

# Semantic web and knowledge graphs as an educational technology of personnel training for nuclear power engineering\*

Victor Telnov<sup>1</sup>, Yuri Korovin<sup>1</sup>

<sup>1</sup> Institute for Nuclear Power Engineering, NRNU «MEPhI», 1 Studgorodok, Kaluga reg., Obninsk, 249040, Russian Federation

Corresponding author: *Victor Telnov* ([telnov@bk.ru](mailto:telnov@bk.ru))

---

**Academic editor:** *Georgiy Tikhomirov* ♦ **Received** 10 May 2019 ♦ **Accepted** 2 August 2019 ♦ **Published** 25 September 2019

---

**Citation:** Telnov V, Korovin Y (2018) Semantic web and knowledge graphs as an educational technology of personnel training for nuclear power engineering. *Nuclear Energy and Technology* 5(3): 273–280. <https://doi.org/10.3897/nucet.5.39226>

---

## Abstract

The technologies of knowledge representation and inference in an artificial intelligence system focused on the domain of nuclear physics and nuclear power engineering are considered. The possibilities of description logics and graph databases of nuclear knowledge for the generation of cognitive hypotheses, using in addition to deduction and other ways of reasoning, such as inductive inference and reasoning based on analogies, are discussed. The use of adequate description logic and measures of semantic similarity is substantiated. Interactive visual navigation and reasoning on the knowledge graphs are performed by means of special retrieval widgets and the smart RDF browser. Operations with semantic repositories are implemented on cloud platforms using SPARQL queries and RESTful services. The proposed software solutions are based on cloud computing using DBaaS and PaaS service models to ensure scalability of data warehouses and network services. Example of use of the offered technologies and software has been given.

## Keywords

Nuclear education, semantic web, knowledge graph, cloud computing

---

## 1. Introduction and motivation

Since the 1960s, in the framework of research on artificial intelligence, various formalisms for knowledge representation (semantic networks, frame systems, etc.) have been developed (Harmelen et al. 2008). In 2019, the ontology description languages RDF, OWL (W3C 2012), knowledge graphs and description logics (Baader et al. 2010) provide a modern theoretical basis for the creation of systems and methods of acquisition, presentation, processing and integration of problem-oriented knowledge in computer systems, which, in particular, is confirmed by the current standards W3C in the field of semantic web.

The reports of International Conference on Semantic Systems, International Workshops on Description Logic noted the growing interest of giants of the IT industry (Google, Facebook, Wikimedia) to graph models of knowledge representation and description logics. As of 2019 educational web-portals of universities, national centers for the exchange of scientific information, world nuclear data centers underused semantic web technologies. As for the inductive inference rules in graphs, the following considerations make them useful. First, inductive inference rules based on consideration of possi-

\* Russian text published: *Izvestiya vuzov. Yadernaya Energetika* (ISSN 0204-3327), 2019, n. 2, pp. 219–229.

ble alternatives (precedents) allow to generate cognitive hypotheses (fuzzy knowledge) that cannot be obtained directly by deductive reasoning on the graph. Secondly, inductive inference is one of the basic technologies of semantic annotation of network content, when it is necessary to redesign, expand and update existing graphs with new knowledge. With the help of inductive inference the problems of classification and clustering of new concepts and individuals in the semantic base of nuclear knowledge are solved.

The aim of the work presented in the paper is to create a semantic web portal of knowledge in the domain of nuclear physics and nuclear power engineering based on ontology and using graph databases deployed on cloud platforms. The task of the study was to create the following graphs of nuclear knowledge:

- World nuclear data centers;
- Nuclear research centers;
- Events and publications from CERN;
- IAEA databases and network services;
- Nuclear physics at MSU and MEPhI;
- Nuclear physics journals;
- Joint nuclear knowledge graph.

The potential beneficiaries of information solutions and technologies that are proposed in the paper are students, teachers, experts, engineers, managers and specialists in the domain of nuclear physics and nuclear power engineering (target audience).

## 2. Methodology and state of the art

### 2.1 Adequate description logic

The choice of adequate description logic (DL) for project (Telnov 2019) is dictated, on the one hand, by the requirement of a complete and accurate knowledge representation about the subject area (domain) as far as possible, on the other hand, by the necessity to work effectively with remote semantic repositories using SPARQL queries. The OWL is a knowledge representation language standardized with the W3C, which is a crucial application of description logics. The main building blocks of OWL are very similar to those of DLs, with the main difference that concepts are called classes and roles are called properties. The expressive description logic underlying the contemporary OWL 2 submission is called SROIQ (Krotzsch et al. 2013, Horrocks and Sattler 2001).

The description logic with signature SROIQ is an extension of the earlier description logic SHOIN by all expressive means that were suggested by ontology developers, and which do not affect its decidability and practicability. Among others, complex role inclusion axioms of the form  $R \circ S \sqsubseteq R$  or  $S \circ R \sqsubseteq R$  to express propagation of one property along another one have been added, which have proven useful for many domains. Furthermore,

SHOIN has been expanded with reflexive, antisymmetric, and irreflexive roles, disjoint roles, a universal role. Named individuals occur naturally in ontologies as names for specific things, persons, institutes, etc. Nominals can be viewed as an artificial supplement to ABox, which provides additional expressive power to DL.

