Mediator Based Approach for Smart Spaces Integration

Yury Korolev, Kirill Yudenok
Saint-Petersburg Electrotechnical University
Saint-Petersburg, Russia
{yury.king, kirill.yudenok}@gmail.com

Abstract

The article considers the main methods of data integration for a variety of storage and knowledge representation systems, namely the integration of ontologies, linked data and smart spaces. The main attention is paid to the integration of smart spaces using a special software component – a mediator agent. The paper describes a formal model of the mediator agent and integration procedures based on the descriptive logic. The architecture of the mediator agent is designed for the smart spaces integration scenario based on mapping rules.

Index Terms: Smart Spaces, Data Integration, Smart-M3, Ontology, Linked Data.

I. Introduction

A smart space is a virtual, service-centric, multi-user, multi-device, dynamic interaction environment that applies to a shared view of resources [1].

The smart spaces forms a smart environment, which is “able to acquire and apply knowledge about its environment and to adapt to its inhabitants in order to improve their experience in that environment” [2]. The information is dynamically shared by multiple heterogeneous participants (humans and machines), allowing each user to interact continuously with a surrounding environment and services will be continuously adapted to the current needs of the users [3].

Smart space is defined [3] as a three-tuple containing a unique name (n) for the smart space, the information (I) provided in the form of RDF graph and a set of rewrite inference rules (r) over that information:

$$S = \langle n, I, r \rangle .$$

Smart spaces require a software infrastructure that turns the constituting spaces into programmable distributed entities. Smart-M3 provides such an infrastructure to use a shared view of dynamic knowledge and services within a distributed application. The Smart-M3 platform is the only general-purpose open-source smart spaces platform currently available. A more detailed description of the architecture and components of the Smart-M3 platform you can find in the following references [2, 4].

Smart Spaces paradigm requires the ability to integrate information between different applications and services from embedded systems to the Web. With an advent of various smart systems (such as Smart Conference System, Home Smart Space, Car Smart Space), inevitably will arise the problem of combining these spaces in a Global Smart Space.

The section II defines methods and algorithms of ontology integration. Section III and IV describes principles of Linked Data and Smart Spaces integration. Formal definition of the integration procedure given in section V. The architecture of mediator agent for the integration scenario and Smart Conference and blogging service is described in section VI, VII accordingly. Here too provides an example of rules for mapping Smart Conference report in SmartScribo post. In future work, we plan to develop a mediator-based framework for automated integration of multiple smart spaces based on the mapping rules. The conclusion summarize the major works that we have done in the course of this work.

II. Ontology integration

A. Ontology integration methods

Today the ontological description of the subject areas is one of the most effective means of the information sources semantics representation. With increasing amounts of integrable data, in particular – if their sources are diverse, there is a problem of integration of various ontologies for semantic representation of the generalized information. In general, ontologies integration is usually determined as the process of finding similarity of two ontologies A and B, and as a result, creation of a new ontology C, uniting and matching semantic representation of the source ontologies. As a result, two systems based on ontologies A and B are able to interact with each other using the ontology [5].

Current methods of ontologies integration are classified into two types, in which:

- new ontology is used instead of integrable;
- integrated and original ontologies are shared.

Methods of the second type are more flexible because you can save and use structure of the existing ontologies in the future. At the same time, in case of a single integrated ontology, it is necessary to include all terms in the source ontologies. When you combine two terms of ontology, there are five variants of relative relations of classes, resulting in five different classes of operations on the association stage[6]:

1. classes are equivalent and represent the same concept in the ontology, therefore, they must be “glued” to one;
2. one class of ontology is a generalization of another class of the ontology, such classes should be submitted as a class and subclass, respectively, with identical attributes must be removed from the subclass, as they will be inherited from the superclass;
3. one class of ontology is a refinement of another ontology class, these classes should be submitted as a subclass of the class, respectively, with identical attributes must also be removed from the subclass, as they will be inherited from the superclass;
4. classes of two ontologies are equivalent to some, they represent a similar concept, that is must have a common superclass, which is a generalization (the superclass is not present in any of the source ontologies) with matching attributes should be removed from the subclass, as they will be inherited from the synthesis of a class;
5. classes are different, they must be copied to the resulting ontology.

