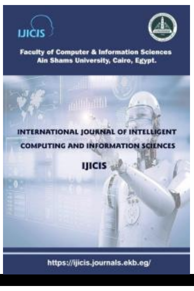
IJICIS, Vol.22, No.2, 74-83
DOI: 10.21608/ijicis.2022.110450.1144

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/



ONTOLOGY-BASED DATA ACCESS TO HETEROGENEOUS DATA SOURCES: STATE-OF-THE-ART APPROACHES AND APPLICATIONS

Naglaa Fathy* Walaa Gad Nagwa Badr Mohamed Hashem

Department of Information Systems, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt

naglaa_fathy@cis.asu.edu.eg walaagad@cis.asu.edu.eg nagwabadr@cis.asu.edu.eg mhashem@cis.asu.edu.eg

Received 2021-12-09; Accepted 2022-03-15

Abstract: *The evolution of heterogeneous data residing in various data sources (e.g., relational, XML, document stores, etc.) increases the data integration challenges. Ontology Based Data Access (OBDA) is a semantic web technology that comprises a set of algorithms and techniques for dealing with data heterogeneity. Ontologies are utilized in OBDA to provide a conceptual view over diverse datasets; and the relationships between them are defined through mappings in two ways: data translation, and query translation. The first method is referred to as materialization, where data transformation is achieved in accordance with the global view. Whereas in the second method, query transformation is carried out from the query language of the global schema into the original data source's query language. In this paper, we present the framework of OBDA by discussing the main components of ontology-based data access, techniques, applications and future challenges.*

Keywords: *Data Integration, OBDA, Mapping Layer, Query translation.*

1. Introduction

The most widely used model for manipulating structured data has been the Relational Database Management System (RDBMS) for almost four decades. However, this kind of databases is not capable of storing and retrieving huge amount of data in a scalable manner. This leads to requiring a new generation of databases that has the ability to store and access terabytes of data in a timely manner. Therefore, a variety of non-relational databases has arisen with different data formats e.g., MongoDB, Neo4j, Cassandra, Couchbase, etc.

***Corresponding Author**: Naglaa Fathy

Department of Information Systems, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt

Email address: naglaa_fathy@cis.asu.edu.eg

Data Integration becomes more challenging when organizations started to provide access to their data via Web technologies (through Linked Data approaches and Semantic Web or using Web services). It is even more relevant when public administrations began publishing open data in accordance with public-sector information reuse initiatives. For this purpose, Ontology based Data Access (OBDA) semantic technology has emerged to tackle the problem of data integration.

In OBDA [1][2], a unified view over diverse data sources is represented by a global ontology. A mapping layer is located between the global ontology and the local data source to describe the relationships between both sources. This is done using mapping languages introduced for relational and non-relational data representations. A uniform access to data from various sources is achieved using queries written in a single query language called SPARQL that uses ontological terms, whereas data can be accessed either physically or virtually. In the first method, the entire data are transformed into RDF [3], based on the defined mappings and then stored in RDF triple stores. In the second method, data is kept in its original form and format, whereas on-the-fly query translation is carried out by mapping the query terms to the data schema.

In this paper, we explain the framework of OBDA and discuss main techniques and applications. In section 2, we describe the principles of OBDA. In section 3, we elaborate the process of OBDA query answering. In section 4, we present some real-world applications to OBDA. Conclusion and research directions are outlined in section 5.

2. Principles of OBDA

A classical OBDA system is represented by the tuple ⟨ G, S, M ⟩, as shown in Figure 1 [4]: the global schema G represents the unified view, the local schema S represents the data source, and the mapping M specifies correspondences between concepts of the local and global schemas. These concepts may be relational tables, ontological classes, CSV data, etc.; depending on the modeling languages of both schemas.

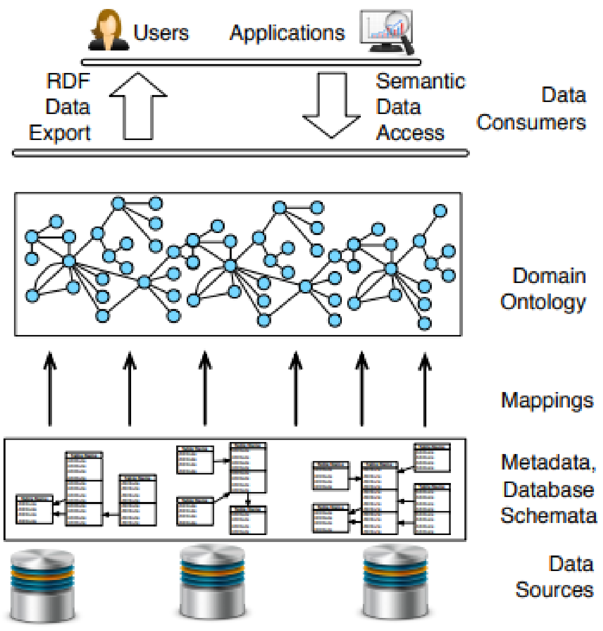


Figure 1. General Architecture of OBDA [4]

