

Yu. V. Bilodray; V. I. Mesiura, Cand. Sc. (Eng), Assist. Prof.

COMMUNICATION SYSTEM OF PROJECT AND OPERATIONAL MANAGEMENT IN SOCIAL MEDIA

The paper proposes a new type of communication systems of project and operational management that assumes the presence of central communicational environment based on social media, common for humans and connected third-party systems. The approach makes use of properties of social media as complex adaptive systems that naturally aggregate relevant data, have significant integration and mobile potential as well as microblogging messaging that facilitates simplified semantic annotation. Such design enables a new economically effective means of communicating heterogeneous current and historical information for its combinative analysis and supporting of directed group activities. The system is the platform for integration of products of third parties on the basis of the popular business model of social media, which significantly increases the number of possible solutions, usually complex and not available to wide circles of the users. The design elaboration is aimed at supporting the activities of distributed non-profit, low budget, scientific and educational organizations, public projects and small businesses.

Keywords: social media, complex adaptive systems, CAS, Enterprise 2.0, controlled language, microblogging, nanoformat, semantic representation, sequential approach, management

Introduction and problem statement

Despite growing interest to social media as the environment of accumulating knowledge capital and as the instruments of organizational activities, insufficient progress of their analytical capabilities for supporting directed group activities in the nearly-real-time mode should be noted. One of the main reasons is complexity of NLP/IE technologies and semantic representation of information. The work suggests an approach to creating the systems on the basis of social media with the set of functions of “classic” corporate systems of the project and operational management. Such systems should be based on social media, that naturally aggregate relevant data as complex adaptive systems [1], have features of Emergent Social Software Platform ESSP / “enterprise social software” [2]. It is assumed, that the information is accumulated with the help of a special agent and translates sequences of user messages regarding activities, events, physical and abstract substances into an integral semantic representation with the least possible complexity of NLP/IE processes and satisfactory accuracy, and stores the model in a special repository. The system must have a rapid descriptive mechanism of filtering the parts of the model with the possibility of data aggregating as well as getting individual data components and visualizing the representative part of the model on the screens of computers and mobile devices. In this way a dynamic process is set up, where participants supplement the knowledge base, and the system maintains integral knowledge representation, optimizes it and routes to the participants, increasing effectiveness of their work [3].

Main part

General schema of the solution proposed. Fig. 1 shows a general schema of the system. Data intake (1A) occurs inside the popular microblogging environment where participants and third-party systems exchange current information as well as supplement and use historical information, utilizing pre-annotation of the text with nanoformats and the controlled language [4]. Using visual constructing (1B), the users form information dashboards of the necessary semantic information as sequences of filters and store them (2B). The information dashboards are sets of SPARQL or SeRQL queries to the semantic repository. Complexity of creating queries is hidden in the mechanism of their auto-generation with the help of sequential approach analog [5]. The partially structured taken in data (1A) are processed by NLP and IE methods (2A) and translated into RDF triples which are stored in the semantic repository (3A). RDF graphs are created on the basis of the ties stipulated by

OWL ontologies. The ontologies play the role of schema not only for searching information elements and their connections, but also govern the process of translating information into RDF as dynamic applied area-specific vocabularies of the controlled languages. The information dashboards are generated periodically or triggered by events (5A) being the sets of SPARQL or SeRQL queries to the semantic repository (4A). The information dashboards are presented to the users in the form similar to that of their formation (6A).