Obviously, the method of using a single monolithic ontology instead of integrable entails difficulties in restructuring the existing relations of terms and a resolution of semantic conflicts. For this reason, it can be used only in cases where a set of integrable ontologies is known in advance and its expansion is not assumed [7].

Sometimes the ontology integration is divided into two types, depending on the number of changes that need to be done in order to get the resulting ontology of the source [5]:

1. Compliance. This operation is a mapping of concepts and relations of one ontology to another. At the same time in one ontology can exist concepts without their equivalents in another. To bring the ontology into compliance in resulting ontology superclasses and subclasses can be added.
2. Unification. This operation is a one correspondence of all concepts and relations in two ontologies, which allows any process of inference expressed in one ontology display to the equivalent output process in another.

The main method of ontologies integration used in modern systems of engineering ontologies [8] is to find disjoint classes based on the opinions of the system or solutions of expert. Intersecting classes are added to the resulting ontology and their attributes are multiple-logical combination of initial classes attributes. That is the correct transfer of the original attributes of classes in the resulting class is the main task of the integration procedure.

B. Algorithm of ontologies integration

Algorithm of ontologies integration are traditionally divided into three relatively independent stages [6], each of them incorporates the results at the previous step:

1. *The degree of equivalence between the two classes of ontologies*.
   This step is often performed by the expert as well as procedure for determining the semantic relationship between classes of different ontologies is poorly formalized and has no universal solutions. The existing formal approaches are based on verbal heuristic methods for the binding of ontological concepts in different contexts [9].
   There are four groups among them [10]:
   - linguistic (terminological, lexical);
   - structure;
   - statistics (extensional);
   - logical (formal, semantic).
2. *Componentwise union sets of classes and attributes of the source ontologies*. This stage actually produces the union of ontologies based on the degree of classes equivalence, as was calculated in the previous step. This stage is usually fully automated, so it doesn’t require making decision regarding the data semantics. Choice of an algorithm of ontologies integration has influence on the degree of intelligence of the whole association procedure. In general the technique ontology integration includes the following operations:
   - integration of equivalence classes;
   - copy the properties of both the original classes;
   - copy links both source classes;
   - binding superclasses of both source classes (if they are present in the resulting ontology);
   - binding subclasses of both source classes (if they are present in the resulting ontology);

  - integration of the equivalent properties of classes;
  - integration of the equivalent relations between the classes;
  - copy classes to the generated ontology;
  - copy properties to the generated ontology;
  - copy links to the generated ontology;
3. *Check the correctness of the association result*. This step can also be fully realized only by the expert, but for automation of this procedure may be determined the formal criteria of consistency of the ontology.

III. LINKED DATA INTEGRATION

By publishing data in RDF model in accordance with common conventions, data publishers enable Linked Data applications to incrementally expand their knowledge about Semantic Web resources. In the body of Linked Data published thus far, owl:sameAs is increasingly used to provide declarative semantics for aggregating distributed data [11]. That is, machines can merge resource descriptions if the resources described are linked with owl:sameAs. The rising use of owl:sameAs can be observed in many important Linked Data datasets such as DBpedia[1], Freebase[2], GeoNames[3].

The owl:sameAs property is a part of the Web Ontology Language (OWL) [1] and frequently used to support Linked Data integration via declaratively interconnecting “equivalent” resources across distributed datasets [12]. Prior to the rise of owl:sameAs, the rdfs:seeAlso property was heavily used in linking Friend of a Friend (FOAF) data: it links from one FOAF document to another where additional descriptions about the resource can be found.

The equivalence relationship represented by owl:sameAs is often context-dependent, and is accurate only in the context of one application [13]. Therefore, the use of owl:sameAs in Linked Data may combine context-dependent descriptions provided in different data sources.