Over the previous few decades, various mapping languages of different formats and syntaxes have been proposed for OBDA [5]. Several mapping languages have been proposed to transform relational databases into the standarized RDF and OWL languages (e.g. D2R, R2O). Concequently, two recommendations were publised by the the RDB2RDF W3C Working Group for relational to RDF content transformation: Direct Mapping [6] and R2RML [7]. Simple transformations are specified in the Direct Mapping approach with no additional user intervention. On the contrary, in R2RML, transformation rules should be specified (for instance, the colums to be used for the transformation, how to generate URIs, etc.). Subsequently, new needs arose to support formats other than relational databases. As a result, RML [8] and xR2RML [9] mapping languages were proposed to deal with XML, CSVs and JSON data sources, and MongoDB document database, respectively.

Two approaches have been proposed in terms of how a mapping is expressed [10]. The first approach is Global-as-View (GAV) that is expressed in terms of local schema queries (or views). Conversely, the Local-as-View (LAV) approach is expressed in terms of global schema queries (or views). We discuss the main characteristics of both approaches and their relevance in different contexts in the following subsections.

2.1. Global-as-View Mapping Approach

Global-as-View (GAV) approach is based on defining each concept of the global schema in terms of a view (query) over data sources [11]. Conceptual simplicity is the main advantage of the GAV mapping, as it explicilty defines how to unfold a query over the global schema. This is done by simply substituting the global schema concepts with the corresponding definitions. Additionally, evolution of the global schema is easily supported by the GAV approach, since the mapping specification of each concept is independent of the others. Therefore, changing a concept definition or adding a new one to the global schema, involves updating or writing the mappings for that specific concept only, without affecting existing mappings.

Conversely, several drawbacks exist in GAV approach. These include:

- When writing the mapping of a concept of the global schema, all local schemas should be known in order to identify the relevant sources to this concept.
- When a new source is added, the mappings of numerous concepts may require updating to include the contributing source.
- Fault-tolerance is not provided by GAV approach; since the absence of one data source might lead to system failure, because it cannot process the concepts of the global schema that involve that source.

Consequently, it is widely acknowledged that the GAV approach is best suited to a small number of data sources, where it is uncommon to add, change or remove a data source. It is commonly argued that adding a new source might harm the system whenever changes are required for the global schema. However, this depends on the context of the system being integrated, as illustrated in the following cases [12]:

- The global schema must be updated when adding a new data source related to a domain not covered by a global data model created earlier to integrate numerous databases for an enterprise. In this case, various mappings need to be updated as well.
- The global schema that defines a knowledge domain is kept unchanged whenever a relevant data source is integrated. This is because mappings to the global schema are written to only relevant source concepts, while ignoring irrelevant ones.

2.2. Local-as-View Mapping Approach

The Local-as-View (LAV) approach is based on defining each concept of the local schemas in terms of a view (query) over the global schema [11]. Unlike the GAV approach, scalability is the main advantage of the LAV mapping approach because each data source is described independently of the others, which facilitates the addition of any number of sources. Furthermore, the absence of a single data source does not lead to system failure, because data sources are independent, so other data sources can still yield the global schema concepts.

There are two major drawbacks in the LAV approach [12]:

- Updating a concept definition or adding a new concept to the global schema might lead to mappings updating of several data sources. In practice, when considering a high number of sources, this process may become impossible.

- When the same global concept is shared by many sources, the computational complexity of query process increases dramatically. The number of query rewritings might be exponential in the size of the query and the number of views that provide data for the same global concept. Therefore, the LAV approach is convenient when only one data source (or a small subset of sources) could provide data for each concept of the global schema.