In project (Telnov 2019) it is required to simulate not only abstract objects (documents, people, institutions, etc.), but also specific properties of objects, for example, string and temporal parameters of events and publications in the knowledge graph named “Events and publications from CERN”, “Journals in nuclear physics”, spatial characteristics in the knowledge graph named “Nuclear research centers”.

Specific values (temporal parameters, spatial characteristics, etc.) cannot be represented in the concept description language directly, because when moving from one interpretation to another, these specific elements and the relationships between them may change, while it is required that they remain unchanged. The solution is to select a separate “concrete” domain (Baader et al. 2018) with a fixed set of predicates. Also, it requires a special roles, connecting abstract elements with specific values. Finally, a new constructs that enable to build concepts on the basis of these linking roles and “concrete” predicates are needed. This requires expanding the concept description language with a set  $\mathbf{D}$  of concrete datatypes and with concepts of the form  $\exists R.d$  and  $\forall R.d$ , where  $d \in D$  and  $R$  is a role. For each  $d \in D$ , a set  $d^D \in \Delta_D$  is associated, where  $\Delta_D$  is the domain of all datatypes. It is reasonable to assume that:

- the domain of interpretation of all concrete datatypes
- there exists a sound and complete decision procedure for the emptiness of an expression of the form  $d_i^D \cap \dots \cap d_i^D$ , where  $d_i^D$  is a concrete datatype from  $D$ .

The corresponding expansion of DL is commonly referred to as SROIQ(D). Summary of the syntax and semantics of the description logic used in the project (Telnov 2019) is presented in Table 1, where  $I$  is the interpretation function.

### 2.2 On the question of inductive reasoning on knowledge graphs

Nuclear knowledge sometimes involves various degrees of uncertainty. For such a reason, in the semantic web context, difficulties arise when modeling real-world domains using only classical logical formalisms. Alternative approaches often suggest probabilistic knowledge, while this is hardly always appropriate and justifiable (Bobillo et al. 2013). In addition, a purely deductive exact inference may be infeasible for web-scale ontological knowledge bases, and it does not exploit statistical regularities in data. Approximate deductive and inductive inferences, which are based on consideration of precedents (alternatives), are offered to alleviate such problems, see d’Amato et al. (2005, 2006, 2009, 2013), d’Amato 2007 and Minervini et al. (2016).

**Table 1.** Summary of the syntax and semantics of the used SROIQ(D) description logic.

Construct name	Syntax	Semantics
atomic concept $C$	$A$	$A^I \subseteq \Delta^I$
abstract role $R_A$	$R$	$R^I \subseteq \Delta^I \times \Delta^I$
concrete role $R_D$	$T$	$T^I \subseteq \Delta^I \times \Delta_D^I$
nominals $I$	$\{o\}$	$\{o\}^I \subseteq \Delta^I, \#\{o\}^I = 1$
datatypes $D$	$d$	$d^D \subseteq \Delta_D$
conjunction	$C \sqcap D$	$(C \sqcap D)^I = C^I \cap D^I$
disjunction	$C \sqcup D$	$(C \sqcup D)^I = C^I \cup D^I$
negation	$\neg C$	$(\neg C)^I = \Delta^I \setminus C^I$
	$\neg d$	$(\neg d)^I = \Delta_D \setminus d^D$
exists restriction	$\exists R.C$	$(\exists R.C)^I = \{x \in \Delta^I \mid \exists y \in \Delta^I : (x, y) \in R^I \wedge y \in C^I\}$
value restriction	$\forall R.C$	$(\forall R.C)^I = \{x \in \Delta^I \mid \forall y \in \Delta^I : (x, y) \in R^I \Rightarrow y \in C^I\}$
atleast restriction	$\geq nR.C$	$\geq nR.C^I = \{x \in \Delta^I \mid \#\{\forall y \in \Delta^I : (x, y) \in R^I\} \cap C^I \geq n\}$
atmost restriction	$\leq nR.C$	$\leq nR.C^I = \{x \in \Delta^I \mid \#\{\forall y \in \Delta^I : (x, y) \in R^I\} \cap C^I \leq n\}$
datatype exists	$\exists T.d$	$\{\exists T.d\}^I = \{x \in \Delta^I \mid \exists y \in \Delta^I : (x, y) \in T^I \wedge y \in d^D\}$
datatype value	$\forall T.d$	$\{\forall T.d\}^I = \{x \in \Delta^I \mid \forall y \in \Delta^I : (x, y) \in T^I \Rightarrow y \in d^D\}$
local reflexivity	$\exists R.Self$	$\{\exists R.Self\}^I = \{x \in \Delta^I \mid (x, x) \in R^I\}$
concept assertion	$C(a)$	$a^I \in C^I$
role assertion	$R(a, b)$	$(a^I, b^I) \in R^I$
individual equality	$a = b$	$a^I = b^I$
individual inequality	$a \neq b$	$a^I \neq b^I$
concept inclusion	$C \sqsubseteq D$	$C^I \subseteq D^I$
concept equivalence	$C \equiv D$	$C^I = D^I$
role inclusion	$R \sqsubseteq S$	$R^I \subseteq S^I$
role equivalence	$R \equiv S$	$R^I = S^I$
complex role inclusion	$R_1 \circ R_2 \sqsubseteq S$	$R_1^I \circ R_2^I \subseteq S^I$
role disjointness	$Disjoint(R, S)$	$R^I \cap S^I = \emptyset$

