A Data Mesh Adaptable Oil and Gas Ontology Based on Open Subsurface Data Universe (OSDU)

Neda Abolhassani, Ana Tudor and Sanjoy Paul
Accenture Labs, 415 Mission St Floor 35 San Francisco, California, 94105, U.S.A.

Keywords: Ontology, Oil and Gas Data Platform, Integration, Data Mesh.

Abstract: Incompatible models, heterogeneous data, and siloed data present challenges for the Oil & Gas industry. Knowledge graphs provide efficient consolidation, improved quality, and universal access to data, addressing these challenges. Developed by major global Oil & Gas and cloud organizations, the Open Subsurface Data Universe (OSDU) platform provides subsurface energy data ingestion, enrichment, and consumption services, as well as metadata storage, indexing, and search services. OSDU data supply chain aligns with the main concepts of the new trending data architecture, Data Mesh, such as federated data governance, decoupling data from applications, and domain specific data products. Data integration in subsurface data industry can be achieved by building a domain knowledge graph based on standard and enriched OSDU framework schemas. A knowledge graph-based solution begins with building a domain ontology. The purpose of this article is to introduce the OSDU ontology, which is publicly available on GitHub under the Apache 2.0 license. This paper discusses OSDU ontology design, development, applications, and evaluation.

1 INTRODUCTION

The Oil & Gas industry is a complex and dynamic domain that involves various activities such as exploration, production, refining, transportation, and distribution of petroleum products. The industry generates massive amounts of data from diverse sources such as sensors, equipment, and production systems. Many companies are held back by an ever-increasing business complexity, uncertainty in managing data silos, and unavailability of their data through diverse modes of access. The challenge of data integration can be resolved by building a knowledge graph and modeling the data and its relationships in a heterogeneous network structure through the logical form of ontologies. Data integration in Oil & Gas industry is one of the significant activities done at Norwegian Continental Shelf. The project group IDP-D&C focuses on complex drilling and completion processes, integrating offshore and onshore data, and real-time simulations for optimizing processes (Thorsen and Rong, 2008). Semantic Web and common Oil & Gas ontologies are utilized in the solution for exchanging domain expert data from operation centers and real-time sensor data from oil fields. (Guan et al., 2019) and (Huang et al., 2020) build intelligent knowledge graph-based search engine applications for Oil & Gas information. Their applications have key functionalities, such as knowledge fusion, topic classification, smart search and recommendation while displaying the results on a map. They use Neo4J as their knowledge graph storage which is a labeled property graph and does not utilize a standard ontology model so it cannot be enabled with semantic reasoning.

As data management systems evolve, knowledge graphs and ontologies should adapt as well. Past evolution of data management platforms has focused on bringing data into a central repository and introduces new problems, such as data ingestion, data extraction, data cleaning, dataset discovery, metadata management, data integration, and dataset versioning (Nargesian et al., 2019). With CoreKG (Beheshti et al., 2018), users can curate, index, and query raw data and metadata and have access to a centralized repository of both raw and contextualized data. Contextualized and curated data is stored in a knowledge graph. However, CoreKG does not localize changes to domains within an enterprise and does not provide interoperability between various domains and their data. Data Mesh is the newest inflection point in data management platforms and it unlocks business agility (Dehghani, 2020). Data Mesh addresses shortcomings of current and previous generations of data lakes and warehouses by applying domain-driven design think-

Abolhassani, N., Tudor, A. and Paul, S.
A Data Mesh Adaptable Oil and Gas Ontology Based on Open Subsurface Data Universe (OSDU).
DOI: 10.5220/0012160000003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 2: KEOD*, pages 29-39
ISBN: 978-989-758-671-2; ISSN: 2184-3228

ing and agile operations(Evans, 2004). Domains in Data Mesh are responsible for serving their own data with a product mindset in a distributed data management system. It is necessary that domain-generated data products are highly available, discoverable, and secure, as well as interoperable with the analytical applications that require access to them. The other key principles of Data Mesh are its self-serve infrastructure and federated governance for managing the end-to-end life cycle of domain data products (Dehghani and Fowler, 2022). Decentralized domain-oriented data mesh architecture can help Oil & Gas companies solve data discovery, consumption, trust, and governance bottlenecks.