For a better characterization of source databases with respect to the global schema, mapping assertions for GAV and LAV approaches are interpreted in three different views [10]:

- Sound views: where a subset of the expected data matching the global schema view is provided by the local source.
- Complete views: where a superset of the expected dats matching the global schema view is provided by the local source.
- Exact views: The corresponding view is satisified by the exact set of tuples from data source D.

3. OBDA Query Answering Approaches

As mentioned that in an OBDA system (G, S, M), there are two approaches for answering a query q posed over the global schema G: materialization and virtualization. Materialization refers to generating RDF graph of the ontology that represents the global schema G from the source database S according to mapping M. The main drawback of this approach is whenever the data source has updates, it implies reapplying the data transformation process to reflect changes.

Conversly, the second approach does not materialize RDF graph. Instead, a new query over the source schema S is obtained from translating the original query q defined over the global schema G, through mappings M. This approach comrpises three main steps: rewriting, unfolding, and evaluation. At first, the submitted query is rewritten according to the global schema (ontology). The resuting query is then translated into another query over the source schema S using the mapping M. Finally, the translated query is evaluated over the source schema S, and results are transformed back into the global schema G using mapping M. In the next subsections, we will discuss different aproaches proposed for OBDA for diverse data sources, summarized in Table 1.

3.1. OBDA for Relational Databases

A detailed review for OBDA to relational databases has been presented in [13]–[15]. Yet, a recent study in [16] introduces Ontop4theWeb, an OBDA-based system for Web data. They translated submitted SPARQL queries into SQL queries on-the-fly, using R2RML mappings. These mappings defined how Web data are mapped to virtual RDF terms and identifies the corresponding Web data sources. They illustrated the applicability of Ontop4theWeb demonstrating how far it is scalable and feasible through three realistic applications (Twitter, Foursquare, and Yelp), considering three use cases: (i) A significant amount of heterogeneous crowd-sourced information is involved to address the data variety dimension, (ii) A snapshot at a given time of the respective information get updated so frequently and might become outdated soon, to address the data velocity dimension, and (iii) This information is widely used by application developers.

Table 1. Summary of OBDA Approaches for relational and non-relatioal data sources

Ref.	Year	Source database	Source Query language	Mapping Model
[13] [14] [15] [16]	2014 2018 2020 2020	Relational	SQL	R2RML
[17] [19] [20]	2016 2016 2019	Document	MongoDB aggregation query language (MUPGL)	Relational view over MongoDB
[21]	2016	Document	jsonPath	xR2RML
[22]	2017	Document	jsonPath	OOP
[23]	2020	CSV	SQL	Virtual Tabular Dataset (VTD) model
[24] [25] [26]	2015 2009 2009	XML	XQuery	XML Mappings
[27]	2020	Property Graph	Cypher	xR2RML
[29]	2019	Data lake	SQL	wrappers for several databases with SQL access

3.2. OBDA for Document Databases

Authors in [17] proposed the use of NoSQL in OBDA applications by extending the well-known Relational-to-RDF framework, Ontop [18]. A document-oriented MongoDB database is utilized to evaluate the modified architecture. In an extended work [19] [20], they concluded that building a completely generic framework that is capable of querying any kind of NoSQL databases is very hard. Every NoSQL database requires a specific translator because very few query patterns are shared among various kinds of such databases, unlike relational databases that use a common query language (SQL).

In another interesting work [21], authors convert a legacy database into a publicly available data source by building a SPARQL to MongoDB query mapping tool. The database should be made available as a virtual RDF database. The SPARQL query is translated into an abstract query utilizing mappings from MongoDB documents to RDF expressed in an intermediary language called xR2RML. The query is then rebuilt to be a concrete MongoDB query.

In [22] the use of an OBDA model is proposed with an Access Interface consisting of an ontology, a mapping, and an intermediary conceptual layer capable of enabling data access to a NoSQL database management system. Our The goal of this technique is to make mapping construction more flexible and general, allowing it to be reused regardless of the DBMS used for the persistence layer. To accomplish so, authors created an intermediate conceptual layer that represents the structure (schema) of the data stored in the database using classes written in an object-oriented programming (OOP) language. A project named onto-mongo-targeted document-oriented NoSQL DBMS is the first implementation of this method.

