On the Issue of Property Transitivity in RDF Datasets

Tatiana Shulga [0000-0002-5521-5960], Alexander Sytnik [0000-0002-1256-7253],
Ekaterina Panteleeva [0000-0002-2693-937X]

Yuri Gagarin State Technical University of Saratov
77 Politechnicheskaya street, Saratov, Russia, 410054
taiss@yandex.ru

Abstract. As part of the development of the concept of the semantic web in recent years have been created a large number of OWL ontologies and RDF datasets based on them. However, in the design and using the properties of OWL ontologies, is extremely important to correctly reflect real-world information, because something that is totally logical in the world of abstract data may poorly correlate with the expected behavior of web applications for the user. For example, SPARQL queries that using the transitive properties of the OWL language can create loops and return incorrect information. In this article we show that in some cases it is preferable to abandon the use of transitivity of properties in ontologies and describe an algorithm for traversing related entities, which allows solving the problem of loops. As an example, illustrating this problem is considered the web application "Linked Open Specialties" (LOS), which sends SPARQL queries to the ontology "Specialties". The ontology “Specialties” represents the structure of official lists of specialties, bachelors and masters graduate programs and research specialties that were valid in recent years in Russian Federation and allows us to establish their correspondence using the transitive property “equalsTo”. Notably, that, although the developed algorithm is formulated and used in terms of a specific subject area to solve the problem of a separate application, it is quite universal and can be used to solve the transitivity problem in RDF-datasets of other subject areas.

Keywords: Transitivity of OWL-properties, ontology, RDF, Semantic Web, recursive algorithm, linked specialties

1. Introduction

The past two decades have been characterized by the rapid development of Semantic Web technologies. [1]. Conceptually, semantic web is a stack (set) of web technologies that allow to store and link data from various sources (systems and documents) in a manner that machine processing is applicable to them. One of the major technologies of this stack is the RDF language, which is used to recording statements about any resources as triplets. This is a flexible data model, which is independent of subject area.

Proceedings of the of the XXIII International Conference "Enterprise Engineering and Knowledge Management" (EEKM 2020), Moscow, Russia, December 8-9, 2020.

However, the capabilities of the RDF language are limited by binary predicates and efficient computer processing of data is possible only when semantics are added to RDF data sets. In practice, for this purpose are developed RDF datasets based on OWL ontologies. In accordance with the definition of the W3C consortium, ontology is understood as a formal model of knowledge representation in a certain subject area that describes types of objects (classes), interaction between them (properties), and ways of joint use classes and properties (axioms) [2]. In recent years, a huge number of RDF datasets based on OWL-ontologies have been developed, some of which are provided in the Linked Open Data Cloud(LOD), and some are described in scientific publications (for example, [3,4,5,6]).

The OWL language implements the basic requirements for ontology languages: clearly defined syntax, formal semantics, sufficient power of expression, ease of expressing knowledge and effective support for logical inference. Support for logical deduction allows to obtain new knowledge using existing knowledge and to detect different kinds of contradictions in ontologies, for example, unforeseen relations between classes or individuals. This process is possible by the mechanism of axioms (restrictions) - a kind of rules that operate in this ontology. Axioms of the OWL language allow us to present “complex” knowledge, for example, to set restrictions of the cardinality (there cannot be more than 20 students in a group), or characteristics of properties, such as transitivity (if lecture1 is included in topic1, and topic1 is included in section1, then this means that lecture 1 is included in section 1).

In this paper, we consider in detail an axiom of the OWL language such as the transitivity of properties and the problem of using transitive properties in RDF datasets. Queries that use transitive properties can create a kind of loops and return incorrect information. In this paper, this problem is described and solved using a specific RDF-dataset as an example. The set is developed on the basis of the OWL-ontology “Speciality”, which is for to present data of the lists of specialties, bachelors and masters graduate programs and research specialties ever operating in the Russian Federation. We propose a method for solving this problem by developing and implementing an algorithm for traversing related entities.

2. Transitivity of properties in OWL

According to the OWL notation, property P is called transitive in the case from the totality of facts that individuals A and B are connected by property P and individuals B and C are connected by property P, it follows that individuals A and C are also connected by property P.