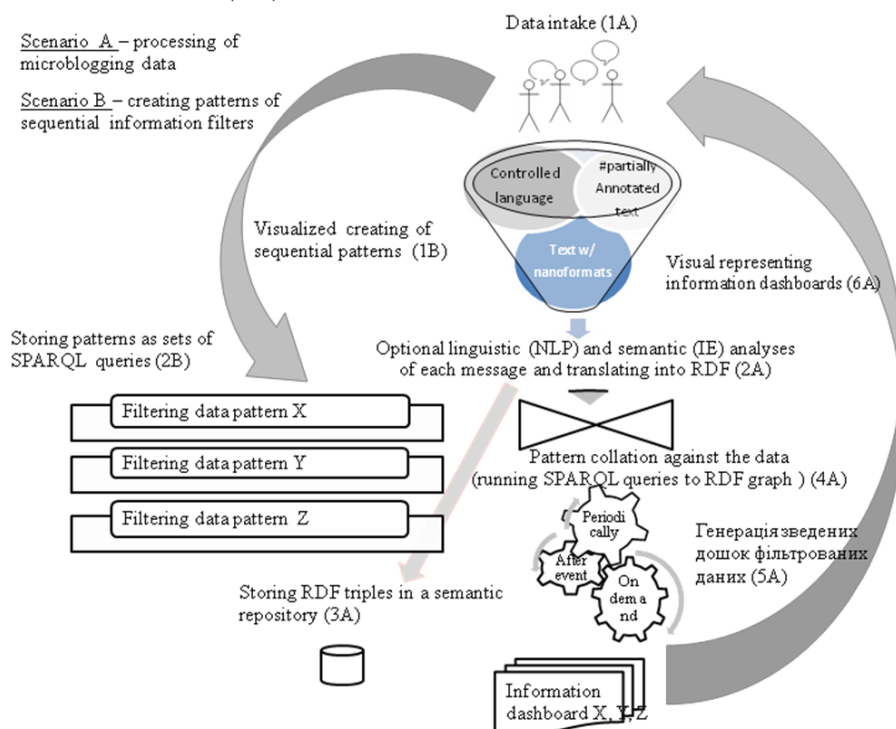


Fig. 1. General schema of the system for supporting and controlling group activities on the basis of microblogging

Connecting to microblogging platforms. The schema of connecting to microblogging is shown in Fig. 2. The main actors are: the users (humans), the users (third-party systems), the user – Information Router (IR) and the system of the project / operational management itself (“system” hereafter). All of them are the subjects of two-directional communication (to one another) and multi-directional (in the common context) information exchange (demonstrated as <<follows>>). All the elements are registered in a standard way as the users of the microblogging platform and the system. IR is a part of the system but is represented as a separate element for demonstrating its central functional role – it is the single point of connection of the microblogging environment and the system. In relation to microblogging and communication process all the elements are connected in peer manner and in the same way technically. Connections between the users and IM are compulsory while connections between the users are optional and desirable. Despite IM being a functionally central unit, the communication system is a peer-to-peer one because microblogging reveals the qualities of Pull communication instrument and Interactive communication at the same time [6]: IM is a “listener” of the timelines of the users it “follows” like other users do. Also IM publishes its own messages that can be read by other users at their discretion. The set of the elements connected in such a way forms the space of communication of operational or project group activities.

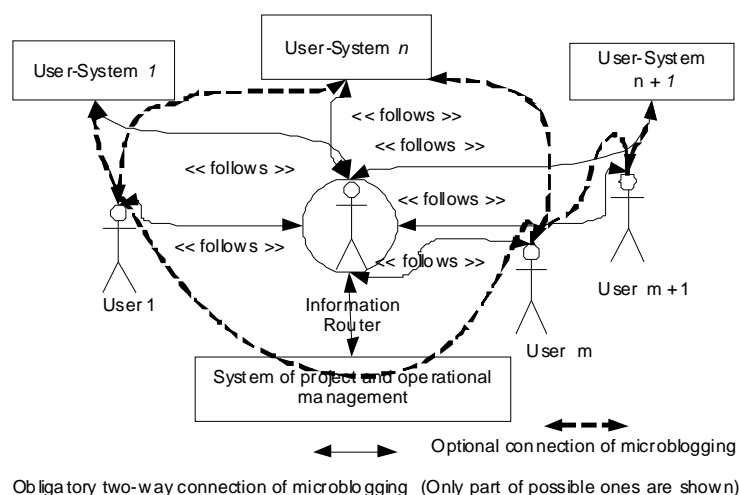


Fig. 2. Connecting of the system users to the microblogging platform

Regarding information security, most of the general-purpose microblogging platforms support private user profiles and information of such users cannot be read by another not authorized user. If all authorized users have private profiles, the entire system will be closed for live access from outside, however information stored by such microblogging platforms usually has lower level of security than applicable for enterprises. This can be a problem for organizations that operate with sensitive information.