3.3. OBDA for Tabular Datasets

In [23], they defined the notion of a Virtual Tabular Dataset (VTD), which is made up of OBDA annotations and alignments for a tabular dataset. They proposed a constraint-based OBDA for a tabular dataset called Morph-CSV. The input is a SPARQL query and a VTD, while the output is an OBDA instance. Morph-CSV operates according to the following steps: (i) constraints generation based on the VTD information; (ii) selection of attributes and sources to answer the query; (iii) pre-processing the selected sources by employing the proper constraints; and (iv) RDB instance generation together with the associated schema corresponding to the predefined constraints, so that it can be attached to any SPARQL-to-SQL engine. Several domains are involved in evaluating Morph-CSV: e-commerce, transportation, and biology with BSBM, Madrid subway and Bio2RDF use cases, respectively. Observed results proved that Morph-CSV is capable of reducing total query execution time while producing all query answers, by up to two orders of magnitude.

3.4. OBDA for XML Datasets

A couple of publications have described SPARQL2XQuery [24] [25] [26]. The SPARQL to XML translation is based on a mapping model defined between an existing or user-defined OWL ontology and an XML Schema. Extraction of mappings can be either automatically or manually by a domain expert by examining the ontology and XML schema. Without knowing the XML format, SPARQL queries are presented against the ontology. The SPARQL query's BGP (Basic Graph Pattern) is normalized to make UNION-free graph pattern, so that each pattern could be handled more efficiently independent of other patterns. Graph pattern variables are connected to XPaths in different ways, depending on the type of variable. The mappings are then used to convert graph patterns into equivalent XQuery expressions.

3.5. OBDA for Graph Databases

In our previous work [27], an Ontology Based Data Access (OBDA) approach was proposed to translate SPARQL queries over an existing into appropriate queries over property graph databases using mappings between them. At first, the proposed work transforms the input mappings written in xR2RML into Hybrid relational Graph Algebra (HGA) expressions, to simplify the translation process. This process is done only once in offline mode. Next, two main modules conduct the SPARQL to Cypher translation procedure in an online phase. The first online module binds the submitted SPARQL query to the appropriate HGA expressions. These expressions are then translated, in the second module, into Cypher query language for property graph databases over which the translated query is evaluated, and results are transformed back into RDF.

3.6. OBDA for Data lakes

The term "Data Lake" [28] refers to a collection of heterogeneous data in its original format and shape, including NoSQL databases, and distributed file system (e.g., HDFS). Authors in [29] proposed an OBDA architecture on top of the data lake, referred to as: SEMANTIC DATA LAKE. The idea is to avoid the need for pre-processing or data materialization of multiple forms of raw data stored in a Data Lake. They provided an easier access to these heterogeneous data by using a single query language (SPARQL [30]). They proposed a method for executing distributed queries, with an emphasis on the multisource join operation. Two query engines (Apache Spark and Presto) were used to create an example of the proposed architecture, which provides wrappers for numerous databases accessed with SQL. Consequently, a custom SPARQL-to-SQL converter is developed. a specific ontology is created to categorize related concepts of NoSQL databases, which is then utilized in data mappings. In their current implementation, they employed a tabular representation during the query processing process, thus when a SPARQL query is issued, only SQL is used.

4. Applications to OBDA

Ontology-based Data Access for data integration is beneficial in many fields. In [31], authors developed an ontology that is focused on a water supply network to facilitate user’s interaction with data by modelling, producing, integrating, publishing, and exploiting it. This ontology was created in OWL 2 and takes into account various concepts, properties, and relationships in order to conceptualize the field of water management supply networks. A semantic model was built to materialize the data sources’ concepts, processes, and necessary components. The concepts are merged into the RDF repository according to the ontology scheme. A set of SPARQL queries enabling federated querying has been developed on top of that.

Authors in [32] have created a Semantic Extract Transform Load framework. The study includes a brief analysis of the previous integration system's problems, as well as a framework for two case studies: home travel and fuel economy data, and data from Integrated MOOCs Providers. The task was divided into three stages: extracting data from multiple data sources, transforming it into clean preprocessed data, and loading it into the desired operational database system. The work employs a variety of semantic terminologies, including RDF, SPARQL, OWL, an XML editor, and ontologies, to implement this strategy. The work is well-structured and acceptable, with a detailed description of the proposed framework. However, because it relies significantly on manual data processing, the method is time demanding.

According to the proposed work in [33], industrial Ontologies were used to support links across various data sources. Inference methods over these ontologies are used to help the identification of relationships in this application. Authors believed that the inference activity can be understood as an iterative process that supports the creation and subsequent implementation of heuristics. As a result, a data strategy to support continuing data integration can be linked to this. They conducted a case study to investigate supply chain risk detection with different source risks of an industry. The research employs a variety of ontologies to implement this concept, including Geo Positioning, Weather, Vehicles, and Sales. The study discusses the ontology federation procedure for risk detection and analysis, although the entire system is based on a time-consuming manual approach.