Today standard ontology markup languages are supported by mature semantics of DL along with a number of available reasoning algorithms (Baader et al. 2010). However, some tasks in the ontology life cycle, such as their construction and/or integration, still largely delegated to knowledge specialists. For the successful development of semantic technologies it is desirable that the construction of the knowledge databases should be supported by automated inductive inference procedures, including entity classification and clustering tasks. The induction of structural knowledge like the taxonomies is not new in machine learning, especially for the task where clusters of similar objects are aggregated in hierarchies according to heuristic criteria or similarity measures (d'Amato 2007). In the Inductive Logic Programming (Muggleton and Raedt 1994) attempts have been made to extend relational learning techniques towards representations based on both clausal and description logics. These methods mostly are based on an empirical search and generally implement bottom-up algorithms that tend to induce overly specific concept definitions and narrowly specialized ontologies.

Generally, the problem of the induction of structural knowledge turns out to be a uneasy task in first-order logic or equivalent representations. In order to overcome the existing difficulties, the last decades have seen the development of research related to the calculation of similarity measures for concepts and individuals in ontologies. Similarity measure plays an important role in information retrieval and information integration as a means for compar-

ing concepts and/or concept instances that can be retrieved or integrated across heterogeneous knowledge databases. It seems that the quite significant from a practical point of view results were obtained by d'Amato et al. (2013). To determine the similarity measure a set of similarity values has to be define, usually a set of the real numbers is used for this. Then it is required to determine a function for a pair of objects that will calculate the measure of their similarity. Formal definitions of similarity and dissimilarity measures were given by d'Amato et al. (2006) and d'Amato (2007).

Naive semantic similarity can be defined as a path distance between entities in the hierarchical structure of the ontology. More meaningful methods to assess semantic similarity within a single ontology are feature matching and information content. There are measures have been developed to compute similarity values among classes belonging to different ontologies. For instance, a similarity function can detect similar entity classes by using a matching process, making use of special dictionaries, semantic neighborhood, and discriminating features. Of particular interest is the approach proposed by d'Amato et al. (2006), aimed at finding commonalities among concepts or among individuals, employs the Most Specific Concept (MSC) method, that turns the instance checking task (that is deciding whether an individual is an instance of a concept) into a TBox reasoning problem (d'Amato et al. 2009).

Let there be a knowledge database  $KB = \langle T, A \rangle$ , contains two components: a TBox  $T$  and an ABox  $A$ . Let  $C$  and  $D$  be two concept descriptions in a  $T$ . Given a concept

$C$  in  $T$ , it is possible to consider its extension  $C^I$ , where  $I$  is the interpretation function. Further the canonical interpretation of the ABox is considered, when constants in the ABox are interpreted as themselves and different names for individuals stand for different domain objects. The semantic similarity measure is defined as in the following (d'Amato et al. 2009):

**Definition 1 (Semantic Similarity Measure).** Let  $L$  be the set of all concepts in DL and let  $A$  be an ABox with canonical interpretation  $I$ . The Semantic Similarity Measure  $s$  is a function

$$s : L \times L \rightarrow [0, 1]$$

which is defined as follows:

$$s(C, D) = \frac{|X^I|}{|C^I| + |D^I| - |X^I|} \cdot \max\left(\frac{|X^I|}{|C^I|}, \frac{|X^I|}{|D^I|}\right)$$

where  $X = C \sqcap D$  and  $(\cdot)^I$  computes the concept extension w.r.t. the interpretation  $I$ .

The measure can be explained as follows. In case of semantic equivalence of the concepts  $C$  and  $D$ , the maximum value of the similarity will be calculated. In case of disjunction, the minimum value of similarity will be assigned because the two concepts are totally different: their extensions do not overlap. Finally, in the case of overlapping concepts, a value in the range  $]0, 1[$  will be computed (d'Amato et al. 2009).

**Definition 2 (Most Specific Concept).** Let there be a knowledge database  $KB = \langle T, A \rangle$ . Given an ABox  $A$  and an individual  $a$ , the Most Specific Concept of  $a$  w.r.t.  $A$  is the concept  $C$ , denoted  $MSC_A(a)$ , such that  $A \models C(a)$  and  $\forall D$  such that  $A \models D(a)$ , it holds:  $C \sqsubseteq D$ . Here  $\models$  stands for the standard semantic deduction.

Once the most specific  $MSC_A(a)$  of an individual  $a$  is known, to decide if  $KB \models D(a)$  holds for an arbitrary concept  $D$ , it suffices to test if  $T \models MSC_A(a) \sqsubseteq D$ . This method, unfortunately, loses its simplicity and efficiency when applied to large and complex ontologies, as it tends to generate very large MSCs that could lead to intractable reasoning. Revised MSC method for DL, allowing it to generate much simpler and smaller concepts that are specific enough to answer a given query, has been proposed by Xu et al. (2015).