Transitive properties are composite properties because they are created in a few steps. For example, on the basis of the statements “Mari isDescendantOf Tom” and “Tom isDescendantOf Mike”, it may be concluded that “Mari isDescendantOf Mike”.

The fact that transitive properties are composite may be the source of some problems associated with the so-called “loops” which can occur unpredictably in the case of sufficiently long transitive chains. This means that if the connection chain of this property is nonlinear and represents some semblance of graph, then at the moment there is no mechanism for specifying the order of its traversal or any restrictions on it, therefore SPARQL queries to resources associated with this property can return results that do not meet expectations [7]. The situation is also complicated by the fact that, although SPARQL query language has a filtering mechanism, it is applied directly to the query results and cannot affect the chain of logical inference during its execution. Since the results returned by the query are simple resources, it is impossible to obtain any information about the transitive relation traversal order, which means that it is also not possible to carry out any filtering from the outside.

More specifically, this problem can be illustrated by the example of the ontology “Specialties”.

3. Ontology “Specialties”

The ontology “Specialties” [8] represents the structure of the official lists of specialties and areas of training bachelors and masters, operating in recent years in Russia, and a list of scientific specialties. The Ministry of Education and Science approve such lists in the Russian Federation. In recent decades, the higher education system of the Russian Federation has changed significantly. In particular, only from 2004 three official lists of specialties and areas of training bachelors and masters was changed, graduate school became an educational program and conformity of graduate school areas with scientific specialties was established. The Ministry of Education and Science publishes data about these changes on the Internet in the form of orders and instructions, but in practice their use by educational organizations and citizens cause’s difficulties. This is due to the fact that this data are published in the form of pdf-documents for which effective machine analysis is impossible Therefore, there is a need to develop special web applications that would allow governing bodies, educational organizations and private citizens not only to have access to information, but also to analyze it effectively. If such applications are developed on the basis of the traditional approach — using relational databases, then many questions arise related to data openness, their support, changing the structure of databases (after changing the structure of lists), relationships with other educational resources. The solution to this problem was the development of the OWL ontology "Specialties" and the corresponding RDF data set. This set is developed based on official documents of the Ministry of Education and Science of the Russian Federation, available in open access (for example, [8]). It contains the output data of the lists such as the names and codes of specialties and educational programs combined in the UGSN from various lists and their correspondence.

The ontology “Specialties” and RDF dataset were developed by a team of teachers and students of Yuri Gagarin State Technical University of Saratov (SSTU) [9]. They can be accessed in several ways: through the Web application Linked Open Specialties (LOS) [10], or through the SPARQL endpoint [11]. The described ontology is also published in the cloud of open linked dictionaries (Linked Open Vocabulary) [12] and can be used by any developers to create any web applications in the area of higher education of the Russian Federation that require data on former or current lists of specialties and educational programs.

Using the data of this ontology, it is possible by specialty in the diploma and the year of its receipt to determine the relevant specialty or educational program currently in force. That is, in particular, it associates the name and code of the specialty or educational program valid at one time with the name and code of the specialty or educational program valid at another time.

The class hierarchy of this ontology is presented in Figure 1.

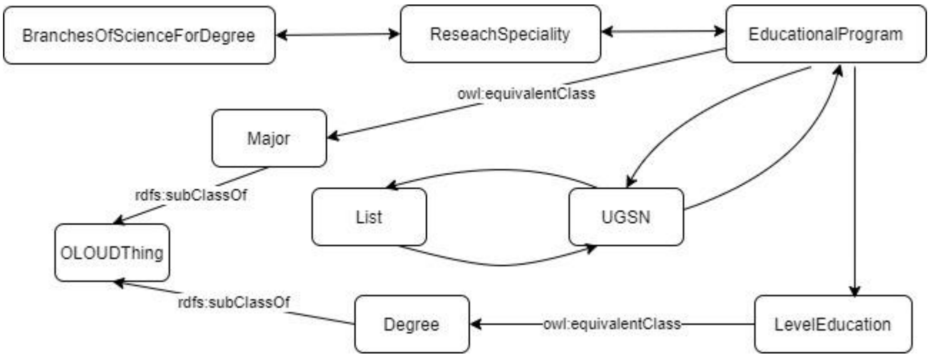


Fig. 1. The structure of the ontology classes “Specialties"

Ontology object properties (properties that connect instances of two classes) are presented in Table 1.