The strategy in [34] highlights an ontological framework for integrating diverse clinical data sources (relational, XML, and ASCII text). A generic, semantically rich, and standardized ontology is created to link numerous medical terminologies using Shapes Constraint Language (SHCL) and Top Quadrant SPARQL Inference Notation (Top SPIN). It also assured that it is linked to medical standards by leveraging Top Braid Composer maestro edition's various capabilities and plugins.

A Common Greenhouse Ontology (CGO) was created in [35]. This CGO adds horticulture concepts to the domain-independent SOSA ontology (Sensor, Observation, Sample, and Actuator). Data Analysis Facility (DAF) tool has been built which consists of various components and is used to understand data structured using the SOSA ontology. The sensors offer data in a non-RDF format, which mappers convert into RDF. In the interoperability layer, the data is saved in several triple stores using a common language that employs SOSA, which in our instance is the CGO. Each data collection has its own triple store in the interoperability layer. A SPARQL endpoint is exposed for each triple storage.

5. Conclusion

In this paper, the main principles of Ontology-Based Data Access (OBDA) are discussed, followed by a review of various OBDA approaches for diverse data sources i.e., relational, XML, Graph, etc. The description of mappings between the global schema and the local data source is the keystone in all these approaches. A flexible mapping description mechanism is required to enable the RDF mapping of data sources with different data models and query capabilities. Further investigations are required to enable the OBDA framework to cope with the evolution of the global schema (i.e., changes, removal, or insertion). Additionally, there is a crucial need for OBDA benchmarks for non relational data sources with complex mappings and huge data instances. Finally, extending the capability of OBDA technology to deal with streaming data efficiently would be beneficial in many fields i.e. the Internet of Things.

References

[1] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati, "Linking Data to Ontologies," *J. Data Semant. X*, pp. 133–173, 2008, doi: 10.1007/978-3-540-77688-8_5.

[2] M. Lenzerini, "Ontology-based data management," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 5–6.

[3] B. McBride, "The resource description framework (RDF) and its vocabulary description language RDFS," in *Handbook on ontologies*, Springer, 2004, pp. 51–65.

[4] E. Kharlamov *et al.*, "Ontology based data access in Statoil," *J. Web Semant.*, vol. 44, pp. 3–36, 2017.

[5] O. Corcho, F. Priyatna, and D. Chaves-Fraga, "Towards a new generation of ontology based data access," *Semant. Web*, vol. 11, no. 1, pp. 153–160, 2020.

[6] M. Arenas, A. Bertails, E. Prud'hommeaux, J. Sequeda, and others, "A direct mapping of relational data to RDF," *W3C Recomm.*, vol. 27, pp. 1–11, 2012.

[7] R. Das, S., Sundara, S., Cyganiak, "R2RML: RDB to RDF mapping language. W3C recommendation," 2012. https://www.w3.org/TR/r2rml/.

[8] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. de Walle, "RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data.," 2014.

[9] F. Michel, L. Djimenou, C. Faron-Zucker, and J. Montagnat, "Translation of relational and non-relational databases into RDF with xR2RML," in *11th International Confenrence on Web Information Systems and Technologies (WEBIST'15)*, 2015, pp. 443–454.

[10] G. Fusco and L. Aversano, "An approach for semantic integration of heterogeneous data sources," *PeerJ Comput. Sci.*, vol. 6, p. e254, 2020.

[11] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2002, pp. 233–246.

[12] F. Michel, "Integrating heterogeneous data sources in the Web of data," Université Côte d'Azur, 2017.

[13] F. Michel, J. Montagnat, and C. Faron-Zucker, "A survey of RDB to RDF translation approaches and tools. Rapport de Recherche. ISRN I3S," 2014.

[14] G. Xiao *et al.*, "Ontology-based data access: A survey," 2018.

[15] T. Schneider and M. Šimkus, "Ontologies and data management: a brief survey," *KI-Künstliche Intelligenz*, vol. 34, no. 3, pp. 329–353, 2020.

[16] K. Bereta, G. Papadakis, and M. Koubarakis, "OBDA for the Web: Creating Virtual RDF Graphs On Top of Web Data Sources," *arXiv Prepr. arXiv2005.11264*, 2020.

