УДК004.6 http://doi.org/10.15407/pp2021.04.019

O. Novytskyi

METHODS AND TECHNIQUES FOR MANAGEMENT OF ONTOLOGY-BASED KNOWLEDGE REPRESENTATION MODELS IN THE CONTEXT OF BIG DATA

Ontology-based knowledge representation models in the context of big data are one way to reduce complexity for data processing across methods of semantic description. This research paper aims to provide an overview of the methods and techniques for efficient management of the ontology-based models that improve big data systems. For this case, the shapes constraint language (SHACL) for information validation was reviewed as the key method. The knowledge representation systems and reasoners are studied and reviewed in the paper as well. The author describes approaches based on ontologies in the context of big data. The proper management of ontology-based knowledge representation models through offered methods and techniques brings improved data integration, big data quality, and business process integration.

Key words: ontology-based model, ontologies, big data, reasoners, representation system, ontologies, shapes constraint language, information validation, ontology-based knowledge representation models.

Introduction

Big data means complex data sets that are unable to process adequately via traditional data applications. Big data management is handled by special-purpose resource planning systems called enterprise information systems. These systems represent business processes adequately and force the overall cost-effectiveness [1]. Modern enterprises are focused on the enterprise-wide centralized information system to validate and integrate large amounts of complex data. To capture and represent complex and big data, ontology-based knowledge representation models are used. One of the factors which impact big data processing is the complexity of understanding the data. Semantic technology is allowed to automatically recognize data. This article is to explain the approach of using ontologies for big data to ensure a common understanding of information. The ontology-based knowledge representation models make explicit domain assumptions [2].

Querying information in the context of big data becomes accessible for large enterprises. Ontologies bring detailed and meaningful distinctions between relationships, classes, and properties. The paper is devoted to ontology-based modeling and its management in semantic graph databases. Big data quality is improved with the help of ontology-based knowledge representation models and the reasoners that enable consistency and satisfiability checks [3].

The research paper also reviews an alternative using ontologies to model data. SHACL (shapes constraint language) is overviewed to demonstrate the benefits of this method for information validation in the triplestore and for validating RDF graphs against a set of constraints. The overview of the OWL reasoners and RDF graph capture systems is used as the guide for big data players (large enterprises, structure that is in the stage of developing a large-scale centralized database) on how to manage ontology-based models and improve the data quality with the help of automated reasoning of the information in the semantic graph database.

OWL Reasoners for ontologies.

The research paper reviews the main two reasoners with the wide range of optimizations that benefit big data improvements. They contain updated algorithms and tableaux algorithms that are native to the ontology-based knowledge representation models.

FaCT++ is one of the newest reasoners that is designed to implement tableaux algorithms and updated heuristic optimization techniques. The table of characteristics of the FaCT++ reasoner is given below.

Description	A new highly-optimized reasoner with tableaux-based SROIQ algorithms
License	LGPL v2
Semantics	OWL DL Classification OWL EL Classification OWL DL Consistency OWL EL Consistency OWL DL Realization OWL EL Realization

Table 1. FaCT++ Characteristics. Source: ORE

The FaCT++ reasoner implementation starts with the preprocessing stage. It is applied to the knowledgebase and can be transformed according to the internal representation requirements. FaCT++ performs classification. With the help of applied optimizations, the FaCT++ reasoner is used to reduce the quantity of subsumption tests to be performed [4].

The main application of the FaCT++ optimizations is to transform concepts into SNF. The simplified normal form lets users implement negation, conjunction, universal restriction, at-most restrictions. The main FaCT++ features for big data optimization are:

(a) Absorption – is suitable for rewriting optimization. There are concept and role absorption techniques to take into consideration. The concept absorption is responsible for GCIs elimination via concept definition axioms. The role absorption eliminates GCIs in the concept-free mode.

(b) TCE (Told Cycle Elimination) – the technique for text optimization. This cycle is often eliminated together with definitional cycles. The user can undertake TCE and definitional cycles with the help of axiom transformations.

(c) Synonym Replacement – this FaCT++ technique aims at extending simplification properties. Synonym Replacement improves clash detection in the early stage. The knowledgebase is transformed in the context of synonym elimination with the help of axioms.

The FaCT++ reasoner is used for satisfiability checking optimizations. New ordering heuristics are available for the implementation of new optimization methods. There is a special-purpose To-Do list. The user can force entry assortment with the help of the FaCT++ To-Do algorithm. It is worth noting that the reasoner provides the Backjumping optimization. The tree label matters when the dependency set of information items is formed. Boolean optimization that is available with the help of the FaCT++ reasoner allows users to implement constant propagation (BCP) [5].