Table 1

Scenarios of using a microblogging platform in the system

Usage scenario	Explanation
User executes the "follow" query to the Information Router (IR)	Action of the work space formation. Actors: all users that take part in the information exchange
IR confirms the "follow" query	Action of the work space formation. Actors: IR, users who are allowed to perform the "follow" function
User publishes information about activities or knowledge	Actors: a human, third-party system, IR
User reads information about activities or knowledge	Actors: a human, third-party system, IR (all except the last – optionally, the last – obligatory)
User sends a command to perform an action or a query for information to another user	Actors: a human, third-party system, IR
User responds to the command or query	Actors: a human, third-party system, IR

Collecting the messaging data. With the help of the system it is possible to gather any kind of applied area-specific, project-specific information (or information of general character, if it is supported semantically): status of task performance, logistics, problems and solutions, information for further calculation, for example, EVM analysis [6], sensor data, task assignments to humans and systems as well as usage of historical information. Examples of possible input messages:

Work package WP2 has EV=1.1 and AC = 0.9;

Project XYZ has workpackage WP1 and workpackage WP2. WP2 is half-complete; (1)

An accident on crossing of Street X and Street Y; Height of tidal wave is 1.5m near the east coast.

At the input of the system (Fig. 1) there should be messages qualified for interpretation, and the system must have data of the classes of substances and of their properties for such interpretation. Two types of input messages are considered within the context of this work: 1) messages with controlled language CLIE; 2) messages with nanoformats like Twitter Data [7]:

$$\begin{aligned} &\#WP2 \text{ has new CR } \#change_data_format. \text{ Find details here } < URL >, \\ &\#WP2 \$hasNewCR \text{ Change data format. Find } \$details \text{ here } < URL >. \end{aligned} \quad (2)$$

The expressions (2) show that text with Twitter Data has the level of descriptive complexity similar to that of the text with the controlled language CLIE and is analogous to usual microblogging messages. Such messages are already semantically annotated in the formats compatible with RDF and OWL.

Semantical correspondence of nanoformats and RDF representation. Twitter Data format is a variety of nanoformats that enable embedding semantic definitions into the microblogging message with the help of opening and closing symbol “\$” (or the end of the message). Semantic data structured as simple pairs *name-value* of the character sequences, tuples *\$name1 value1 \$name2 value2* or triples *\$s subject \$name value* (where *\$s subject* may be @<user name>, #hash-tag or URI):

$$discuss \$project \text{ New Project } \$module \text{ New Module}, \quad (3)$$

$$\begin{aligned} &<subject> <predicate> \quad <object> \quad <predicate> \quad <object> \\ &\#WP2 \$hasNewCR \text{ Change data format. Find } \$details \text{ here } < URL >. \end{aligned}$$

In RDF format with Notation3 serialization the expression (3) has the following form:

$$myprj:WP2 \text{ prj-ont:hasNewCR } [\text{prj-ont:subject } "Change data format"; \text{prj-ont:details } "URL"]. \quad (4)$$

A modified CLIE approach. The paper [4] describes CLIE approach to processing of the text with the controlled language for creating ontologies. In this work a new approach of the controlled language on the basis of CLIE and nanoformats is suggested, the difference of which from CLIE lies in presence of the two constituents of the sentence semantic patterns: static and dynamic. The static constituent defines rigid rules of relative positions of the parts of sentence by rules of the nanoformats. The dynamic constituent defines the character of semantic relations of substances described in the sentence (*predicate* for *object* and *subject*). The approach allows creating RDF descriptions of text messages with nanoformats on the basis of semantic repositories RDF with RDF-S and OWL schemas.

Table 2

An example of patterns of Twitter Data sentences

Pattern of the sentence	Usage / example
$\$<predicate> <object>$	Transforming a variable-value pair: <i>\$population 2300</i>
$@<subject> \$<predicate> <object>$	Transforming a triple of an event description <i>@ChangeBoard \$hasMeeting tomorrow</i>
$\#<subject> \$<predicate> <object> \$<predicate> <object>$	Transforming a triple with additional information about subject: <i>#WP2 \$hasNewCR change_data_format. Find \$details here < URL ></i>
$@<?> \$<inferred/default_predicate> <?>$ Unknown subject, predicate or object	Default behavior 1) try to find the class by the keyword – name of the substance; 2) try to find the <i>predicate</i> property by the class found; 3) otherwise generating new objects of the ontology and setting their “default” relation, e.g., «hasRelation»