An Oil & Gas data platform that supports various contexts (e.g. production or geology) related to a single concept (e.g. well) via Domain Driven Design is Open Subsurface Data Universe (OSDU™, 2023). It is a cross industry collaboration led by Open Group to develop a standard data platform for Oil & Gas exploration and production cycle. Leading Oil & Gas industry operators, cloud providers, and Open Group came forward in 2018 to build a standard data platform for accelerating deployment of emerging digital solutions which helps with more enhanced data discovery and decision making for subsurface energy data. It provides standard schemas and data types for upstream data with the intention of extending to other, newer energy data types. Although the design and development of OSDU data platform has been started before the existence of Data Mesh, it follows most of its principals (Landre, 2021). Federated data governance, decoupling data from application, and data products with domain specificity are some of the commonalities between OSDU data platform and Data Mesh.

In this work, we introduce OSDU ontology which encapsulates subsurface energy business domain, technology terminology, and common data access standard based on the schema files defined by the OSDU Open Group. Ontology defines a relationship between objects using W3C's standardized format enabling deeper semantic queries that domain specialists are interested in. Moreover, an ontology can be used by knowledge graph technologies that are gaining momentum. Oil & Gas is such a specialized domain, just as many other domains that it requires its own ontology (like FHIR in Healthcare, MaRCO in Manufacturing, FIBO in Finance, and Semantic Sensor Network for IoT). This ontology is open sourced and licensed under the permissive Apache 2.0; it follows the standards defined by the major global Oil & Gas organizations; and it covers various subsurface domains to be performed as a knowledge graph-based catalogue to support Data Mesh architecture.

This paper is organized as follows: Section 2 discusses prior work in geology and energy ontology and positions our contributions against the literature; Section 3 describes OSDU ontology, its design rationale, implementation pipeline, and significant entities; Section 4 shows the evaluation results; Section 5 demonstrates some use cases for our work. We conclude and discuss future work in Section 6.

2 RELATED WORK

The integration of geology and petroleum data has been the subject of many studies. Data from disparate domain data sources and vocabularies is blended using semantic web and ontologies.

GeoCore (Garcia et al., 2020) is a geological ontology and includes definitions of limited but generic concepts within the wide domain of geology, such as geological time intervals, process, structure, earth material, and rocks. It is based on BFO top level ontology and facilitates the communication of the geologists through their domain applications. Although GeoCore can be used in the petroleum exploration and production, it lacks detailed operational and business logic of this specific domain.

Another generic ontology which is more specific to the energy domain is Open Energy Ontology (OEO) (Booshehri et al., 2021). In addition to integrating several relevant domain terminologies, it is developed for the general domain of energy systems. The concepts and vocabularies are integrated from multiple domains: location of energy generation, consumption, and transmission from geography domain, fluctuating renewable energy generation and extreme weather conditions from meteorology domain, modeling methods from math, energy and emission market, prices, and costs from economics domain, technology, future development, and efficiency from engineering domain. Some standard ontologies like BFO, Relation Ontology, Unit Ontology, and Information Abstract Ontology Models are imported to make a more extensible canonical model. The OEO covers many aspects of energy modeling, but not enough in the subsurface energy sector.

There are some initiatives of using ontologies in the Oil & Gas industry, such as SmartWellOnto (Oprea et al., 2006), IIP (Gulla et al., 2006), AKSIO (Norheim and Fjellheim, 2006), and OGO (POSC, 2020). SmartWellOnto is one of the earliest ontologies based on Prolog language designed for a knowledge-based system that analyzes the monitored parameters of an oil reservoir, e.g., pressure, temperature, and injection flow. It provides a solution for optimized exploitation. IIP is an OWL ontology for Norwegian petroleum business which is used in production reports. Majority of the IIP development is based on converting ISO 15926 data model and reference data library into OWL. AKSIO is a process-enabled knowledge management system to provide experience of specialists on offshore oilfield operations using a contextual ontology driven approach. OGO is also based on ISO 15926 data model and is developed by POSC Caeser Association for consistent oil and gas data integration. It covers domain specific terminologies relevant to drilling and completion, development and modification, reservoir and production procurement and logistics, operation and maintenance, and transport.