HermiT is the reasoner for ontology-based knowledge representative models. It is used for the identification of subsumption relationships (between classes and other specifications). This reasoner is public and available for the users without restrictions. The notable feature of the HermiT reasoner is its new versions with the updated reasoning algorithms [5], [7].

The main HermiT characteristics are given in the table below [8]

Description	The conformant reasoner for the ontology-based knowledge representation models. The HermiT uses direct semantics. It is based on the hyper-tableaux algorithms.
License	LGPL 3.0
Semantics	OWL DL Classification, OWL EL Classification OWL DL Consistency OWL EL Consistency OWL DL Realization OWL EL Realization

Table 2. HermiT Characteristics

The HermiT reasoner allows users to classify the ontology-based knowledge representation models faster. The manual classification often takes hours. The reasoner makes it possible to classify even big data knowledgebase and complex information for minutes.

The HermiT reasoner uses direct semantics for optimization processes and hypertableaux algorithm implementation. The last version of the reasoner is called HermiT 1.3.8. Besides the main function of DL Safe rule handling, the new version of reasoner allows big data players to add new rules directly to the ontology-based models [6].

The number of optimization techniques of the HermiT reasoner is similar to the FaCT++ one described in the research paper above. The

significant feature of the HermiT reasoner is the high-level DL Safety rules compliance. DL Safety rules will be considered incomplete if:

a) the knowledgebase contains property chains in the rule bodies;

b) the KB includes transitivity axioms in the rule bodies of the knowledgebase;

c) the complex properties are used in the rule bodies of the ontology-based model.

The HermiT reasoner is one of the newest reasoners that is recommended for ontology-based knowledge representation model management in the context of big data. The direct semantics use and hyper-tableaux algorithm approach improve the quality of data and simplify business processes related to ontologies and semantics.

RacerPRO is the improved version of the former Racer knowledge representation system. As above-described reasoners and other programs suitable for ontology-based knowledge representation model management, RacerPRO is used for optimized tableaux algorithm implementation. The description logic of $\mathcal{SHIQ}(\mathcal{D}^{-})$ is used for this knowledge representation system.

Description	Racer is a knowledge representation system that implements a highly optimized tableau calculus for a very expressive description logic. It provides the reasoning for T-boxes and A-boxes as well.
License	-
Semantics	OWL DL Classification, OWL EL Classification OWL DL Consistency OWL EL Consistency OWL DL Realization OWL EL Realization

Table 3. HermiT Characteristics

The RacerPRO license is BSD 3-clause. This system is required for big data projects because it is the separate-standing knowledge representation system for solving main reasoning problems [9].

The reasoning procedure takes place in the streaming model that is suitable for complex data proceedings. Both T-boxes and A-boxes often include issues to solve when it comes to knowledge representation. RacerPRO solves these reasoning problems with the help of standard tableaux algorithms and unique interference services (e.g. logical abduction). The architecture of the latest version of the RacerPRO system is presented in Figure 1.

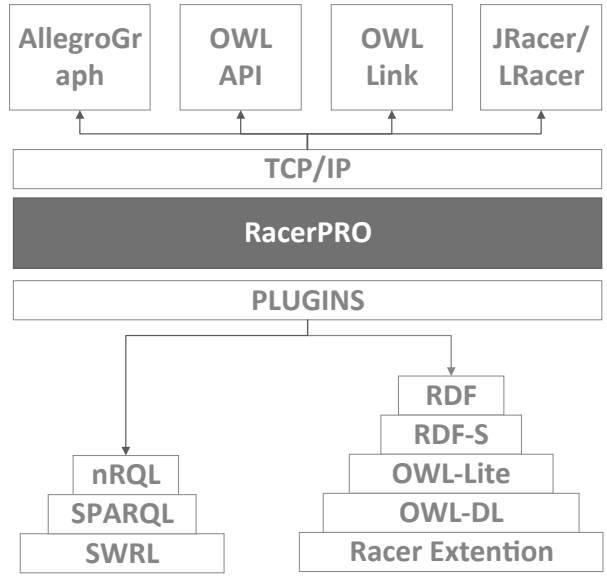

Figure 1. RacerPRO Architecture.

The additional benefit of the RacerPRO reasoning and knowledge representation system is its query language called nRQL. Using the new Racer Query Language means supplementary assistance when it comes to ontology-based model management:

attribute values of different individuals;

improved propertied for string attributes;

negation-as-failure support.