If we connect the URIs using owl:sameAs, an OWL reasoner can infer the integration of two datasets, even if a unique URI refer to the same entity. It still remains a problem that needs solving.

A further question on owl:sameAs publishing is that of how to deal with third-party asserted owl:sameAs relations. A third-party asserted owl:sameAs relations could be used to facilitate linked data integration. Indeed, sameas.org play this role and collect millions of third-party asserted owl:sameAs relations [11].

The empirical usage of owl:sameAs only captures the equivalence semantics on the projection of the URI on social entity dimension. In this way, owl:sameAs is used to indicate partial equivalence between two different URIs, which should not be considered as full equivalence. Knowing the dimensions covered by a URI and the dimensions covered by a property, it is possible to conduct better data integration using owl:sameAs.

There are many examples where the likely meaning of an owl:sameAs assertion in Linked Data is intended to be the official semantics as defined by OWL. Nonetheless, we cannot assume that it is never used with the intended semantics of absolute identity in mind. Since suitable alternatives to owl:sameAs do not exist (or are rarely used in

[1] http://dbpedia.org/ – DBpedia
[2] http://rdf.freebase.com/ – Freebase
[3] http://sws.geonames.org/ – GeoNames

practice), a Linked Data application is forced to make a choice with respect to each owl:sameAs link it encounters.

In [11] propose several conditions (rules) for integrating and merging information from the URIs in an owl:sameAs network:

- Complementary descriptions: if the associated descriptions of the URIs linked by owl:sameAs are orthogonal, then they can safely be merged. In linked data consumption, this kind of URIs should be dereferenced to collect the most complete description of the resource.
- Alternative descriptions: if the associated descriptions of the linked URIs are asserting different values for the same property, conflicts may occur when users expect a unique value from the property.
- Reconcilable descriptions: if the associated descriptions of the linked URIs are neither fully orthogonal nor fully alternative, users may have more options. They can simply alter the portion of conflicting descriptions in an application-specific way, by taking the context of the descriptions into account.
- Redundant descriptions: if the associated descriptions of the linked URIs forms a subset (or implication) relations, only the URI with broader coverage needs to be dereferenced.

Linked data integration using the property owl:sameAs has its own characteristics, both positive and negative. The main drawback of owl:sameAs, that then we combining data, OWL reasoner may still be wrong to interpret the relationship that will lead to failure of integration, but there are certain rules that can help in solving this problem. Alternative solution to the property owl:sameAs do not exit, so you have to tolerate and look for additional solutions.

IV. SMART SPACES INTEGRATION

In Section II and III, we describe the basic approaches of ontologies and linked data integration, it is obvious that smart spaces integration should be based on similar principles. In the context of integration problem, the basic difference between smart spaces and systems, for which the above methods have been developed, is the dynamic nature of smart space. In the smart space, new objects can appear and old objects can be deleted. Therefore it is not simply enough to map any objects of one ontology to objects of other ontology, the objects must be synchronized between different spaces. In this case, we mean that the synchronization is the process of constructing the correspondence of unique identifier of integrable objects from different smart spaces. This operation is also very useful for the integration of hierarchical dynamic objects.

Traditionally, the problem of data integration is solved by delegating to a special software mediator. In the case of smart spaces integration, this approach has already been discussed in [14]. If the Smart-M3 terminology is using, such software mediator represents the knowledge processor (KP). Further in this article this KP will be named a mediator agent.

The model of the mediator agent for smart spaces integration for n information sources $\Delta_1 ... \Delta_n$ is formally defined as:

$$\Psi = \left\langle \{ \mathrm{O}_{\Delta i} \}_{i=1..n}, R, S \right\rangle,$$

where: $\{ \mathrm{O}_{\Delta i} \}_{i=1..n}$ – a finite set of ontologies (RDF data) of smart spaces or integrable information sources, consistent and expressed in the language of descriptive logic.

