (Smart-M3) and geo-coding (Geo2Tag) front-ends (FE) and their functionality; This level is responsible for data representation and processing for applications and services;
- **Integration** – this level contains components for translating geographical data (geo-tags) from Geo2Tag format to Smart-Space format and vice versa;
- **Domain engines** – level contains particular implementations of smart-space geo-coding middleware (in our case it is Smart-M3 and Geo2tag);
- **Data cloud backend** – optional components, which is being used for providing advanced services like off-line data (pre-)processing, storage for BLOB objects, indexing, caching etc.

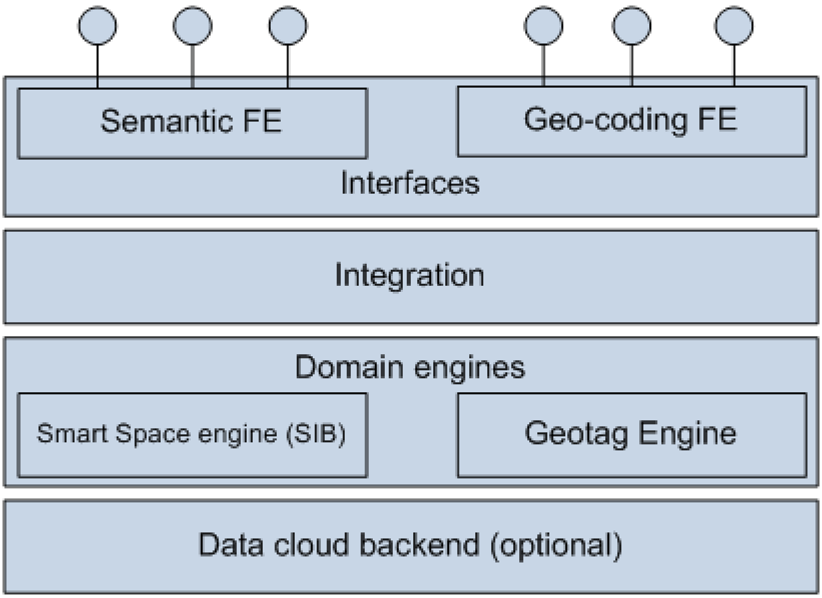

Fig. 3. Layered design of GCSS

The main suppliers of GCSS system functionality are a Smart-M3 platform and Geo2Tag LBS system. Details on this platforms and their main features can be found in [5, 6, 12].

Integration level is responsible for the platforms data conversion into one common format. For the system data storage responds GCSS Domain engines level, which is composed of the main data storage of the Geo2Tag and Smart-M3 platforms. To work with a large volumes of geo-data and increasing the overall performance of the system is planned to add the ability to work with Data cloud, as a repository for offline processing.

4.3 Location Based Engine

According to Smart-M3 architecture all knowledge of smart space is being presented as RDF triples. Set of RDF-tuples describing particular domain is an ontology. So, to present geo-coding data we need to use ontology also.

On the Fig 2. Geo2tag ontology is presented, it is defined according to Web Ontology Language Specification (OWL) as the tree of classes ans properties. The root of ontology tree is the class *User*. It contains all geographical data from personal geo-space. With this approach we can use personal and shared geo-tags, last can be implemented by introducing common user or group. Each instance of *User* class is connected with one or more *Channels* (Geo2tag terminology is used [6]). User has a relation *subsribesTo* so that follow different sets of geo-tags. The *Channel* could contain any amount of geo-tags by using property *containsA*.

Usially the *User* contains constant or rarely seldom changing properties about user credentials in geo-tagging system. This class can be presented according to FOAF.

The *Channel* includes "name" and "description" properties for identification of channel purpose and contains (through the property containsA) geo-tags itself, which are presented by time coordinates and data.

The main object of this Geo2tag ontology is a geo-tag (class Tag). Geo-tag is basic information, which manages and handles the Geo2Tag platform and it is the basic knowledge of the space, which describes the location of its subjects.

The size of a one geo-tag nearing 1K, Geo2Tag system allows you to store an unlimited number of data values, for it has developed special synchronization and optimization algorithms of system database [14]. After converting them into space triplets, the tag size will not increase, but now to work with them in the space will be used Smart-M3 platform storage. To improve the performance of data search is planned to develop the necessary algorithms for searching and filtering smart space data.