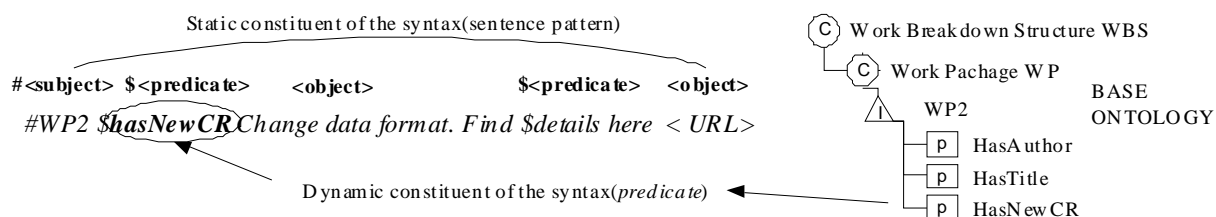


Fig. 3. Sentence patterns CLIE with static and dynamic constituents for Twitter Data

Fig. 4 shows the schema of NLP/IE analysis of the new communication system which is a modified CLIE realization to be able to use nanoformats of microblogging. If the messages are pre-annotated with nanoformats, it is possible to use simplified NLP procedures for filtering out the symbols that are not significant and then tagging the text with definitions of *subject*, *predicate* and *object* for consequent semantic translation into RDF. For this purpose it is possible to employ Brill tagging or Common Pattern Specification Language (CPSL) [8].

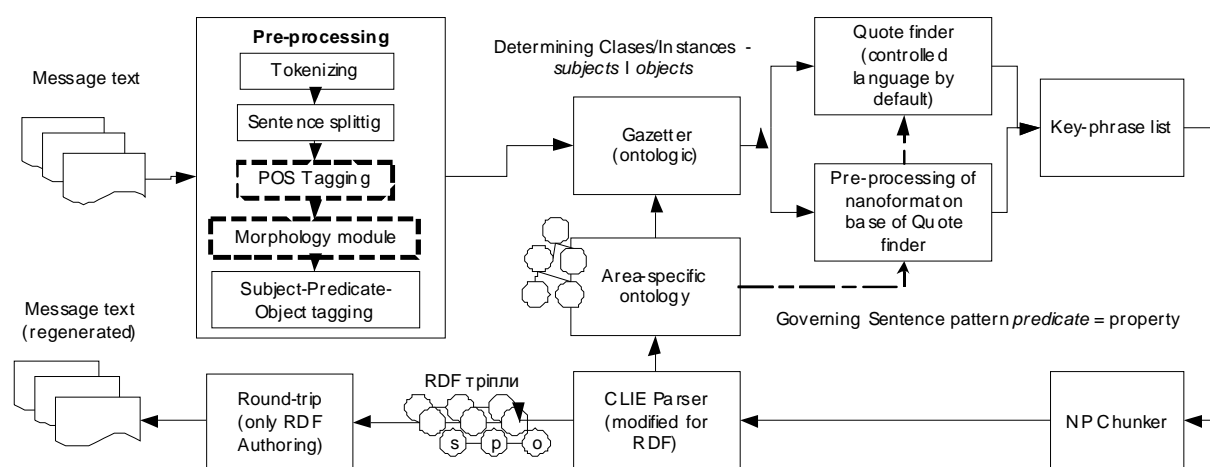


Fig. 4. Translating messages with nanoformats and controlled language into RDF with CLIE

Ontologic gazetter is a new element of CLIE schema that annotates a text with references to URI of ontological resources. Those can be area- and project- specific ontologies or their conjunctions with more general ontologies. This instrument must have ability of annotating not only with the *class* values, but also with the *instance* values of the classes. As a basis, it is planned to use OBIE approach (Ontology Based Information Extraction) [9].

Quote finder is a modified CLIE module that annotates text using the principle of Table 2 (excluding the last case) and finds key phrases of the messages that correspond to the combination of 1) allowed patterns of nanoformats and 2) properties of substances taken from the ontologies as *predicates*. The system can work with the standard CLIE approach also if it does not find any “nanoformatted” annotations in the message. *Key-phrase list* contains key phrases found by *Quote finder*. *NP Chunker* is an auxiliary instrument.

Modified *CLIE Parser* 1) finds the elements not identified by *Ontologic gazetter* and *Quote finder* (<subject>, <predicates> i <objects>) and supplements the ontology with newly generated objects, obligatory specifying a default value of *predicates*, for example, “hasRelation” (this is illustrated by the last case of Table 2); 2) generates RDF triples of useful information.

The modified Round-trip module regenerates the text of the message incoming from RDF triples for synthesis of individual data elements (optionally, depending on the task of information visualization).