Table 1. Object properties of the ontology “Specialties”

Object property	Designation of property
dcterms:hasPart	Property from the ontology "Dublin Core". Represents a resource that is physically or logically included in the described resource.
dcterms:isPartOf	Property from the ontology "Dublin Core". Represents a resource in which the described resource is physically or logically included in any resource.
partOf (included in)	A property that indicates that a particular object is part of another object. It is transitive and inverse to the “consistsOf” property and is a subclass of a class dcterms: isPartOf.
isPartOfList (included in the list)	Subproperty properties «PartOf», shows that certain UGSN included in a certain list. Domain is UGSN, the range – is a list, and it is the inverse of the property «listConsistsOf».
isPartOfUGSN (included in the UGSN)	Property that indicates that a certain specialty included in certain UGSN. Domain is a specialty, the range – is a UGSN, and it is the inverse of the property «UGSNConsistsOf».
consistsOf (consists of)	A property that indicates that a particular object has specific components. It is transitive and inverse to the property “isPartOf” , is a subclass of the class dcterms: hasPart class.
listConsistsOf (list consists of)	The sub-property of the “consistsOf” property, showing that a particular list has components (UGSN). Domain is the list, the range – is a UGSN, and it is the inverse of the property

	«isPartOfList».
ugsnConsistsOf (UGSN consists of)	The sub-property of the “consistsOf” property, showing that a certain UGSN has components (specialties). Domain is UGSN, the range - is a specialty, and it is the inverse of the property «isPartOfUGSN»,
hasLevelEducation (has level education)	Shows the level of training for a particular specialty. Domain is a specialty, and the range - is a level of education. Also is a functional property.
owl:sameAs	A property from the ontology "Web Ontology Language", showing that two links actually refer to the same object, that is, the objects are identical.
equalsTo (equals to)	Property showing that one specialty from one list corresponds to another specialty from another list. Domain and range - "Specialty". Is transitive and symmetric property, is a subclass of owl: sameAs.
UGSNHasUDC	A property indicating that a specific UGSN corresponds to a specific UDC from the ontology "UDC-Scheme".

4. The problem of transitivity of the property “equalsTo”

From the object properties of the ontology “Specialties”, the most important in the context of this work is the property equalsTo. It shows that one specialty from one list corresponds to another specialty from another list. The domain and range of this property is a specialty. It is transitive and symmetric property and is a subclass of a class owl: sameAs.

The transitivity of this property is necessary in order to establish correspondence through related lists, which is logical when establishing consistency between specialties.

However, the results of SPARQL queries directed at obtaining a list of equivalent specialties did not meet the needs of the task, since they contained the full path of property transitivity. Perhaps this is correct from the point of view of the abstract application of this characteristic of the property, but it does not correspond to the general logic of the LOS application, since it contains a kind of “cycle”. In this case, “loop” means the non-linear course of transitivity in which the output chain returns to the specialty from the same list as the initial specialty for which compliance is establishing. This leads to the fact that the user of the application often gets a result that looks incorrect.

An example of such a result can be seen in Table 2. When requesting correspondences for the specialty "Economics" from the list “OKSO” with the code "080100", output contain such specialties as "Foreign regional studies", "Regional studies of Russia", "Applied mathematics and computer science", which is not the expected result for the user.

Table 2. Fragment of correspondences list for the specialty "Economics"

Educational program	Code	Level education	UGSN	List	Period of validity

Educational program	Code	Level education	UGSN	List	Period of validity
Foreign Regional Studies	032000	Bachelor's degree	Liberal arts	337	2011-2012
Regional Studies of Russia	032200	Bachelor's degree	Liberal arts	337	2011-2012
Regional Studies of Russia	41.03.02	Bachelor's degree	Political sciences and regional studies	1061	2012-our days
Foreign Regional Studies	41.03.01	Bachelor's degree	Political sciences and regional studies	1061	2012-our days
Applied Mathematics and Computer Science	010400	Bachelor's degree	Physics and Mathematics sciences	337	2011-2012
Applied Mathematics and Computer Science	01.03.02	Bachelor's degree	Mathematics and Mechanics	1061	2012-our days
Business Informatics	080500	Bachelor's degree	Economics and Management	337	2011-2012
Business Informatics	38.03.05	Bachelor's degree	Economics and Management	1061	2012-our days
Applied Computer Science	230700	Bachelor's degree	Computer Science and Computer Engineering	337	2011-2012
Applied Computer Science	09.03.03	Bachelor's degree	Computer Science and Computer Engineering	1061	2012-our days