The property *data* plays significant role in integration mechanism. It contains set of identifiers and properties, which allow to specify objects or relations in smart-space.

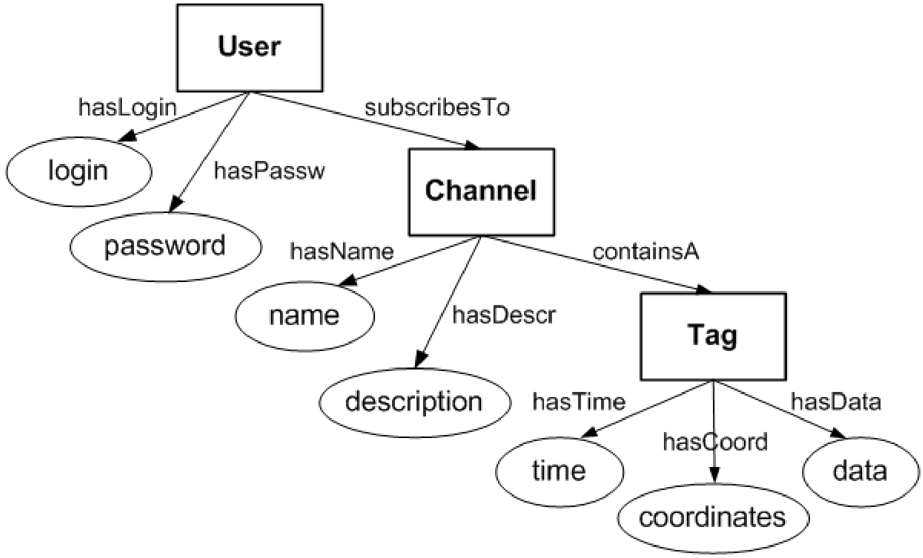

Fig. 4. Geo2Tag ontology in GCSS

On the integration level we introduce Geo2tag Agent. Geo2Tag agent is required for monitoring and synchronizing data between geo- and semantic spaces. Agent is Knowledge Processor (KP) in terms of Smart-M3.

Basic architecture of Geo2Tag KP is presented on Fig 3. There are next main components:

- Geo2Tag service handler;
- Smart-M3 handler;

Geo2Tag platform responsible for storing and processing of LBS-system geo-data, Geo2Tag-agent converts these data for use them in the space.

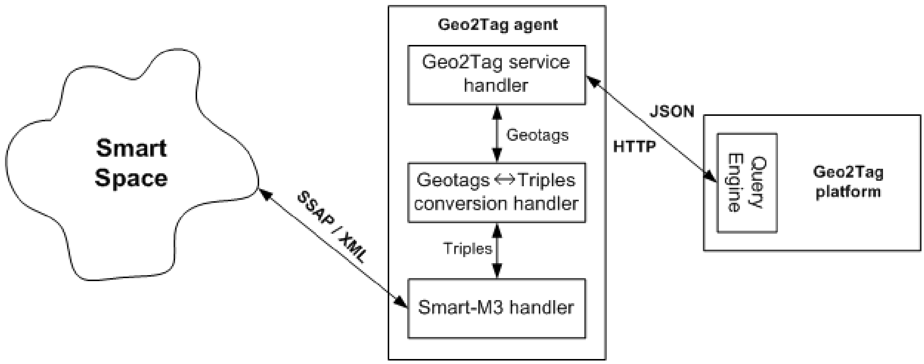

Fig. 5. Geo2Tag knowledge processor architecture

Let's discuss collaboration with Geo2Tag platform. Geo2Tag Platform is the centralized high performance geo-tagging database with dedicated server, which is provided REST API for access geo-tags. All communications could be implemented only over HTTP protocol, which could be a reason for performance issues. All geo-tagging data is presented as JSON objects.

As mentioned earlier, the basic data format of the Smart-M3 platform – RDF-triples <S, P, O>. To work with Geo2Tag platform geo-data it is necessary to convert them to the data format of the Smart-M3 platform, for these purposes serves a separate component of the Geo2Tag-agent – «Geotags <-> Triples conversion handler», whose main task is to convert the geo-data to the space RDF-triplets.

Geo-tag consists of a tuple <t, L, B, H, data>, where

- t – time;
- L, B, H – coordinates;
- data - text data (~ 1K).