Potential of data usage in RDF representation. In the context of this work there is a necessity

to create an easy to use mechanism of searching the information accumulated in such system. As specified above, it is supposed to store the semantically processed information of microblogging messages in the form of RDF triples in a semantic repository, an alternative to relational database (RDBMS), which offers similar functions of storing and querying the data. Information in such repository is stored not as data in the related tables, but in a repository of the triple statements where every line stores one triple $\langle object, predicate, subject \rangle$ that has pointers to various data with the help of URI/IRI (Fig. 5).

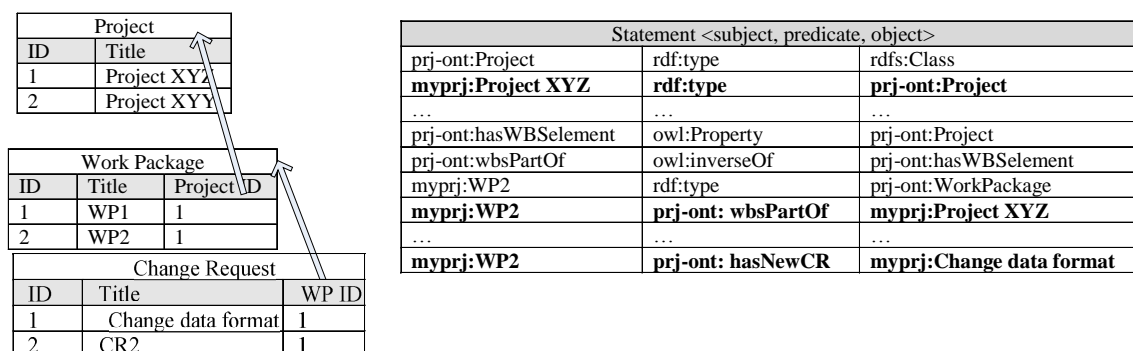


Fig. 5. Differences of data representation in RDBMS and semantic repositories

Semantic repositories are more suitable for effective usage as knowledge bases than relational databases as they have embedded mechanisms of implicit data mining [11]. Modern semantic repositories, as a rule, use ontologies as semantic schemas. Semantic schemas of ontologies define models of the data stored: data types, possible data relations as well as the rules by which the implicit data is being sought for, *entailment rules*.

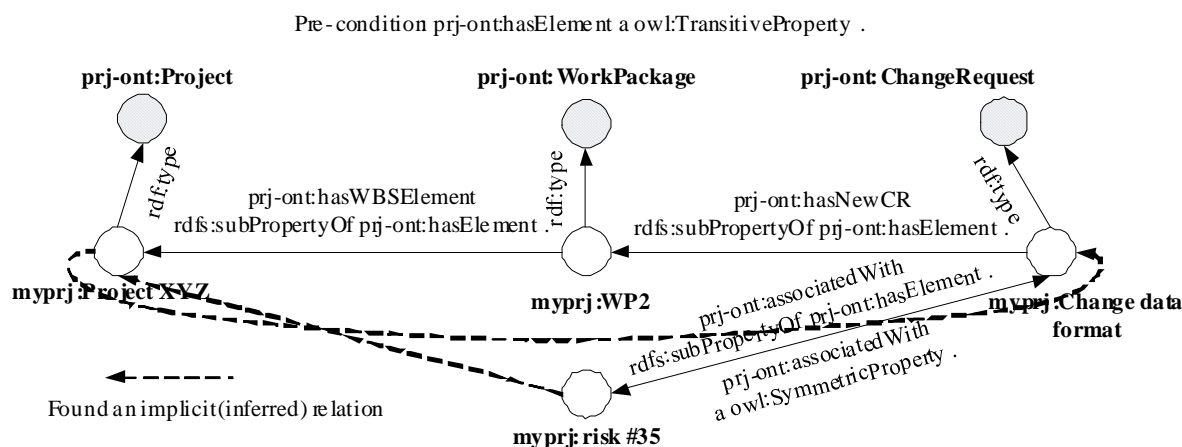


Fig. 6. An example of an implicit RDF data inference with the help of an OWL scheme

Fig. 6 depicts a part of RDF graph that uses semantics of schema-ontologies for inferring relations between the substances. Due to transitive properties of *hasElement* predicate and properties of *hasWBSelement*, *hasNewCR* and *associatedWith* as its *subProperty*, it is possible to make conclusions that: 1) the request on change in the work package WP2 with the name “Change data format” belongs to project *Project XYZ* (because WP2 belongs to that project); 2) *risk #35* that has appeared as a result of raising change request “Change data format” belongs to project *Project XYZ*. Also there is a two-way relation between change request “Change data format” and risk *risk #35*.