Let  $c$  and  $d$  two individuals in a given ABox. Then it is possible to calculate  $C = MSC_A(c)$  and  $D = MSC_A(d)$ . According to d'Amato et al. (2009), now the semantic similarity measure  $s$  can be applied to these concept descriptions, thus yielding the similarity value of two instances:

$$\forall c, d : s(c, d) = s(C, D) = s(MSC_A(c), MSC_A(d))$$

The similarity value between a concept  $C$  and an individual  $a$  can be computed by determining the MSC of the individual and then applying the similarity measure:

$$\forall a : s(a, C) = s(MSC_A(a), C)$$

The complexity of  $s$  calculation depends on the complexity of the instance checking task for the adopted DL language, denote it as  $C$  (*InstanceChecking*). Similarity between concepts:  $s$  is a numerical measure, all calculus have constant complexity, instance checking is repeated three times: for concepts  $C, D$  and their intersection, so:

$$C(s) = 3 \cdot C(\text{InstanceChecking})$$

Similarity between an individual and a concept: in this case, besides of the instance checking operations required by the previous case, the MSC of the considered individual is to be computed. Thus, denoted by  $C(\text{MSC})$  the complexity of the MSC computing, get the complexity estimate:

$$C(s) = C(\text{MSC}) + 3 \cdot C(\text{InstanceChecking})$$

Similarity between individuals: this case is analogous to the previous one, the only difference is that now two MSC is to be computed for the arguments. So the complexity in this case is:

$$C(s) = 2 \cdot C(\text{MSC}) + 3 \cdot C(\text{InstanceChecking})$$

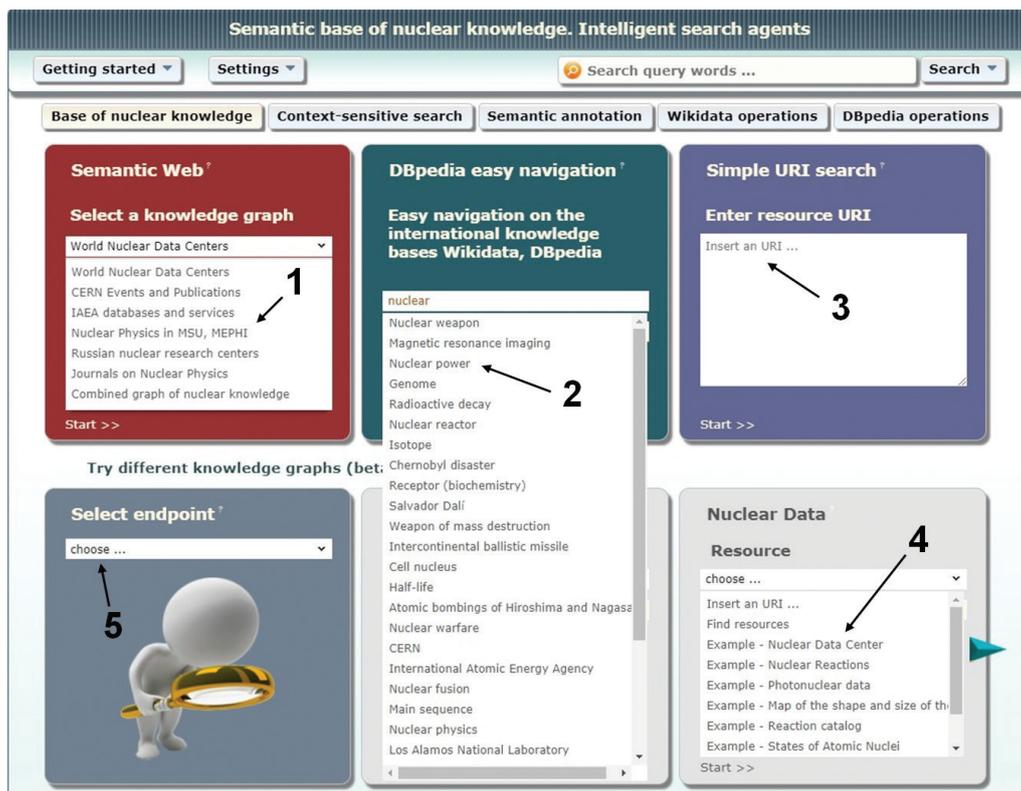
From the previous formulas it is clear that the computational complexity of the similarity measure sensitive to the choice of the DL. For the ALC logic,  $C(\text{InstanceChecking})$  has polynomial complexity. Computation of the MSC also implies instance checking and depends on algorithm properties.