From this we can easily generate triplet of <time, coordinates, data> type, which will describe the space geo-data. Geo-tags can also be combined into channels and users can subscribe to the tags channels. Which is easily described with triplets, using the agent ontology.

Smart-M3 handler retrieves JSON objects and translated them into Smart-M3 triples, and handles operations with smart space such as subscribing, publishing. There are several methods for ontological data processing here:

- ontological model, where all data is being kept in agent memory as RDF graph similar to RDF graph of smart space.
- object-oriented model, where all RDF data is transforming from RDF triples into objects with properties and methods. And reverse transformation is being used only for publishing data into smart space. We suggest to use this type of geo-tagging data processing.

One of the disadvantage of data representation into Smart-M3 is inability of presenting media data. It cannot be presented as triples. Geo2Tag architecture is designed for supporting such king of data. This features could be used for extending Smart-M3 functionality. Main problem should be discussed is a privacy and data protection.

5 Conclusion

It this paper we proposed high level design of smart-space and geo-coding middleware integration. This integration could be made by using special Knowledge Processor, which monitors both spaces and translates data from one to another and vice versa. There are still open question for future development: overall system performance, effective object monitoring, temporal and spatial filtration, integration with media objects.

Acknowledgments. The presented work is done as a part of the research and development work done in the project funded by grant #11555p/20929 of FASIE Foundation.

References

1. Floch, J., Angermann, M., Jennings, E., Roddy, M.: Exploring Cooperating Smart Spaces for Efficient Collaboration in Disaster Management. In: Proceedings of the 9th International ISCRAM Conference – Vancouver, Canada (2012)
2. Nabian, N., Ratti, C., Biderman, A., Grise, G.: MIT GEOblog: A platform for digital annotation of space for collective community based digital story telling. In: 3rd IEEE International Conference on Digital Ecosystems and Technologies, pp. 353–358. IEEE, Piscataway (2009)
3. Rishede, J., Man, T., Yiu, L.: Effective Caching of Shortest Paths for Location-Based Services, SIGMOD 2012, Scottsdale, Arizona, USA (2012)
4. Kolomvatsos, K., Papataxiarhis, V., Tsetsos, V.: Semantic Location Based Services for Smart Spaces. In: 2nd International Conference on Metadata and Semantics Research (MTSR), Corfu, Greece (2007)
5. SourceForge, Smart-M3, `http://sourceforge.net/projects/smart-m3`
6. Honkola, J., Laine, H., Brown, R., Oliver, I.: Cross-Domain Interoperability: A Case Study, Nokia Research Center, Helsinki, Finland (2009)
7. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (2001)
8. Oliver, I., Honkola, J.: Personal Semantic Web Through a Space Based Computing Environment Middleware for Semantic Web 08 at ICSC 2008, Santa Clara, CA, USA (2008)
9. Oliver, I., Honkola, J., Ziegler, J.: Dynamic, Localized Space Based Semantic Webs. In: WWW/Internet Conference, Freiburg, Germany (2008)
10. Boldyrev, S., Oliver, I., Honkola, J.: A mechanism for managing and distributing information and queries in a smart space environment, MDMD 2009. Milan, Italy (2009)
11. Roffia, L., D'Elia, A., Vergari, F., Manzaroli, D., Bartolini, S., Zamagni, G., Cinotti, T.S., Honkola, J.: A Smart-M3 lab course: approach and design style to support student projects. In: Balandin, S., Ovchinnikov, A. (eds.) 8th FRUCT Conference of Open Innovations Framework Program FRUCT, pp. 142–153. Saint-Petersburg State University of Aerospace Instrumentation (SUAI), Lappeenranta (2010)
12. Bezyazychnyy, I., Krinkin, K., Zaslavskiy, M., Balandin, S., Koucheravy, Y.: Geo2Tag Implementation for MAEMO. In: 7th Conference of Open Innovations Framework Program FRUCT, Saint-Petersburg, Russia (2010)
13. Dearle, A., et al.: Architectural Support for Global Smart Spaces. In: Chen, M.-S., Chrysanthis, P.K., Sloman, M., Zaslavsky, A. (eds.) MDM 2003. LNCS, vol. 2574, pp. 153–164. Springer, Heidelberg (2003)
14. Zaslavsky, M., Krinkin, K.: Geo2tag Performance Evaluation. In: Proceedings of the 12th Conference of Open Innovations Association FRUCT and Seminar on e-Travel, Oulu, Finland (2012)