[17] E. Botoeva, D. Calvanese, B. Cogrel, M. Rezk, and G. Xiao, "OBDA beyond relational DBs: A study for MongoDB," 2016.

[18] T. Bagosi *et al.*, "The ontop framework for ontology based data access," in *Chinese Semantic Web and Web Science Conference*, 2014, pp. 67–77.

[19] E. Botoeva, D. Calvanese, B. Cogrel, M. Rezk, and G. Xiao, "A formal presentation of MongoDB (Extended version)," *birth*, vol. 1926, pp. 8–27, 2016.

[20] E. Botoeva, D. Calvanese, B. Cogrel, J. Corman, and G. Xiao, "Ontology-based data access--Beyond relational sources," *Intelligenza Artif.*, vol. 13, no. 1, pp. 21–36, 2019.

[21] F. Michel, C. Faron-Zucker, and J. Montagnat, "A mapping-based method to query MongoDB documents with SPARQL," in *International Conference on Database and Expert Systems Applications*, 2016, pp. 52–67.

[22] T. H. D. Araujo, B. T. Agena, K. R. Braghetto, and R. Wassermann, "OntoMongo-Ontology-Based

Data Access for NoSQL.," *ONTOBRAS*, vol. 1908, pp. 55–66, 2017.

[23] D. Chaves-Fraga, E. Ruckhaus, F. Priyatna, M.-E. Vidal, and O. Corcho, "Enhancing virtual ontology based access over tabular data with Morph-CSV," *Semant. Web,* no. Preprint, pp. 1–34, 2020.

[24] N. Bikakis, C. Tsinaraki, I. Stavrakantonakis, N. Gioldasis, and S. Christodoulakis, "The SPARQL2XQuery interoperability framework," *World Wide Web,* vol. 18, no. 2, pp. 403–490, 2015.

[25] N. Bikakis, N. Gioldasis, C. Tsinaraki, and S. Christodoulakis, "Querying xml data with sparql," in *International conference on database and expert systems applications*, 2009, pp. 372–381.

[26] N. Bikakis, N. Gioldasis, C. Tsinaraki, and S. Christodoulakis, "Semantic based access over xml data," in *World Summit on Knowledge Society*, 2009, pp. 259–267.

[27] N. Fathy, W. Gad, N. Badr, and M. Hashem, "Querying Heterogeneous Property Graph Data Sources Based on a Unified Conceptual View," in *Proceedings of the 2020 9th International Conference on Software and Information Engineering (ICSIE)*, 2020, pp. 113–118, doi: 10.1145/3436829.3436855.

[28] D. E. O'Leary, "Embedding AI and crowdsourcing in the big data lake," *IEEE Intell. Syst.*, vol. 29, no. 5, pp. 70–73, 2014.

[29] M. N. Mami, D. Graux, S. Scerri, H. Jabeen, S. Auer, and J. Lehmann, "Squerall: Virtual ontology-based access to heterogeneous and large data sources," in *International Semantic Web Conference,* 2019, pp. 229–245.

[30] E. Harris, S., Seaborne, A., Prud'hommeaux, "SPARQL 1.1 Query Language," *W3C recommendation 21, no. 10,* 2013. http://www.w3.org/TR/sparql11-query/.

[31] P. Escobar, M. del M. Roldán-Garc\'\ia, J. Peral, G. Candela, and J. Garc\'\ia-Nieto, "An ontology-based framework for publishing and exploiting linked open data: A use case on water resources management," *Appl. Sci.*, vol. 10, no. 3, p. 779, 2020.

[32] S. K. Bansal and S. Kagemann, "Integrating big data: A semantic extract-transform-load framework," *Computer (Long. Beach. Calif).*, vol. 48, no. 3, pp. 42–50, 2015.

[33] D. Ostrowski, N. Rychtyckyj, P. MacNeille, and M. Kim, "Integration of big data using semantic web technologies," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC),* 2016, pp. 382–385.

[34] M. Yadav, V. Singh, and Prachi, "Ontology based data integration and mapping for adverse drug reaction," in *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC),* 2021, pp. 719–727, doi: 10.1109/ISPCC53510.2021.9609499.

[35] R. Bakker, R. van Drie, C. Bouter, S. L. van Leeuwen, L. A. van Rooijen, and J. L. Top, "The Common Greenhouse Ontology: an ontology describing components, properties, and measurements inside the greenhouse," 2021.