Reasoning over ontology as a rule is a complex task. In real tasks for reasoning, we have to store facts in RDF. Therefore, in the next part, we make a short review semantic reasoner with RDF storage.

Snorocket. This is the special-purpose algorithm based on the healthcare terminology classifiers. Snorocket will be suitable for big data projects related to the clinical, medical, healthcare, science directions [10]. Snorocket is not a multifunctional solution for ontology-based knowledge representation model management. This is suitable for working with ontology related to medical data only.

Snorocket is available for users in the extension format. The classifiers of the algorithm allow healthcare representatives to manage semantic data related to medi-

cal terminology. Big data projects based on healthcare or medical content, imagery, and other information can benefit from using Snorocket. Nevertheless, this extension with the implementation of the unique Dresden algorithm is not suitable for any other knowledgebase. The limited ways of the application make Snorocket the last RDF store system in the list of top ones overviewed in the research paper.

Methods for RDF graphs validation. Shapes Constraint Language (SHACL)

RDF is a main part of the Semantic Web. Its simple data model provides powerful expressiveness which can be applied to represent information in any scope. Practical Semantic Web applications require some technology to describe and validate the RDF data [11]. One of such technology for RDF is SHACL [12], [13], which has developed to model some restrictions in the form of constraints on data.

The shapes constraint language (SHACL) is considered as the alternative to traditional ontologies that are used for data modeling. SHACL is used for RDF graphs validation. There is a set of constraints that are applicable to the validation process. SHACL includes shapes that specify metadata according to its resource. The big data knowledgebase is compliable with the shapes constraint language. The special-purpose shape specifies the resource in the context of big data as well. This resource can be the principle of data use, the reason for data use, and the frequency of data use. The SHACL data validation process is applicable for both unavailable and available data in the triplestore. The shape constraint language conditions are called shapes expressed in the RDF graph format. The main purpose of the SHACL data validation is to check information according to the range of conditions. Those pieces of data that meet the shape constraints can be viewed as a description of data graphs. It is worth noting that SHACL-generated descriptions based on the shape constraint language validation of graphs can be used out of the validation process [12].

It makes SHACL the key method for ontology-based knowledge representation model management. The ready-done descriptions with the help of shape constraint language validation algorithms can be implemented in the context of big data:

- for code generation;
- for data generation.

These descriptions are suitable for code building that is one more technique out of the validation process. The separate-standing aspect to take into account is the relationship between SHACL and RDFs inferencing. The shape constraint language includes the property entailment to identify the interference specifications. To protect the knowledgebase items and bring a smooth validation process, it is recommended to use only verified RDF resources to proceed in SHACL RDF-based technologies [13].

The SHACL validation is recommended for big data because, in comparison with the standard ontologies and semantic techniques, this is the efficient way to avoid ontology limitations (limited set of property constructs). The RDF resource validation is suitable for ontology-based knowledge representation models in the context of big data for its shape-generated failure determination and data improvement properties.

Reasoners with built-in RDF store features

The range of special-purpose databases for graphs that store triples is called the RDF database. It is worth noting that triples or RDF databases are considered as data points. These points are represented in the SPO relationship (subject-predicative-object relationship). All the data items are stored in the same format – a triple format. The database receives and uses information and stores it in triple form. The RDF database is suitable for ontology-based knowledge representation management in the context of big data because all the complex information is well-organized with the help of triple sets.

One more reason to use the RDF database as the ontology-based model management when it comes to big data is the convenience to display information in graphs provided by this type of database. To carve the graphs from the triple database, any query language is used. The functionality and flexibility of the RDF database benefit enterprise-centric knowledgebase and big data projects.

Not all the databases can be included in the category of triple ones. There is a range of requirements for the digital product to be named as the RDF database. The main features of the triple database to potential-to-inclusion products are:

a) sufficient data storage is provided;

b) data as recorded as triples;

c) users are allowed to retrieve the data with the help of query language.

These are features of average RDF store. For the purpose to determine the most efficient triple databases for ontology-based knowledge representation model management.

HyLAR is the special-purpose reasoner for ontology-based knowledge representation models that contains RDF-based libraries. These libraries obtain a wide range of functionality for ontology-based model management. The HyLAR reasoner can be considered as the supplementary reasoning engine for big data. Its rdfstore.js, SPARQLs, and RDF-ext libraries are used as the triple databases [14].

The HyLAR reasoner is available in three versions to implement for the knowledgebase:

a) NPM module

b) A server-based solution

c) Browser version.