Most of the existing ontologies are too generic and offer only partial solutions for subsurface energy industry domain. Additionally, most of these ontologies and data models are not publicly available, so their reusability is limited. Aside from that, they are hard to align with Data Mesh, the newest paradigm in data architecture. Federated data governance concepts such as access control and different data product personas are not reflected in these domain ontologies. This work was motivated by the above-mentioned insufficiencies, as well as compatibility with the well-received OSDU data platform. The OSDU terms are widely accepted by the Oil & Gas industry, including Shell, Schlumberger, Chevron, BP, Total, etc. Our ontology conforms to the standards set by these companies, making it applicable to them.

3 OPEN SUBSURFACE DATA UNIVERSE ONTOLOGY

The OSDU ontology includes physical objects, such as wellbore and basin, or business activities, such as seismic acquisitions. These digital objects contain static context, such as units of measurement. Metadata are also included, such as file owner and ancestry information, access controls, versions, and sources. Using the OSDU ontology, domain experts can perform semantic queries not possible otherwise, as well as represent a data management model. By itself, OSDU is a centralized system. On the other hand, converting the OSDU standards to OSDU ontology facilitates the deployment of distributed (federated) data by adding a knowledge graph layer on top of all the existing domain data.

We defined a new namespace *https://w3id.org/osdu* and we also registered the prefix *osdu* at *http://prefix.cc* for all the resources used in the ontology in order to easily distinguish it while integrating with external ontologies. In addition, all the code, documentation, and ontology Turtle files are available to public via Github[1] for getting further contributions from the community.

```
{ "data": {
    "allOf": [
      { "$ref": "../abstract/AbstractCommon
        Resources.1.0.0.json" },
      { "type": "object",
        "properties": {
          "WellID": {
            "type": "string",
            "pattern": "^[\\w\\-\\.]+:master
            -data\\-\\-Well:[\\w\\-\\.\\:
            \\%]+:[0-9]*$",
            "x-osdu-relationship": [
              { "GroupType": "master-data",
                "EntityType": "Well" }
            ]
          }
        }
      }
    ]
  }
}
```

Figure 1: Sample OSDU JSON schema file.

3.1 Design Rationale

Since OSDU was built with an interoperability goal and adopts a domain-driven design, the OSDU ontology is designed based on its schema and data definitions. The data loaded into the platform must adhere to a predefined JSON standard, which can be found in the OSDU schema files[2]. An example of these schema files can be found in Figure 1. In OSDU, different group types are explained by schema files, including Master Data, Reference Data, File, Work Product Component, and Work Product. The Master Data group type refers to the information about the business and physical objects, providing context and properties for the associated digital objects. As with Master Data, Reference Data refers to concepts outside of a company's business processes and is more static. Digital files are represented by File group type. They are immutable and contain metadata about the files. The Work Product Component group type represents business metadata and logical concepts of a set of files. In addition to being versioned, immutable, and GUID-enabled, it may also pertain to measurements, observations, and interpretations of business

[1] *https://github.com/Accenture/OSDU-Ontology*

objects. The Work Product group type is a set of Work Product Components wrapped together for data ingestion into OSDU data platform.

Using the OSDU schema JSON files, the following rules are followed in order to construct an OWL 2 ontology:

1. Create a class for each OSDU data platform group type and JSON schema. A class *osdu:AbstractCommonResources* could be defined for *AbstractCommonResources.1.0.0.json*, for example.
2. Most OSDU schema files contain similar properties, including id, kind, legal, meta, version, and tags. As a result, we create an *osdu:System* class and add all of these properties as datatype properties. Through a *rdfs:subClassOf* link, each class corresponding to a JSON schema file will be connected to *osdu:System*. ACL is the only exception, which is defined as *osdu:ACL*, a subclass of *osdu:AbstractAccessControlList*, and connected to *osdu:System* by *osdu:hasACL*. This design choice is so that we can apply the generic properties of *osdu:AbstractAccessControlList* to *osdu:ACL* and define *osdu:owners* and *osdu:viewers* datatype properties for each instance.
3. If the OSDU schema file specifies required attributes, the entities listed as required data must have *owl:minCardinality* of 1. For instance, since *osdu:kind* datatype property and *osdu:ACL* class are required entities, they must have a minimum cardinality of 1.
4. There is also a *rdfs:comment* associated with each class, object property, and datatype property corresponding to the schema entity's description field.
5. There is a *data* key in every OSDU schema JSON file that describes the properties of its classes. Our ontology includes the nested properties of *data*. Assuming the *allOf* nested property of *data* contains a *$ref* entity, we select the referenced schema file as the superclass of the class associated with this OSDU schema JSON file. Attributes of *properties* that are themselves nested properties of *data* are subject to the following rules.
6. If an attribute has a property of *type* with the value of string or integer, a datatype property should be created in the ontology and named after the attribute. Created datatype property has the class corresponding with the schema JSON file as its domain and the value of *type* as its range.
7. Created datatype properties can also include *xsd:pattern*, which contains the value associated with the attribute's *pattern* entity.
8. Datatype property names that are already occupied by classes in our ontology must be extended with *"_property"*.
9. If an attribute has a property of *type* with the value of object, an object property should be created in the ontology. The object property's name begins with *"has"* and continues with the attribute's name. A new class must be created with the property's name in capitalized camel-case. All attributes nested in this class must be added, with this new class as their domain, in recursive fashion according to rules 6-14.
10. An object property must be created for attributes with *$ref* nested entities. The object property's name begins with *"has"* and continues with the attribute's name. The domain of the object property is the class associated with the schema JSON file, or the currently nested class under consideration, while the range is the class associated with the referenced schema file.
11. Whenever an attribute has a *type* of array but does not have either a *$ref* nested entity under its items, or a nested entity in its *items* with *type* "object" or "array", we only need to add a datatype property with the same name as the attribute, a range for the *type* subelement of *items*, and a domain name as the schema JSON file's name.
12. If there is an *x-osdu-relationship* entity associated with the attribute, which is now a datatype property in the ontology, the nested entities *GroupType* and *EntityType* need to be checked. Ontology classes must exist with the name of the *EntityType*, e.g. *WellboreTrajectoryType*, which are subclasses of classes with the name of the value of *GroupType*, e.g. *Reference Data*. The datatype property that has a subclass name similar to its value needs an *owl:allValuesFrom*.
13. If an attribute has a *type* of array and a property of *minItems*, an *owl:PropertyRestriction* must be placed on the property. If the attribute is being constructed as a Datatype property, the restriction is of type *owl:minCardinality*. If the attribute is being constructed as an Object property, the restriction is of type *owl:minCardinality*, with the cardinality restriction specified only on the Class

[2] https://community.opengroup.org/osdu/platform/data-flow/data-loading/open-test-data/-/tree/master/rc–3.0.0/3-schema

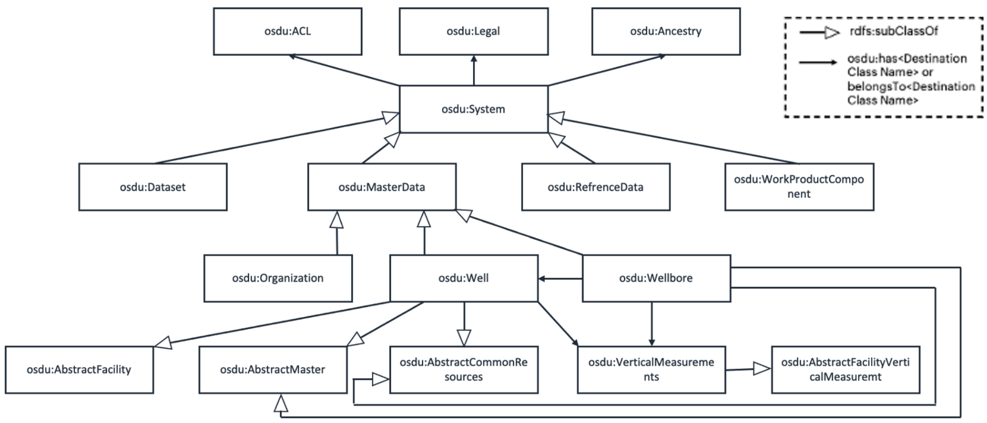


Figure 2: Partial schema diagram of the OSDU ontology.

constructed as the range for the property in this case.