## 3. Technology and case study

### 3.1 Semantic repositories, search widgets, intelligent RDF browser

From a practical point of view, knowledge graphs are placed in the data warehouse, which are called RDF-repositories or triple repositories. The project (Telnov 2019) largely uses the Google Cloud Platform (<http://console.cloud.google.com>) and Apache Jena (<http://jena.apache.org/>) framework on the free quota with each of the repositories serviced by a dedicated virtual machine. Remote asynchronous work with the Google Cloud Platform is performed using the standard SPARQL 1.1 query language through application programming interfaces in Java and JavaScript. Common operations are creating, reading, updating and deleting data in knowledge graphs. For the practical implementation of network requests to repositories, HTTP protocol methods GET and POST are used.

Each of the knowledge graphs contains thousands of triplets. Search widgets, shown in Figure 1, allow users to get to the right place of a specific knowledge graph, where the desired information objects will be detected and visualized. The principle of operation of search widgets is similar to the way information samples from the web us-



**Figure 1.** Search widgets designed for a quick immersion in knowledge graphs: 1 – selection of a knowledge graph for work; 2 – quick navigation in DBpedia; 3 – search by URI in the world semantic web; 4 – examples of working with knowledge graphs; 5 – selection of knowledge graph for demonstration.

ing popular search engines (Google, Yandex, etc.). As the user types the characters of the keywords in the search widget’s input line, the system rolls out an adequate list of entities from the corresponding knowledge graph. The user is prompted to select a suitable concept or individual and dive directly into the desired area of knowledge graph. Thereafter, a more accurate interactive visual navigation through the graph and inductive reasoning on graph becomes possible, which is implemented in an intuitive way using the intelligent RDF browser, as described below.

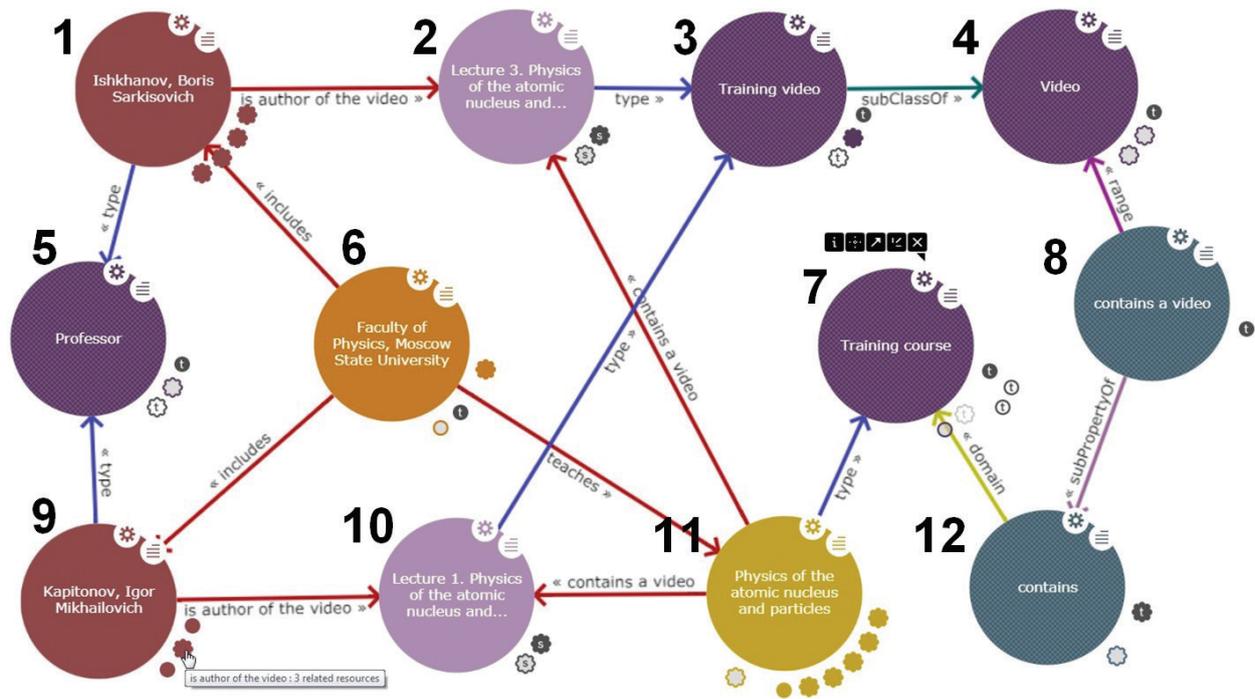
### 3.2 Interactive reasoning on the graph of knowledge (example)

The RDF browser is an essential attribute of the project (Telnov 2019), which distinguishes it from other known solutions in the field of semantic web. Once on the desired location of the desired knowledge graph using the search widget, then the user through the RDF browser can perform visual navigation on the graph, visiting its nodes in the correct order and extracting metadata, hypertext links, full-text and media content associated with the node. In this case, the neighborhood (environment, closure) of each node of the graph becomes visible and navigable. This neighborhood includes the nodes of the graph, through which the user initially entered the semantic web, as well as adjacent nodes of other graphs that are supported by the knowledge database.

The visual way of specifying the inference rules on the graph makes it stand out from the more traditional known reasoner’s interfaces, where inference rules are specified using SWRL language, logical predicates or a SPARQL-like syntax. It seems, that the intuitively clear interactive visual way of specifying inference rules is more friendly for unsophisticated users of knowledge graphs.