The HyLAR reasoner with its RFD libraries supports business database rules. This is one more reason to use the HyLAR-based database for big data projects. The database processing generated by the HyLAR reasoner and its RDF-based libraries is presented in the infographics below [14].

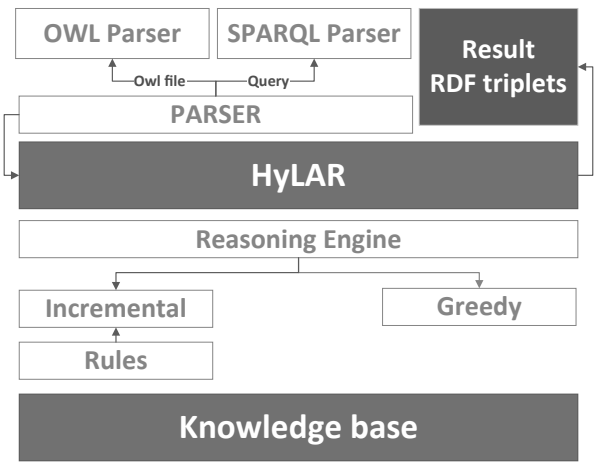

Fig 2. HyLAR Architecture.

HyLAR is used as the reasoning engine combined with the OWL and SPARQL parsers. The reasoner brings results in the format of triples that is well seen on Fig 2. The knowledgebase with ready-done available triples can be applied together with the HyLAR reasoning engine for conversion and creation of enterprise-centralized big data projects with qualitative and checked information.

Apache Jena. The Jena open-source Java framework includes a special-purpose RDF API. Jena contains information only in the format of RDF triples. The collection of RDF triples forms the general database and is included in the Jena data structure called *Model*. Jena is optimal for ontology-based knowledge representation model management because the data structure model of this framework easily determines PDF graphs and provides them relations. The database is well-structural and easy-to-navigate which is required for big and complex data [15].

The way on how the relationships go in the one-direction mode through the triple coding exemplified in Fig 3 [16].

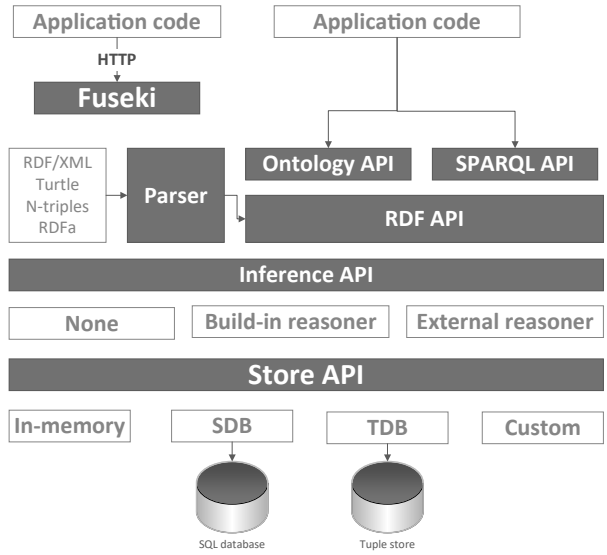

Fig 3. Jena Architecture

Another benefit of Jena in the context of big data is the availability of both RDF and Ontology APIs. The distinct concepts of the framework with the RDF-based triple collection are its opportunity to build up direct relations between graphs (nodes) in the structure, rich-in-functions APIs that provide sufficient management tools for the ontology-based knowledge representation models, and big data

orientation with Jena's in-memory structures in combination with intended methods of the complex data simplification.

RDFox is the semantic reasoning engine with the functionality of the RDF triple store. This one of the core systems for big data with its unique conception of shared memory parallel reasoning [17].

RDFox is notable with memory-economical properties that are suitable for enterprise-centric knowledgebase and big data projects. About 1.5 billion triples can be stored in 50 GB of the RDFox RDF store. The following table presents the main characteristics of the RDFox reasoning engine.

Description	The latest version of the former RDFox semantic reasoning engine. The latest version was launched in 2021. It contains a triple store that is suitable for knowledge representation purposes.
Additional Features	Rule reasoner OWL reasoner RDFS Reasoner
Semantics	RDF OWL SPARQL

Table 4. RDFox Characteristic.

The ontology-based knowledge representation model management in the context of big data can be undertaken through the RDFox semantic reasoning engine. The key benefit of the system is its triple store with memory-efficient stock. Additionally, big data projects can benefit from RDFox named graphs, Data-log extensions, and incremental update & aggregation.