Physically, the information represented as a set of facts in some way is stored in the data source i, and is available through the query interface corresponding to the adapter.
R – finite set of mapping rules between smart spaces ontologies.
S – structure of correspondence of the UIDs of integrable objects. As this structure may be a canonical matching ontology, projected by the expert proceeding on the basis of mapping rules.

The integration procedure implementing by mediator agent can vary greatly depending on the complexity and heterogeneity of integrable sets. In simple cases, when it is required to match the equivalent objects, such as linked data integrating, it is enough to define the bridge rules for mapping some objects to another. We give a brief methods classification of smart spaces integration. There are two basic approaches of the integration methods classification on the basis of the intellectuality criterion:

1. *Depending on intellectuality of integrated systems.* In the simplest case integrated systems do not know anything about each other. The mediator agent integrates systems using the mapping rules. In a case of more complex scenario, the system manages the integration procedure by sending appropriate notification.
2. *Depending on intellectuality of integration procedure.* In the simplest case, the integration is based on the mapping rules defined by the expert. In more complex usage scenarios, the mediator agent generates mapping rules in line with the current context by itself.

V. RULE-BASED MAPPING FORMALISM

We use the descriptive logic for a formal definition of the integration procedure. There are two major formalisms of descriptive logic: concept and role. In other branches of mathematical logic, they correspond to the concept of "one-place predicate" (or set, class) and the "two-place predicate" (or a binary relation). Intuitively, the concepts used to describe certain classes of object, such as "People", "Female", "Cars". Roles are used to describe binary relations between objects, for example, on the set of people is the binary relation "X is_parent_for Y", and between humans and machines is a binary relation "X have Y", where as the X and Y can substitute arbitrary items.

Let *F* be a set of predicates and *V* be a set of predicate variables; $f(v_1,...,v_n) \in F$ is a n-ary predicate; $v_1,...,v_n \in V$ is a set of terms. Then an expression of the form $c(v)$ is an atomic concept, and $p(v,w)$ is an atomic role. Let O_1 and O_2 be two ontologies. A class mapping is defined through a set of mapping rules of the similar form:

$$\beta_1(w_1) \Leftarrow \alpha_1(v_1),...,\alpha_n(v_n),$$

where $\alpha_1(v_1)$ – an atomic concept or an atomic role of the source ontology O_1, and $\beta_1(w_1)$ is an atomic concept and atomic role of the target ontology O_2. In the case of heterogeneous mapping, that use Skolem functions to express the semantic relationships between two ontologies, for example, when information represented as a class in one ontology and an object property in the other ontology [15].

Although the descriptive logic is elegant and useful tool, logic programming also provides important opportunities for knowledge management. In particular, it is well tuned to the procedural handling of objects, that is not available for descriptive logics. Therefore, the idea of combining descriptive logics and logic programming is attracts considerable interest [16].

Mapping rules using the notation of logic programming, can be written as follows:

$$<\beta_1>:-<\alpha_1>,...,<\alpha_n>.$$

The left side of the rule – <head> – is a formalized description of the conclusion, while the right side of the rule – <body> – formal description of the conditions determining the truth of the conclusion.

VI. MEDIATOR-BASED AGENT ARCHITECTURE

Let's describe the architecture of mediator agent for the integration scenario, when integrable systems function independently of each other and mapping rules are defined by the expert. The mediator architecture (Figure 1) consists of four basic modules: the module of interaction with information sources, data synchronization module, the mapping module and the coordination module. We describe the function of each module in more detail.