The knowledge graphs, presented in the project (Telnov 2019), all have built-in common patterns of reasoning, which, among other things, provide a means of navigating through graphs and means of searching in graphs. All reasoning and querying are implemented by means of smart RDF browser (created in JavaScript), which automatically generates the necessary SPARQL queries, then processes and classifies the results. Moreover, the results of many standard reasoning have already been calculated and combined into groups, which in the RDF browser have the form of petals around the nodes of the graph, see Figure 2. A click on the petal allows to expand any group of entities and explore elements of the group.

As an example, consider the following situation. Some student is preparing to pass the exam in nuclear physics at the Physics Faculty of Moscow State University. Let the student know only the name of the training course: “Physics of the atomic nucleus and particles” and the name of the professor: “I.M. Kapitonov”. Let us formulate the task: using the semantic educational web portal



**Figure 2.** Fragment of the knowledge graph titled “Nuclear Physics at MSU, MEPHI” as an example of the implementation of inductive reasoning on graph.

(Telnov 2019), it is necessary to find and study all the video lectures of this professor on this training course.

Also suppose, that the student discovered in YouTube a video lecture titled “Lecture 1. Physics of the atomic nucleus and particles”. He suggests, that this video lecture may be relevant to the training course being studied. Let us formulate the hypothesis: “Lecture 1. Physics of the atomic nucleus and particles” is taught by the professor “I.M. Kapitonov” at the Physics Department of Moscow State University and it is included in the training course “Physics of the Atomic Nucleus and Particles”.

To solve the task and test the validity of the hypothesis, it is necessary to perform the following obvious inductive reasoning on the knowledge graph step by step.

**Step 1.** Go to the educational web portal (Telnov 2019) and select the knowledge graph “Nuclear Physics at MSU, MEPHI”. It is possible to start the reasoning either with the classes “Training course”, “Training video”, “Professor”, etc. or with the specific entities “Physics of the atomic nucleus and particles”, “Lecture 1. Physics of the atomic nucleus and particles”, “Kapitonov”, etc.

**Step 2.** Let the student decided to begin the reasoning from the class “Training course”. The RDF browser workspace opens and the node of the graph with this name appears.

**Step 3.** The appeared node of the graph in Figure 2 is shown under number 7. We are interested in objects belonging to the class “Training course”. There are three such objects and they are associated with our node by the “type” property. With a mouse click, we will open the ob-

ject that is taught at the Physics Faculty of Moscow State University (see nodes with numbers 11 and 6 in Figure 2).

**Step 4.** Continuing to similarly disclose neighboring nodes for the object “Physical Faculty of Moscow State University” (node number 6 in Figure 2) by the “includes” property, for the “Kapitonov” object (node number 9 in Figure 2) by the “is author of the video” and / or for the object “Physics of the atomic nucleus and particles” (node number 11 in Figure 2) by the property “contains a video”, the student finally will make sure of the validity of the hypothesis and get the solution to the task, see node number 10 in Figure 2.

**Step 5.** The result obtained in Step 4 could also be achieved in the course of deductive reasoning, without considering possible alternatives. However, the use of inductive inference allows one to naturally extract from the graph additional knowledge that will not be easy to obtain with a simple deductive inference. Acting as described above, it is easy to find that some video lectures on the training course “Physics of the atomic nucleus and particles” at the Physics Faculty of Moscow State University are also taught by professor B.S. Ishkhanov, see node 1 in Figure 2. All video lectures and other learning materials of both professors for this training course became available. Through the knowledge graph, the full content of any training course is visually revealed and all the relationships are graphically shown.

As was shown in the above example, the process of inductive inference on knowledge graphs resembles a computer adventure game, does not require special skills, and is accessible to the inexperienced user. Knowledge

graphs, similar to the above, are used in the educational process at the NRNU MEPhI. Practice shows that university students master the techniques of interactive work with knowledge graphs within a few minutes.

### 3.3 Case study

The metrics of the computational processes presented in Table 2 below were obtained under the following test conditions:

- semantic repositories are hosted on the Amazon Web Service cloud platform at a free quota (the datacenter is located in Western Europe), each knowledge graph is served by its virtual computer;
- the measured speed of the Internet connection is about 90 Mbit/s;
- a standard workstation with an Intel Core i5-8400 2.8/4.0 GHz processor and 16 MB memory is used as a client computer.
- experiments were conducted on test ontologies, which included no more than a thousand entities (DBpedia is an exception).

The software architecture is presented in (Telnov 2017), (Telnov and Korovin 2019). An example of a knowledge graph “Nuclear physics at MSU and MEPhI” in a serialized format is available (Knowledge graph 2019).

The consumption of the computing resource is detailed in the debugging panels of browsers (Google Chrome, Mozilla Firefox), the corresponding parameters can be observed live when working with the semantic web portal (Telnov 2019). In all experiments, the total processing time did not exceed two or three seconds, with most of the time and the vast majority of computing resources spent on the operation of the interface to the knowledge database and network traffic.