It should be noted that in the above example already including the filtering by specialties, which belong to the same list as the initial specialty. However, as mentioned above, the filters in the SPARQL-query works directly with its results and have no influence on the inference process and the construction of the transitivity chain, therefore, they do not fulfill the required task.

5. A recursive traversal algorithm for related specialties

In order to solve the considered problem, we suggest removing the transitivity restriction on property equalsTo and using the following algorithm that obtains the list of related specialty for a given specialty and returns values, which is the correct from the user’s point of view.

Algorithm to obtain the list of related specialties

This algorithm allows getting the set of all equivalent specialties from other lists by the code of a given specialty.

Input data: code of the specialty s;

Output data: set of specialties A corresponding to the specialty s filed by the input.

The steps of the algorithm:

1. Get the list code p, to which specialty s belongs;
2. Get a list of specialties B, associated with s by property equalsTo;

3. Remove specialty s from list B, if B is contained s;
4. For each specialty s_i from list B do the following steps:
   - 4.1. Get the code of list p_i that contain specialty s_i;
   - 4.2. If p_i corresponds with p, break the current iteration of the loop and go to specialty s_{i+1};
   - 4.3. If the set of specialties A, contains s_i, break the current iteration of the loop and go to specialty s_{i+1};
   - 4.4. Add specialty s_i to the set of specialties A;
   - 4.5. Do steps 2-4 for specialty s_i.

For ease of developing and testing this algorithm, on the base of the ontology “Specialties” a small ontology “Test” was developed and filled with data. It is an abstraction of that part of the original ontology, which will interact directly mechanism for establishing correspondences.

The class hierarchy of the developed ontology is presented in Figure 2 and includes only two classes: EducationalProgramm - the specialty of education, and List - a list that contains specialties.

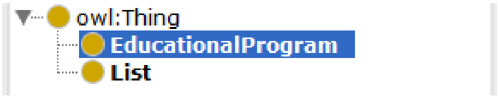


Fig. 2. The structure of the ontology classes “Test”

The object properties hierarchy of the ontology «Test» is presented in Figure 3. The main properties in it are the following properties: equalsTo - establishes the correspondence between specialties from various lists, isPartOfList - shows that the specialty is part of the list, listConsistOf - shows that the list consists of components - specialties.

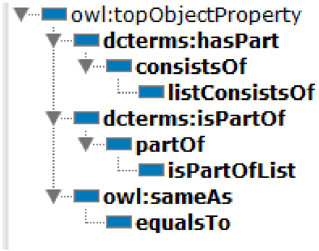
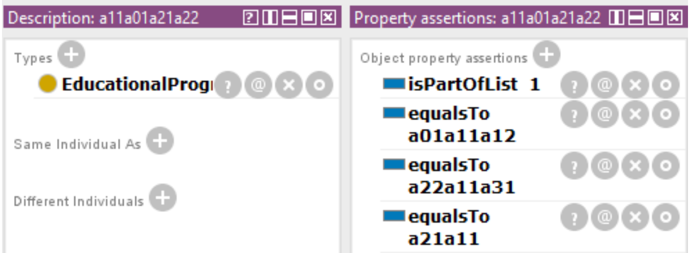


Fig. 3. The structure of the ontology object properties “Test”

The resulting ontology was filled with data about individuals - specialties, and for the convenience of testing the name of each contains names of those specialties with which it is directly related by the property equalsTo. An example of a description of such specialty is presented in Figure 4.