Conclusions

The ontology-based knowledge representation models bring strong benefits to the digital world. The big data sphere is not the exception. Essential relationships between concepts automated data reasoning, and semantic advantages are key benefits of the ontology-based models for big data projects. But ontologies require proper management when it comes to accurate knowledge representation. The current research paper faces the problematics of the poor ontology-based knowledge representation model management in the context of big data.

The most top-ranking systems, reasoners, and other digital solutions were overview by the author. The best ones in the article are described in the research paper. Besides theoretical information given in the general paragraphs about reasoners, shape constraint language, and triple database, there is an analytical background for each reviewed solution. Pros & cons are presented under the description of the reasoners and reasoning engines. According to the undertaken research, the ontology-based knowledge representation models in the context of big data can be easily managed by the reasoning engines, reasoner extensions, query languages, hyper-tableaux algorithms, SHACL implementation, RDF database usage. The future prospects of digital transformation and new technique and method development with a focus on big data are real. The ontology-based knowledge representation models are successfully managed by the digital solutions now. The huge progress in the big data-driven direction is predicted over the coming decade.

Reference

[1] B. Mouad, «An Evaluation and Comparative study of massive RDF Data management approaches based on Big Data Technologies,» *International Journal of Emerging Trends in Engineering Research,* т. 7, pp. 48-53, 2019.

[2] Y. Sure-Vetter, S. Staab та R. Studer, «Methodology for Development and Employment of Ontology Based Knowledge Management Applications.,» *ACM SIGMOD Record 3,* т. 4, № 31, pp. 18-23, 2002.

[3] P. Haase та L. Stojanovic, «Consistent Evolution of OWL Ontologies,» *The Semantic Web: Research and Applications,* pp. 182-197, 2005.

[4] T. Dmitry , «Incremental and Persistent Reasoning in FaCT++,» в *ORE*, 2014.

[5] T. Dmitry та H. Ian , «FaCT++ description logic reasoner: System description,» в *International joint conference on automated reasoning*, Berlin, 2006.

[6] R. Shearer, B. Motik та I. Horrocks, «HermiT: A Highly-Efficient OWL Reasoner,» *Owled,* т. 432, p. 91, 2008.

[7] Data and knowlege group. University of Oxford., «HermiT OWL Reasoner,» [online]. Available: http://www.hermit-reasoner.com/. [Date: 05 2021].

[8] D. Michel , G. Birte , G. Rafael , H. Matthew, J.-R. Ermesto, N. Matentzoglu та P. Bijan , «ORE Live Competition,» 05 2021. [online]. Available: http://dl.kr.org/ore2015/vip.cs.man.ac.uk_8008/reasoners.html.

[9] V. Haarslev, «The RacerPro knowledge representation and reasoning system.,» *Semantic Web,* т. 3, № 3, pp. 267-277, 2012.

[10] M. J. Lawley та C. Bousquet, «Fast classification in Protégé: Snorocket as an OWL 2 EL reasoner.,» в *6th Australasian Ontology Workshop (IAOA'10). Conferences in Research and Practice in Information Technology*, 2010.

[11] G. Jose Emilio Labra, «Validating and Describing Linked Data Portals using RDF Shape Expressions.,» в *LDQ@ SEMANTICS*, 2014.

[12] J. Corman, J. L. Reutter та O. Savković, «Semantics and validation of recursive SHACL,» в *International Semantic Web Conference*, Cham, 2018.

[13] W3C, «Shapes Constraint Language (SHACL),» 20 July 2017. [online]. Available: https://www.w3.org/TR/shacl/. [Date: 05 2021].

[14] M. Terdjimi, M. Lionel та M. Mrissa, «Hylar: Hybrid location-agnostic reasoning.,» в *ESWC Developers Workshop 2015*, 2015.

[15] A. Ameen, K. Ur Rahman Khan та R. B. Padmaja, «Reasoning in semantic web using Jena.,» *Computer Engineering and Intelligent Systems,* т. 5, № 4, pp. 39-47, 2014.

[16] Apache Software Foundation, «Jena architecture overview,» [online]. Available: https://jena.apache.org/about_jena/architecture.html. [Date: 05 2021].

[17] Oxford Semantic Technologies, «RDFox,» [online]. Available: https://www.oxfordsemantic.tech/product. [Date: 05 2021].

Received: 27.10.2021

About author:

Oleksandr Novytskyi,
PhD, Researcher.
Number of scientific publications
in Ukrainian journals – 13.
https://orcid.org/0000-0002-9955-7882.

Affiliation:

Інститут програмних систем
НАН України,
проспект Академіка Глушкова, 40.
Тел.: 526 5139
E-mail: alex.googl@gmail.com