14. New classes should be defined for attributes of *type* of array, which have nested entities of type *type* array. Such classes receive the name of the upper-level class associated with the attribute, and the name of the attribute appended, and then the word "Array" following. These classes receive cardinality restrictions on their number of items as specified in the previous rule.

As part of our efforts to align and extend the OSDU ontology to other domains, we have imported some open ontologies. FOAF[3], ACL[4], OWL-Time[5], and GeoNames[6] are among the imported ontologies. Using *owl:sameAs* or *rdfs:subClassOf* links, they are connected to the corresponding classes in our domain ontology. In addition, they are linked to their corresponding properties via the *rdfs:range*.

3.2 Implementation

There are some studies on building OWL ontologies based on JSON structured data (Delva et al., 2021)(Sbai et al., 2019). Inspired by these works, a Python implementation of the rules outlined in section 3.1 was undertaken. The data flow is shown in Figure 3, and the codebase is available on Github[7]. The algorithm proceeds as follows:

- The OSDU JSON schema structures are compiled into a nested dictionary by an intake script.
- A JSON parser receives this schema dictionary, and extracts input to form all instances of classes and properties that may be present in the data, based on the rules outlined in section 3.1 .
- An ontology modeler receives these classes and properties, and uses the expected hierarchical and property connections to form a linked graphical model of the ontology.
- The ontology modeler connects ontology properties and classes to public-available ontologies using a researcher-defined configuration file.
- The .ttl generator outputs a .ttl file in the expected format, complete with all specified classes, properties, and description logic.

In the ontology modeler's post-processing, the OSDU ontology is connected to external open ontologies. To allow researchers to specify which ontology properties and classes should be connected to open ontologies, a configuration file was designed. Once the ontology is reviewed, the high-level ontology generation can be re-run to establish new connections.

With the scalable automated implementation of this ontology generation, ongoing changes to the OSDU schema can be incorporated efficiently as they occur. The OSDU ontology will be updated as OSDU schema standards are updated in the future, so users of the OSDU data standard can link their databases to this proposed ontology. Additionally, since the algorithm correctly models the ontology's graphical, hierarchical nature on the backend, the ontology's structural integrity can be verified rapidly and procedurally. As will be covered in the section 4, the OSDU ontology scores highly on metrics of ontological quality, and this was ensured through its procedural formation. Finally, modularity and easy configuration allows for the fast generation of a very large ontology, with high amounts of nuance captured as specif-

[3] http://xmlns.com/foaf/0.1/

[4] https://www.w3.org/ns/auth/acl

[5] https://www.w3.org/TR/owl-time/

[6] https://www.geonames.org/ontology/documentation.html

[7] https://github.com/Accenture/OSDU-Ontology

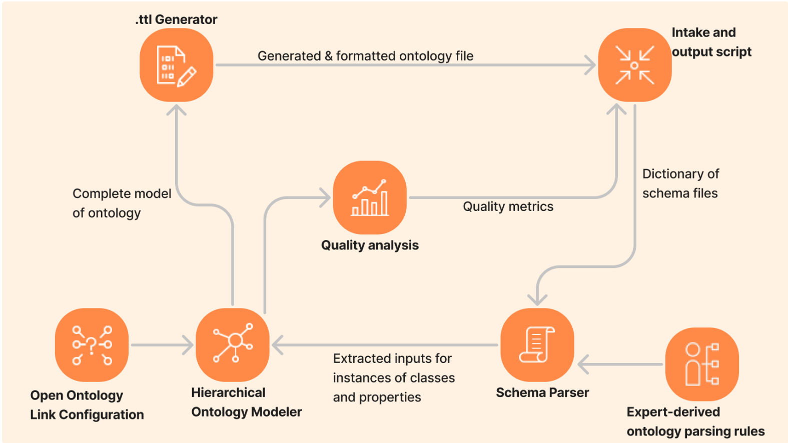


Figure 3: Data flow diagram of the OSDU ontology generation algorithm.

ically defined by the OSDU schema designers. It is now possible for ontology experts to review, correct, and augment the ontology with more time. Thus, OSDU's algorithmic ontology generation is one of its key highlights. The code is provided as a reference for ontology generation researchers to use as an algorithmic template for creating large ontologies in other domains.