A peculiarity of the modern approach to the usage of semantic representation languages for se-

mantic repositories is that the set of capabilities for searching implicit data, which is based on the known information, depends on the chosen language of semantic schema [11]. Thereby, using various languages and schemas of the dialects (RDF-S, OWL Light, OWL DLP, OWL Horst, OWL 2 EL/QL/RL/DL) depending on the need, it is possible to choose different sets of *entailment rules* and vary the “depth” of search of hidden information. More powerful the entailment rules are, more burden the process bears. OWL DLP [12], which is supposed to be used, is a dialect of the powerful OWL DL with some restrictions, and logic programming that will enable simplification of using schemas.

Sequential information filters and search result representation. In work [5] Mokin V. B. and Yasholt A. R. have elaborated theoretical and algorithmic principles of new method for automated formation of complicated reports of information system relative data bases, which is an analog of the sequential method for designing technical systems. The method, based on representing input and output data in the form of a sequential model, allows synthesizing the program code for formation of the report for the given data sets without requiring the knowledge of SQL language.

Languages of queries to semantic data, SPARQL and SeRQL, have many features common with SQL. It is possible to apply an analogy and to make use of the theoretical and algorithmic apparatus for automatic formation of queries to semantic repositories with Ontologic OWL schemas, with the complexity of forming queries being hidden from the users.

As stated in [5], the key point of the algorithm of the sequential method special apparatus lies in creating a model of the report that ties its input and output data. Special expression (5), which is referenced to in [5], describes the main element of the sequence apparatus that formalizes relations of inputs and outputs of the finite state machines:

$$\phi | - \varphi, \quad (5)$$

where ϕ and φ are the functions of algebra of logic. The right side of this expression (function φ) takes a unit value in the case when a single value (function ϕ) is taken by the left side of the expression. This also can be interpreted as “statement φ is true if statement ϕ is true”. In the work [5] a sequential expression of the model of information systems with relational databases for two input and one output variables is represented as:

$$x_1 x_2 | \frac{\tilde{s}_1}{\tilde{s}_2} y_1, \quad (6)$$

where input variables x_1, x_2 , which are the fields of the table \tilde{s}_1 , correspond to (have logical connection) output variable y_1 , which is the field in table \tilde{s}_2 .

The key fact is that data of semantic RDF repository is an aggregate of RDF triples (a graph) unlike the relational tables with data fields (see Fig. 5). One of the main ideas of semantic web technologies is the possibility of unique data identification with the help of URI irrespective of the physical place of data location. Thus, in semantic repositories SELECT statement and FROM clause have a somewhat different meaning. In SPARQL [13] FROM clause is intended for referring to RDF graphs and is optional. In SeRQL [14] FROM is referring intended for referring to “path expressions” and is also optional.

Drawing an analogy with relational databases where, in general case, tables describe substances of a program application (e. g., projects, work packages, change requests), it is possible to choose classes of *objects* and *subjects* of RDF triples as analogs of tables \tilde{s}_1, \tilde{s}_2 of expression (6). *Classes* are stable key elements of RDF graphs (like the tables for RDBMS database schema) and are always associated with information about relations in graphs (*predicates*). Both input and output variables of SPARQL and SeRQL queries are references to actualized *instances* of *classes* or *properties* (*attributes*) of those *instances* described with the help of RDF-S / OWL.

Expression (6) for RDF triples can be represented in such a way that input variables x_1, x_2 ,

which are instances of the class or properties of RDF representation class \tilde{s}_1 , correspond to (have a logical connection with) output variable y_1 , an instance or a property of class \tilde{s}_2 . E. g., class \tilde{s}_1 may be *prj-ont:Project* (class of Project), and properties-variables may be *rdf:label* (title of a project), and *prj-ont:hasWBSElement* (indicator of the presence of elements associated with the project WBS). Class \tilde{s}_2 may be *prj-ont:ChangeRequest* (class of Project's Change Request), and output variable may be a list of the project change requests. It is important to mention that whatever query form is used, in the principle of the hidden information search in semantic repositories with ontology-based schemas will always work (Fig. 8).

Drawing an analogy with [4], we will demonstrate building of SPARQL queries and sequences for two sequential filters of the information dashboard, shown in Fig 7, with the help of expressions (7).