## 4. Related works and conclusion

University of Manchester, Stanford University, University of Bari and a number of other universities are focused

on the issues of theory development and technology’s implementation for semantic web, description logics and incarnations of the ontologies description language OWL. Special mention should be made on the project (d’Amato et al. 2009), where for the first time an attempt was made to put into practice the methods of inductive reasoning for the purpose of semantic annotation of content from the web. To date, such network services are offered by some software companies (<http://jena.apache.org/documentation/inference/>).

As for the issues of visualization linked data (Bikakis and Sellis 2016), here one of the first successful projects was Lodlive (Camarda et al. 2012), which provided a tool for easier surfing through the DBpedia knowledge base. It is important continue to develop and improve tools for intuitive perception of linked data for non-professionals. VOWL (Schlobach and Janowicz 2016) is one of the modern project for the user-oriented representation of ontologies, it proposes the visual language, which is based on a set of graphical primitives and an abstract color scheme. LinkDaViz (Thellmann et al. 2015) propose a web-based implementation of workflow which guides users through the process of creating visualizations by automatically categorizing and binding data to visualization parameters. The approach is based on a heuristic analysis of the structure of the input data and a visualization model facilitating the binding between data and visualization options. The resulting assignments are ranked and presented to the user. SynopsViz (Bikakis et al. 2014) is a tool for scalable multi-level charting and visual exploration of very large RDF & Linked Data datasets. The adopted hierarchical model provides effective information abstraction and summarization. Also, it allows efficient –on the fly– statistic computations, using aggregations over the hierarchy levels.

In contrast to the above solutions, the project (Telnov 2019) is mainly focused on the implementation in educational activities of universities and is not limited to visualization of knowledge graphs and interactive navigation, but is aimed at the introduction of the latest semantic web technologies to the learning process, taking into account the achievements in the field of uncertainty reasoning.

**Table 2.** Some results of testing the performance of the knowledge graphs.

Test script	Test data set	Computing process metrics
Launching the main page of the web portal (Search widgets)	Knowledge graph “World nuclear data centers”	Number of network requests: 162. Web page loading time: 1140ms. Repository loading time: 850ms. Server timeout: 150ms.
Discovery and visualization of the knowledge graph (RDF Browser)	Knowledge graph “World nuclear data centers», object «Center for Photonuclear Experimental Data”	Number of network requests: 39. Web page loading time: 893ms. Repository loading time: 402ms. Server timeout: 44ms.
Discovery and visualization of the knowledge graph (RDF Browser)	Knowledge graph “Nuclear physics at MSU and MEPhI”, object “Physics of the atomic nucleus and particles”	Number of network requests: 57. Web page loading time: 1050ms. Repository loading time: 643ms. Server timeout: 17ms.
Discovery and visualization of the DBpedia knowledge base (RDF Browser)	DBpedia knowledge base, object “World War II”	Number of network requests: 37. Web page loading time: 2060ms. Repository loading time: 753ms. Server timeout: 15ms.

## Acknowledgments

The reported study was funded by Russian Foundation for Basic Research according to the research projects №18-07-00583, №19-47-400002, by Government of the

Kaluga Region according to the research project №19-47-400002 and was funded by Vladimir Potanin Foundation according to the project №GC190001383.