3.3 Main Classes and Properties

A partial schema diagram is shown in Figure 2 showing some of the main classes and their relationships, while Figure 4 shows the top-level classes, object properties, and partial list of datatype properties.

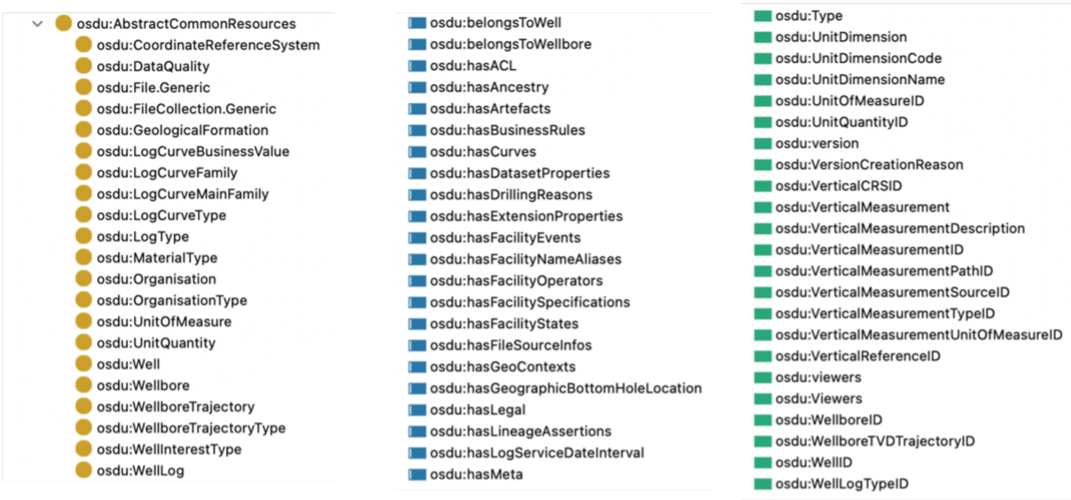
Classes whose names begin with *osdu:Abstract* define the parent properties of each associated class, such as alias names, owners, viewers, facility information, and spatial locations of wells or organizations. The *osdu:System* class contains metadata about its subclasses, such as the creation and modification times, users, access control, legal, and tag information. High-level information about subsurface energy concepts is included in the *osdu:MasterData* class, such as drilling reasons, trajectory types, hole locations, and vertical measurements. In the *osdu:ReferenceData* class, values such as units of measurement and units of quantity are stored for data validation. A class *osdu:WorkProductComponent* represents data resulting from a business activity, such as curve quality, top depth, base depth, log version, service company, drilling fluid property, log source, and business activity. Additionally, it contains links to the related files. As part of the *osdu:Dataset* class, the path to the actual file/s and metadata for the file/s are included.

4 EVALUATION

The final OSDU ontology is very large - 633 custom classes, 3939 property relationships, and 1498 inheritance relationships are proposed to semantically represent concepts in OSDU-specified data. A large scale and level of semantic specificity is recommended in order to capture the nuance of OSDU data specifications. Working with such a large ontology, however, requires careful evaluation of its usability (Hlomani and Stacey, 2014). Initially, subsections of the automatically generated ontology were compared with equivalents manually created by an expert ontologist, and the generation algorithm was modified until all these gold standard test cases were matched. The ontology is then evaluated according to two general approaches. Ontology quality was assessed using standard metrics, and the ontology was validated by verifying its suitability to answer competency questions relevant to OSDU data use cases.

4.1 Ontology Quality Metrics

To assess ontology structural quality(Yao et al., 2005), the number of leaf classes (NOL) in the ontology hierarchical structure and the average depth of the inheritance tree (ADIT-LN) were both measured. See Table 1 for all metric values. The OSDU ontology introduces a large number of highly specialized concepts due to its large number of leaf classes. Generally, a deep inheritance tree may be a sign that the ontology introduces too many intermediary classes. Small average depths can indicate that too many classes are independent and don't share any attributes. Having a measured depth of 3 for the



(a) Classes (b) Object Properties (c) Data Type Properties

Figure 4: Top Level OSDU ontology constructs.