Description: a11a01a21a22

Types

EducationalProg

Same Individual As

Different Individuals

Property assertions: a11a01a21a22

Object property assertions

isPartOfList 1

equalsTo a01a11a12

equalsTo a22a11a31

equalsTo a21a11

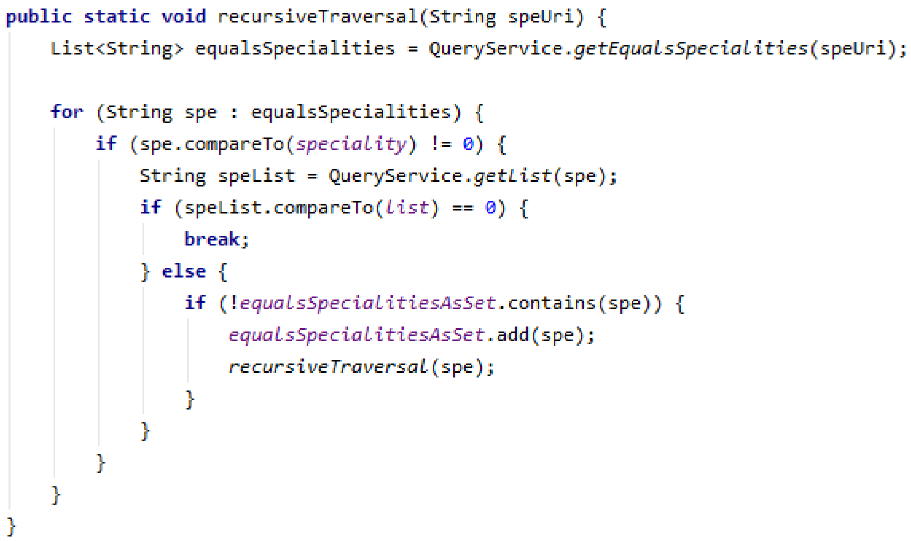
Fig. 4. Description of the specialty a11a01a21a22

The next step was the development of a console application that implements this algorithm in Java. The main reason for choosing this programming language was the fact that LOS application is written in this language and thus further integration of the developed mechanism would be the most simple.

The following classes were created during application development:

- App, directly responsible for the logic of establishing correspondence between specialties and the formation of the required result;
- SparqlQuery containing the texts of all used SPARQL-queries;
- QueryService, necessary for substitution of variable values and correct formation of SPARQL queries before sending to the server;
- Util, responsible for connecting to a SPARQL-endpoint, sending SPARQL-queries and receiving their results.

The App class requires special attention, since the algorithm for obtaining the list of correspondences for the specialty is implemented precisely in it. The main function in it is recursiveTraversal, a recursive function to bypass the list of specialties related by property equalsTo, its code is shown in Figure 5.

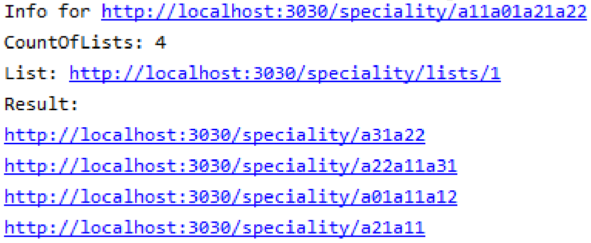


```
public static void recursiveTraversal(String speUri) {
    List<String> equalsSpecialities = QueryService.getEqualsSpecialities(speUri);

    for (String spe : equalsSpecialities) {
        if (spe.compareTo(speciality) != 0) {
            String speList = QueryService.getList(spe);
            if (speList.compareTo(list) == 0) {
                break;
            } else {
                if (!equalsSpecialitiesAsSet.contains(spe)) {
                    equalsSpecialitiesAsSet.add(spe);
                    recursiveTraversal(spe);
                }
            }
        }
    }
}
```

Fig. 5. Function recursiveTraversal

The result of work of the application for the specialty "a11a01a21a22" is presented in Figure 6. All necessary correspondences were obtained, including those whose connection by the property equalsTo with this individual was not setup directly, but was obtained as a result of a recursive traversal of related specialties.



```
Info for http://localhost:3030/speciality/a11a01a21a22
CountOfLists: 4
List: http://localhost:3030/speciality/lists/1
Result:
http://localhost:3030/speciality/a31a22
http://localhost:3030/speciality/a22a11a31
http://localhost:3030/speciality/a01a11a12
http://localhost:3030/speciality/a21a11
```

Fig. 6. The result of work of the application

The integration of this algorithm in the LOS application will provide correct results when performing a request to search for specialties that correspond to the specialty "Economics" from the list “OKSO” with the code "080100" (Fig. 8.)