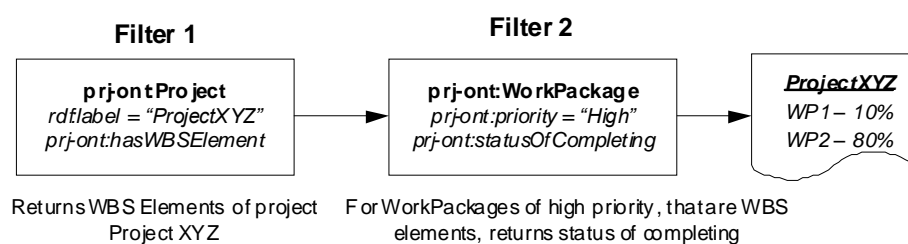


Fig. 7. An example of semantic pattern of the information dashboard with two filters

SELECT ?workpackage

WHERE prj-ont:Project rdf:label "ProjectXYZ" ;prj-ont:hasWBSelement ?workpackage .(7)

SELECT ?statusOfCompleting

WHERE<[output information of the first query]> prj-ont:hasPriorityLabel "High";
prj-ont:hasStatusOfCompleting ?statusOfCompleting .

With the help of new symbolic notations we transform the expressions of input, output data and classes from (7) into the form that allows writing down the queries as sequences:

$$\begin{aligned}
 x_1 &= (rdf:label "ProjectXYZ"), x_2 = (prj-ont:hasWBSelement ?workpackage), \\
 x_3 &= (prj-ont:hasPriorityLabel "High"), \\
 x_4 &= (prj-ont:hasStatusOfCompleting ?statusOfCompleting), \\
 \tilde{s}_1 &= (prj-ont:Project), \tilde{s}_2 = (prj-ont:WorkPackage), \\
 y_1 &= [myprj:WP_n \text{ rdf:type } prj-ont:WorkPackage], \\
 y_2 &= [prj-ont:statusOfCompleting \text{ rdf:type } rdfs:Literal],
 \end{aligned} \tag{8}$$

where expressions specify that the variable contains a set of data.

As specified above, a peculiarity of semantic data representation is their structure as *classes*, *instances* and *properties (attributes)*. For further development of the theory of sequential forms of queries to semantic data, it is expedient to take this peculiarity into account. With the help of the expressions of the form $A(B)$, symbolizing belonging of object B to object A, we can write the following example of the description of query elements:

$$\tilde{s}_1 = (C_1, x_2 = P(I(C_1)), y_1 = [I(C_1)]), \quad (9)$$

where C is a class, I – an instance of the class, P – a property of the instance of the class.

Then in the sequenced generalized form the expressions of queries (7), where output variable of the first query is an input variable of the second query, will get such a form:

$$P(I(C_1))_1 P(I(C_1)) \left| \frac{C_1}{C_2} [I(C_2)] \text{ and } [I(C_2)] P(I(C_2))_1 P(I(C_2))_2 \right| \frac{C_2}{C_2} [I(C_1)]. \quad (10)$$

It is important to mention that the sequences for the queries to semantic data on the basis of classes may have the same input and output classes. This can be seen from the second sequence of expression (10).

The sequential apparatus makes it possible to perform operations with the sets of sequences such as “And” and “Or”, which allows reducing them to shortened expressions. The simplified sequences enable automatic generation of the already optimized SPARQL code.

UML Component diagram of the system prototype. For prototyping, NLP/IE set of GATE tools has been chosen as well as semantic repository Ontotext SwiftOWLIM because of their vast set of instruments, effective integration, production-level technology, continuous improvement due to research activities and availability of free solutions for research projects [8]. Fig. 8 shows the component diagram of the system.

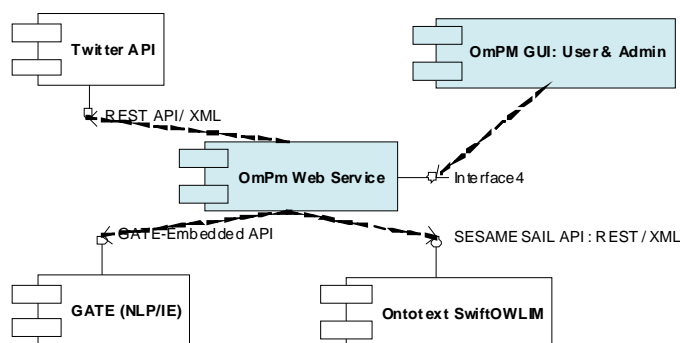


Fig. 8. Component diagram of the system

Conclusions

This work suggests a new type of management system which uses the properties of social media as CAS. Its main features are: “communication-centric concept”; using of non specialized social media tools as central communication environment of organizational and project management; using short of microblogging messages instead of large amounts of text, which simplifies the semantic annotation process and increases accuracy; using sequential filters of information search in semantic representations, which simplifies the usage. The systems of such kind are intended, first of all, for distributed non-profit and small profit organizations, scientific and educational organizations, implementation of communication rationality approach [15] in public projects. The next step is the system prototype creation.