## References

- d'Amato C, Fanizzi N, Esposito F (2005) A Dissimilarity Measure for ALC Concept Descriptions. In: SAC Proceedings of the 2006 ACM symposium on Applied computing, Dijon, France, April 2006, 1695–1699. <https://doi.org/10.1145/1141277.1141677>
- d'Amato C, Fanizzi N, Esposito F (2006) Analogical Reasoning in Description Logics. In: Uncertainty Reasoning for the Semantic Web I. Springer-Verlag Berlin, Heidelberg, 330–347. [https://doi.org/10.1007/978-3-540-89765-1\\_19](https://doi.org/10.1007/978-3-540-89765-1_19)
- d'Amato C, Fanizzi N, Fazzinga B, Gottlob G, Lukasiewicz T (2013) Combining Semantic Web Search with the Power of Inductive Reasoning <http://ceur-ws.org/Vol-527/paper2.pdf> [accessed Aug 15, 2019]
- d'Amato C, Fanizzi N, Esposito F (2009) A Semantic Similarity Measure for Expressive Description Logics. In: International Conference on Flexible Query Answering Systems. Springer, Berlin, Heidelberg, 406–417. [https://doi.org/10.1007/978-3-642-04957-6\\_35](https://doi.org/10.1007/978-3-642-04957-6_35)
- d'Amato C (2007) Similarity-based Learning Methods for the Semantic Web. PhD thesis, Bari, Italy: University of Bari. [http://www.di.uni-ba.it/~cdamato/PhDThesis\\_dAmato.pdf](http://www.di.uni-ba.it/~cdamato/PhDThesis_dAmato.pdf) [accessed Aug 15, 2019]
- Apache Jena (2019) Apache Jena. <http://jena.apache.org/> [accessed Aug 15, 2019]
- Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P (2010) The Description Logic Handbook: Theory, Implementation and Applications. Second Edition. Cambridge University Press, New York, 505 pp. <https://doi.org/10.1017/CBO9780511711787>
- Baader F, Kusters R, Wolter F (2018) Extensions to Description Logics (Concrete domains). In: Semantic Scholar. <http://pdfs.semanticscholar.org/1ec8/069f6341f3800a9ba3200dc92da65a8eb23f.pdf> [accessed Aug 15, 2019]
- Bikakis N, Skourla M, Papastefanatos G (2014) rdf:SynopsisViz – A Framework for Hierarchical Linked Data Visual Exploration and Analysis. In: Proceedings of the European Semantic Web Conference ESWC 2014: Satellite Events, Heraklion, Crete, Greece, May 2014, 292–297. [https://doi.org/10.1007/978-3-319-11955-7\\_37](https://doi.org/10.1007/978-3-319-11955-7_37)
- Bikakis N, Sellis T (2016) Exploration and Visualization in the Web of Big Linked Data: A Survey of the State of the Art. <http://arxiv.org/pdf/1601.08059.pdf> [accessed Aug 15, 2019]
- Bobillo F, Carvalho R, Costa P, d'Amato C, Fanizzi N, Laskey K, Lukasiewicz T, Nickles M, Pool M (2013) Uncertainty Reasoning for the Semantic Web III. In: SWC International Workshops URSW 2013, Revised Selected Papers. <https://doi.org/10.1007/978-3-319-13413-0>
- Camarda D, Mazzini S, Antonuccio A (2012) Lodlive, exploring the web of data. In: Proceedings of the 8<sup>th</sup> International Conference on Semantic Systems, I-SEMANTICS, ACM, Graz, Austria, 197–200. <https://doi.org/10.1145/2362499.2362532>
- Google Cloud Platform (2019) <http://console.cloud.google.com> [accessed Aug 15, 2019]
- Harmelen F, Lifschitz V, Porter B (2008) Handbook of Knowledge Representation (Foundations of Artificial Intelligence), vol. 3. Elsevier Science Oxford, England, 1006 pp.
- Horrocks I, Sattler U (2001) Ontology Reasoning in the SHOQ(D) Description Logic. In: Proceedings of the 17<sup>th</sup> international joint conference on Artificial intelligence, vol. 1, Seattle, WA, USA, August 2001, 199–204. <http://dl.acm.org/citation.cfm?id=1642117> [accessed Aug 15, 2019]
- Knowledge graph (2019) “Nuclear physics at MSU and MEdPhI” presented in serialized format. <http://drive.google.com/file/d/1AIXMs-m3cfAxR6NX220R4ZeFeoSfP0mj5> [accessed Aug 15, 2019]
- Krotzsch M, Simancik F, Horrocks I (2013) A Description Logic Primer, Semantic Scholar <http://arxiv.org/pdf/1201.4089.pdf> [accessed Aug 15, 2019]
- Minervini P, d'Amato C, Fanizzi N, Tresp V (2016) Discovering Similarity and Dissimilarity Relations for Knowledge Propagation in Web Ontologies. Journal on Data Semantics 5: 229–248. <https://doi.org/10.1007/s13740-016-0062-7>
- Muggleton S, Raedt L (1994) Inductive Logic Programming: Theory and methods. The Journal of Logic Programming 19–20: 629–679. [https://doi.org/10.1016/0743-1066\(94\)90035-3](https://doi.org/10.1016/0743-1066(94)90035-3)
- Jena inference (2019) Reasoners and rule engines: Jena inference <http://jena.apache.org/documentation/inference/> [accessed Aug 15, 2019]
- Schlobach S, Janowicz K (2016) Visualizing ontologies with VOWL. Semantic Web 7(4): 399–419. <https://doi.org/10.3233/SW-150200>
- Telnov V (2017) Semantic Educational Web Portal. In: Selected Papers of the XIX International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2017), Moscow, Russia, October 2017, 50–56. <http://ceur-ws.org/Vol-2022/paper11.pdf> [accessed Aug 15, 2019]
- Telnov V (2019) Semantic educational web portal. Graphs of nuclear knowledge. Intelligent search agents (2019) <http://vt.obninsk.ru/x/> [accessed Aug 15, 2019]
- Telnov V, Korovin Y (2019) Programming of Knowledge Graphs, Reasoning on Graphs. Software Engineering 8(2): 59–68. <https://doi.org/10.17587/prin.10.59-68>
- Thellmann K, Galkin M, Orlandi F, Auer S (2015) LinkDaViz – Automatic Binding of Linked Data to Visualizations. In: Proceedings of the 15<sup>th</sup> International Semantic Web Conference, Bethlehem, PA, USA, 147–162. [https://doi.org/10.1007/978-3-319-25007-6\\_9](https://doi.org/10.1007/978-3-319-25007-6_9)
- W3C OWL 2 Web Ontology Language (2012) <http://www.w3.org/TR/owl2-overview/> [accessed Aug 15, 2019]
- Xu J, Shironoshita P, Visser U, John N, Kabuka M (2015) Converting Instance Checking to Subsumption: A Rethink for Object Queries over Practical Ontologies. International Journal of Intelligence Science 5(1): 44–62. <https://doi.org/10.4236/ijis.2015.51005>