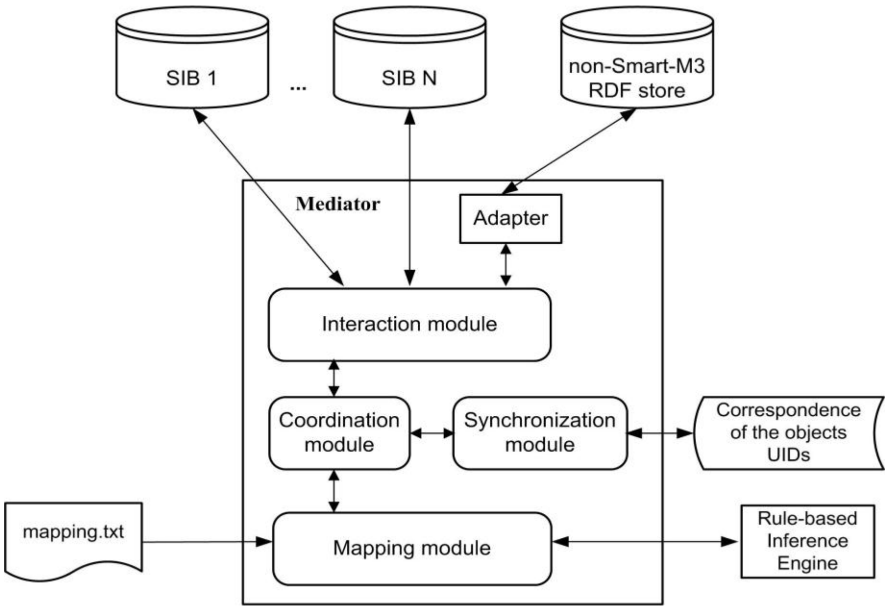

Fig. 1. Mediator-based agent architecture

The module of interaction with information sources interacts with semantic information brokers (SIBs) or other RDF data stores. In the case of third-party RDF storages, it is required to implement a special adapter to communicate with a specific information source API. This module interacts with the SIB using the Smart Space Access Protocol (SSAP) and allows four basic operations on the data (query, insert, remove, update), and operation of the subscription to data change. The synchronization module generates and stores a structure of correspondence of the UIDs of integrable objects, and also provides information about the synchronization process: how many objects were integrated. The mapping module loads the mapping rules defined by the expert from a text file, translates them into a logical language (AnsProlog), and based on the inference engine performs the generation of new objects based on the original. The coordination module is the main module of the system coordinating the work of all other modules.

VII. INTEGRATION OF SMART CONFERENCE SYSTEM AND BLOGGING SERVICE

The idea of integration of Smart Conference System[4] and blogging service already have been discussed in [14], so we are not going to dwell on it, just remind the basic idea. The mediator agent[5] is designed to comment presentations for Smart Conference System. Further in this article this mediator agent will be named a SC-BS mediator. The basic idea is to provide conference participants with communication tools, which enable the exchange of comments, allow to address the speakers with questions and get answers before, during and after their performances. The SC-BS mediator allows to remotely located people monitor the conference via Internet and participate in the presentations discussions. This is achieved in such a way: the post is created for each report on LiveJournal server and remote participants can add comments to it. The task of the SC-BS mediator is the synchronization of posts and hierarchy of LiveJournal comments with information from the conference Smart Space. Speakers and direct participants of the conference can participate in the discussion using a mobile multi-blogging system SmartScribo[6]. Thus, a LiveJournal server and the conference Smart Space will store the identical to-date information, and both direct participants of the conference and remote participants will have access to it. The architecture of the current version of the SC-BS mediator is presented in Fig. 2.