REFERENCES

1. John H. Miller. Complex Adaptive Systems: An Introduction to Computational Models of Social Life / H. Miller John, E. Page Scott. – Princeton University Press, 2007. – 284 p.
2. McAfee Andrew. Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges / Andrew McAfee. – 1st edition. – Harvard Business School Press, 2009. – 240 p.
3. Manas Jerry. Managing the Gray Areas: Essential Insights for Leading People, Projects & Organizations / Jerry

Manas. – RMC Publications, Inc, 2007. – 232 p.

4. User-friendly ontology authoring using a controlled language [Електронний ресурс] / V. Talban, T. Polanjnar, H. Cunningham, K. Botcheva // Department of Computer Science, University of Sheffield, 2006. – Режим доступу: <http://gate.ac.uk/sale/lrec2006/clie/clie.pdf>.

5. Мокін В. Б. Новий метод автоматизованого формування звітів інформаційних систем на основі секвенційного підходу / В. Б. Мокін, А. Р. Яшолт // Вісник Вінницького політехнічного інституту. – 2008. – № 3. – С. 50 – 56

6. A guide to the project management body of knowledge (PMBOK Guide)-Forth Edition : ANSI/PMI 99-001-2008 An American National Standard. – Project Management Institute, Inc., 2008. – 470 p.

7. Twitter Data – a simple, open proposal for embedding data in Twitter messages [Електронний ресурс] / Todd Fast, Jiri Copsa. – 2009. – Режим доступу : <http://twitterdata.org/>.

8. The Common Pattern Specification Language [Електронний ресурс] / Douglas E. Appelt, Boyan Onyshkevych // Artificial Intelligence Center SRI International. – Режим доступу: <http://www.aclweb.org/anthology/X/X98/X98-1004.pdf>.

9. OBIE Ontology Based Information Extraction [Електронний ресурс] // The University of Sheffield. 1995 – 2001. – Режим доступу: <http://gate.ac.uk/demos/obie/obie.html>.

10. Wilcock Graham. Introduction to Linguistic Annotation and Text Analytics / Graham Wilcock. – 1st edition. – Morgan & Claypool Publishers, 2009. – 160 p.

11. Ontotext and GATE Training (Module 15) – RDF, SPARQL and Semantic Repositories : GATE Training Course August 2010 [Електронний ресурс] // The University of Sheffield. 1995 – 2001. – Режим доступу : <https://gate.ac.uk/wiki/TrainingCourseAug2010/track4-slides/module-15.pdf>.

12. DLP – An introduction [Електронний ресурс] / Denny Vrandečić, Peter Haase, Pascal Hitzler, York Sure, Rudi Studer // Institute AIFB University of Karlsruhe, Germany. – Режим доступу: <http://logic.aifb.uni-karlsruhe.de/dlp/download/dlpintro.pdf>.

13. SPARQL Query Language for RDF, W3C Recommendation [Електронний ресурс] // W3C. – 2008. – Режим доступу : <http://www.w3.org/TR/rdf-sparql-query/>.

14. The SeRQL query language (revision 1.2) [Електронний ресурс] // Aduna. 1997 – 2009. – Режим доступу : <http://www.openrdf.org/doc/sesame/users/ch06.html#section-blanknodes>.

15. Erwann Michael-Kerjan. The Irrational Economist: Making Decisions in a Dangerous World / Erwann Michael-Kerjan, Paul Slovic. – Public Affairs, 2010. – 336 p.

Bilodray Yuriy –Leading software engineer, PMP, Senior Engineering Project Manager (Eng.), Yuriy.Belodray@gmail.com.
Arisent Technologies Ukraine.

Volodymyr Mesiura – Cand. Sc. (Eng.), Professor of the Computer Science Department, Institute of Information Technologies and Computer Engineering.
Vinnitsa National Technical University.