OSDU ontology is a good middle ground that has a fair amount of shared attributes.

Table 1: Ontology Quality Metric Values for OSDU Ontology.

Metric	Metric Value
NOL	595
ADIT-LN	2.92
Relationship richness	0.72
Inheritance richness	2.37
Attribute richness	6.16

A few additional metrics were computed to quantify the richness of information conveyed by the ontology (Tartir et al., 2005) . Ontology richness was 0.72, indicating that a reasonable proportion of non-inheritance properties are represented. Ontology inheritance richness was 2.37, which is similar to average inheritance depth, suggesting a fair distribution of information across levels in the inheritance tree. Finally, attribute richness was measured as 6.16, which means that around 6 non-inheritance properties occur per class. Based on this data, it can be concluded that OSDU is well-suited to describe rich concepts since it contains a high amount of information about each node.

4.2 Competency Questions

In order to verify that the OSDU ontology can be used to model and query real-world data semantically, competency questions were devised. To this end, sample data provided by OSDU users in TNO[8] was mapped in Knowledge Graph format to the OSDU ontology, using the Stardog application. Numerous competency questions were devised and were answerable using SPARQL queries on this sample database. Two questions are selected as example cases, available in Figure 5 and Figure 6.

It is clear from the answers to these questions that the OSDU ontology is capable of capturing the connections present in real-world data and is capable of answering questions using simple queries over modeled data.

5 OSDU ONTOLOGY USAGE

Now that we have seen how OSDU ontology can be generated automatically from OSDU schemas, it would be useful to understand its broader implication in the context of data architecture. Specifically, we would like to explain how it can expedite the creation of Data Mesh architecture for the Oil & Gas industry.

Data Mesh is the latest trend in data architecture and every industry is moving away from the concept of Data Lake and adopting Data Mesh. There are sev-

[8]https://community.opengroup.org/osdu/platform/data-flow/data-loading/open-test-data/-/tree/master/rc–3.0.0/1-data/3-provided/TNO

```
SELECT ?geoentityID2 ?geoentityID1 ?stateTypeID
WHERE {
    ?well rdf:type osdu:Well ;
        osdu:facilityStates ?facilityState ;
        osdu:id ?wellID ;
        osdu:geoContexts ?geocontext1 ;
        osdu:geoContexts ?geocontext2 ;
    .

    ?facilityState rdf:type osdu:AbstractFacilityState ;
        osdu:facilityStateTypeID ?stateTypeID ;
    .

    ?geocontext1 a osdu:AbstractGeoContext ;
        osdu:geoPoliticalEntityID ?geoentityID1 ;
        osdu:geoTypeID ?geotype1 ;
    .

    ?geocontext2 a osdu:AbstractGeoContext ;
        osdu:geoPoliticalEntityID ?geoentityID2 ;
        osdu:geoTypeID ?geotype2 ;
    .

    FILTER(?stateTypeID = "Abandoned" && ?geoentityID1 =
"Netherlands_Country" && ?geotype2 = "Province")
}
GROUP BY ?geoentityID2 ?geoentityID1 ?stateTypeID
```

(a) SPARQL Query.

geoentityID2	geoentityID1	stateTypeID
"Groningen_Provinc...	"Netherlands_Country"	"Abandoned"
"Friesland_Province"	"Netherlands_Country"	"Abandoned"
"Gelderland_Provin...	"Netherlands_Country"	"Abandoned"
"Zeeland_Province"	"Netherlands_Country"	"Abandoned"
"Flevoland_Province"	"Netherlands_Country"	"Abandoned"
"Zuid-Holland_Provi...	"Netherlands_Country"	"Abandoned"
"Drenthe_Province"	"Netherlands_Country"	"Abandoned"
"Limburg_Province"	"Netherlands_Country"	"Abandoned"
"Noord-Brabant_Pr...	"Netherlands_Country"	"Abandoned"
"Utrecht_Province"	"Netherlands_Country"	"Abandoned"

(b) Result.