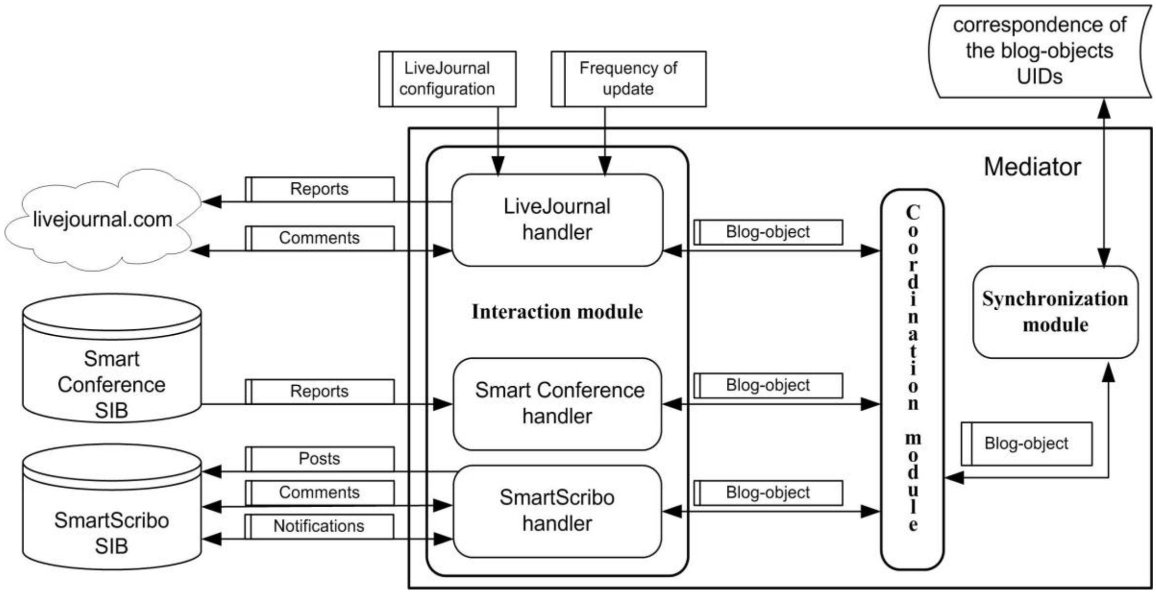

Fig. 2. The architecture of mediator agent for integration Smart Conference and blogging service

Currently, the mapping rules are hard-coded in the SC-BS mediator in the form of C++ logical structures. Further we plan to represent the mapping rules in AnsProlog language and execute them using a bunch of tools: ssls + smodels + lparse.

In the conclusion we result a rule for mapping Smart Conference System report in SmartScribo post in AnsProlog notation (Fig. 3).

[4] http://sourceforge.net/projects/smartconference/ – Smart Conference System
[5] https://github.com/OSLL/scblog – Scblog project
[6] https://gitorious.org/smart-scribo – SmartScribo application

```
newReport(name,title,keywords) :-
 "is"(userId, name),
 "presents"(userId, presentationId),
 "Title"(presentationId, title),
 "Keywords"(presentationId, keywords).

createPost(title, text) :-
 newReport(name,title,keywords),
 generateText(text,name,keywords).

3{i(id,"rdf:type","http://www.cs.karelia.ru/smartscribo#Post"),
 i(id,"http://www.cs.karelia.ru/smartscribo#title",title),
 i(id,"http://www.cs.karelia.ru/smartscribo#text",text)}3 :-
    createPost(title, text),
    generateUID(id).
```

Fig. 3. Rules for mapping Smart Conference report in SmartScribo post

VIII. FUTURE WORK

Further, we plan to develop a mediator-based framework of automated integration of multiple smart spaces based on the mapping rules. The main research problem for the implementation of this framework is investigation and development of declarative domain model language based on some descriptive logic dialects, which describes smart spaces integration the most complete and conveniently. It is also necessary to develop a method of translation of this domain model language to the set of rules of logical language. After implementation of this step, we plan to investigate any ways to automate more complex problem of integration, when the mapping rules are not set by the expert, but automatically generated based on the current context of global smart space.

IX. CONCLUSION

The article considered the main methods of data integration for a variety storage and knowledge representation systems, namely the integration of ontologies, linked data and smart spaces.

The main attention is paid to the integration of smart spaces and the creation of a special software component (mediator agent), designed to address this problem. The paper describes a formal model of the mediator agent, integration procedure based on the descriptive logic and architecture for the smart spaces integration scenario. An attempt was made to provide mapping rules using logic programming tools.