Table 3. Correspondences list for the specialty "Economics"

Educational program	**Code**	**Level education**	**UGSN**	**List**	**Period of validity**
Economics	521600	Bachelor's degree	Liberal arts and socio-economic science	The List	2001-2004
Economics	080100	Bachelor's degree	Economy and Management	337	2011-2012
Economics	032200	Bachelor's degree	Economy and Management	1061	2012-our days

6. Conclusion

Thus, we investigated the problem of property transitivity in RDF-datasets using a specific example. To do this, we analyzed the work of the LOS application that performs SPARQL queries to the RDF-dataset, which was based on the ontology "Specialty". As a result, a problem was revealed in constructing the output chain of the transitive property equalsTo: queries using this property returned an incorrect result containing "loops". A recursive traversal algorithm for related specialties, described in this article is correct and can be successfully implemented in the application “Specialties of higher education of the Russian Federation” to solve the problem of transitivity of the property equalsTo.

In the future, the re-engineering of the application “Specialties of higher education of the Russian Federation” will be held using this algorithm, for which the developed Java application that implements this algorithm will be useful.

Notably, that, although the developed algorithm is formulated and used in terms of a specific subject area to solve the problem of a separate application, it is quite universal and can be used to solve the transitivity problem in RDF-datasets of other subject areas.

References

1. A Semantic Web Primer (Cooperative Information Systems series) 3rd (third) Edition by Antoniou, Grigoris, Groth, Paul, van Harmelen, Frank van. Published by The MIT Press (2012).
2. Linked Data Glossary. W3C Working Group Note 27 June 2013. [Электронный ресурс] URL: http://www.w3.org/TR/2013/NOTE-ld-glossary-20130627/#ontology (accessed on 30.09.2020).
3. Guarino, N., Musen, M. Applied ontology: The next decade begins (2015) Applied Ontology, 10 (1). pp. 1-4.
4. Schulz Stefan The Role of Foundational Ontologies for Preventing Bad Ontology Design. (2018) CEUR Workshop Proceedings, 2205.
5. Shulga, T., Sytnik, A., Kumova, S., Isaev, D. Web service for the dissertation opponents selection based on ontological approach (2019) CEUR Workshop Proceedings, 2413. pp.145-151.
6. Kelle Pereira, Crystiam & Siqueira, Sean & Pereira Nunes, Bernardo & Dietze, Stefan. (2017). Linked data in Education: a survey and a synthesis of actual research and future challenges. IEEE Transactions on Learning Technologies. 1-1. 10.1109/TLT.2017.2787659.
7. Fionda, Valeria & Pirrò, Giuseppe & Consens, Mariano. (2019). Querying knowledge graphs with extended property paths. Semantic Web. 10. 1-42. 10.3233/SW-190365.
8. Order of the Ministry of Education of the Russian Federation of December 4, 2003 N 4482 “On the Application of the all-Russian Classifier of Specialties in Education”. Available online: https://www.vyatsu.ru/uploads/file/1403/prikaz_minobrazovaniya_rossii_perehodnik_okso.pdf (accessed on 30.09.2020)

9. Sytnik A.A., Shulga T.E. Ontological engineering knowledge in the field of higher education of the Russian Federation // Engineering enterprises and knowledge management (IP & UZ-2018): collection of scientific papers of the XXI-th Russian scientific conference. April 26-28, 2018 / under scientific. ed. Yu. F. Telnova: in 2.t. - Moscow: FGBOU VO "REU them. G. V. Plekhanov", 2018.T1. Pp. 234-239. ISBN 978-5-7307-1359-8. (v.1)
10. SPARQL-endpoint to the ontology "Specialties". Available online: http://sparql.sstu.ru:3030 (accessed on 30.09.2020)
11. Web application "Specialties of higher education of the Russian Federation". Available online: http://los.sstu.ru (accessed on 30.09.2020)
    Ontology "Specialties" in the open dictionary of related data LOV. Available online: http://lov.okfn.org/dataset/lov/vocabs/losp (accessed on 30.09.2020)