The mediator agent consists of several modules that perform the basic functions to interact with data storage's, synchronization integrable objects, working with the mapping rules.

The idea of mediator-based agent for the integration of Smart Conference and SmartScribo was successfully demonstrated at the 9th and 10th International Conference FRUCT, which was held in Petrozavodsk (Russia) and Tampere (Finland) respectively.

This article provides a basis for further investigation of smart spaces integration problem and the development of an automated integration system.

REFERENCES

[1] Oliver, J. Honkola, and J. Ziegler, “Dynamic, localised space based semantic webs,” in Proc. IADIS Int’l Conf. WWW/Internet 2008. IADIS Press, Oct. 2008, pp. 426–431.

[2] Dmitry G. Korzun, Sergey I. Balandin, Vesa Luukkala, Petri Liuha, Andrei V. Gurtov "Overview of Smart-M3 Principles for Application Development," in Proc. Int'l Conf. Artificial Intelligence and Systems(AIS/CAD 2011).
[3] Oliver, S. Boldyrev, "Operations on Spaces of Information", Semantic Computing, 2009. ICSC '09. IEEE International Conference
[4] J. Honkola, H. Laine, R. Brown, and O. Tyrkko, "Smart-M3 information sharing platform," in Proc. IEEE Symp. Computers and Communications, ser. ISCC '10. IEEE Computer Society, Jun. 2010, pp. 1041–1046.
[5] Masloboev A.V., Lomov P.A. Intelligent Systems. 2010. vol.14. No.1. pp. 4-10. (rus)
[6] Palagin A., Mikhailyuk A., Velichko V., Petrenko N., "By the integration of domain ontologies," Information Models of Knowledge, ITHEA 2010. pp. 69–85. (rus)
[7] Wache H., and other. "Ontology-Based Integration of Information," A Survey of Existing Approaches. in Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing, 2001.
[8] Ovdey O. M., Proskudina G.Y., "Overview of ontology engineering tools," Digital libraries - Moscow: Institute of the Information Society. 2004. vol.7. No. 4. (rus)
[9] Ckvortsov N., "Use clarifying concepts in solving problems of ontologies manipulating," Proceedings of the Ninth Russian Scientific Conference. "Digital Libraries: Advanced Methods and Technologies, Digital Collections". RCDL 2007. pp. 225-229. (rus)
[10] Kudryavtsev D.V., "Practical methods of mapping and integration of ontologies," in Proceedings of the 11th National Conference on Artificial Intelligence with International Participation (CAI-08), a seminar "Knowledge and Ontologies". Dubna, Russia. M.:URSS. 2008. No 3. pp. 164-173. (rus)
[11] Li Ding, Tim Finin, Joshua Shinavier , Deborah L. McGuinness. "owl:sameAs and Linked Data: An Empirical Study," In: Proceedings of the WebSci10: Extending the Frontiers of Society On-Line, April 26-27th, 2010, Raleigh, NC: US
[12] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, and L. A. Stein. "OWL Web Ontology Language Reference," W3C Recommendation, February 2004. www.w3.org/TR/owl-ref.
[13] Jaffri, H. Glaser and I. Millard. "Uri disambiguation in the context of linked data," In Proceedings of the 1st International Workshop on Linked data on the Web, April 2008.
[14] Dmitry G. Korzun, Ivan V. Galov, Alexey M. Kashevnik, Nikolay G. Shilov, Kirill Krinkin, Yury 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.
[15] Vania Maria Ponte Vidal, Jose Antonio Fernandes de Macedo, Joao C. Pinheiro, Marco A. Casanova, Fabio Porto. "Query Processing in a Mediator Based Framework for Linked Data Integration," IJBDCN, 2011, pp. 29-47
[16] Grosof, B.N., Horrocks, I., Volz, R. "Description Logic Programs: Combining Logic Programs with Description Logic," Proc. of the Twelfth International World Wide Web, May 2003, ACM, pp. 48-57.