Figure 5: SPARQL query and answer for the competency question: *What is the list of provinces in Netherland where their well facility state types are "Abandoned"?*

```
SELECT ?drillReasonName ?primaryMaterial ?wellbore_iri ?well_iri
WHERE {
    ?wellbore_iri rdf:type osdu:Wellbore ;
        osdu:hasWell ?well_iri ;
        osdu:drillingReasons ?drillingReasons ;
        osdu:primaryMaterialID ?primaryMaterial ;
    .

    ?well_iri a osdu:Well ;
        osdu:id "8111" ;
    .

    ?drillingReasons a osdu:AbstractWellboreDrillingReasons ;
        osdu:drillingReasonTypeID ?drillreason_id ;
        osdu:hasDrillingReasonType ?drillreasontype_iri ;
    .
    ?drillreasontype_iri a osdu:DrillingReasonTypeID ;
        osdu:name ?drillReasonName ;
    .
}
```

(a) SPARQL Query.

drillReasonName	primaryMaterial	wellbore_iri	well_iri
"Development Hydrocarbons"	"Flr"	osdu:Wellbore-8111	osdu:Well-8111
"Development Hydrocarbons"	"Gas"	osdu:Wellbore-8976	osdu:Well-8111
"Development Hydrocarbons"	"Flr"	osdu:Wellbore-8956	osdu:Well-8111
"Development Hydrocarbons"	"Gas"	osdu:Wellbore-8599	osdu:Well-8111
"Development Hydrocarbons"	"Flr"	osdu:Wellbore-8244	osdu:Well-8111
"Development Hydrocarbons"	"Flr"	osdu:Wellbore-8462	osdu:Well-8111
"Development Hydrocarbons"	"Flr"	osdu:Wellbore-8463	osdu:Well-8111
"Development Hydrocarbons"	"Gas"	osdu:Wellbore-8464	osdu:Well-8111

(b) Result.

Figure 6: SPARQL query and answer for the competency question: *What are the primary materials in the trajectory of Wellbores in Well 8111? What are their drilling reasons?*

eral key features that characterize the Data Mesh Architecture, such as, decentralization of domains, data as a product, self-serve infrastructure, and federated governance. Note that data products are created and published by the producers, searched for and used by the consumers either directly or used in combination with other products to create new data products. The core component of the Data Mesh Architecture that makes the connection between the producers and consumers seamless is what is called the Data Catalog. Data Catalog, as the name suggests, holds the metadata about all the data products and the interconnection between them. As new products are created, their metadata gets added to the Data Catalog and new connections/links are built as necessary. A knowledge graph powered by the OSDU ontology acts as the Data Catalog for the Data Mesh architecture in the Oil & Gas industry. Data Catalogue is needed to provide federated governance and interoperability between decentralized domains, such as wellbore, basin, well construction, and well delivery. What lends credibility to this architecture for the Oil & Gas industry is that the OSDU concepts developed by the Oil & Gas industry consortia are aligned with the architectural tenets of Data Mesh architecture as shown in Table 2. Specifically, the knowledge graph-based Data Catalog powered by OSDU ontology helps to do authentication and validation of metadata for the creation of data products. Moreover, it helps to find connections between cross-domain products that may be combined to create new products in a seamless manner not possible otherwise. Similarly, from the consumer's perspective, the Data Catalog powered by OSDU ontology, by virtue of preserving the semantic relationship between the data products, enables complex semantic searches not possible otherwise.

According to Figure 7, the proposed architecture includes three horizontal layers: Raw Data Ingestion, Data Product Creation, and Data Product Consumption. There are also three vertical layers, corresponding to Data Flow, Metadata, and Human Expert. Dashed line boxes represent jobs pertinent to each domain based on its own requirements. Solid line boxes are common to all domains in a Data Mesh architecture.

The Raw Data Ingestion layer connects to various operational data stores via data virtualization, i.e., data can reside in any data store anywhere, but metadata corresponding to the physical data enables access to it. OSDU ontology provides domain standards, rules, access control, and semantics for each user persona.

The data products are generated using AI-powered techniques such as deduplication, cleaning, tagging,

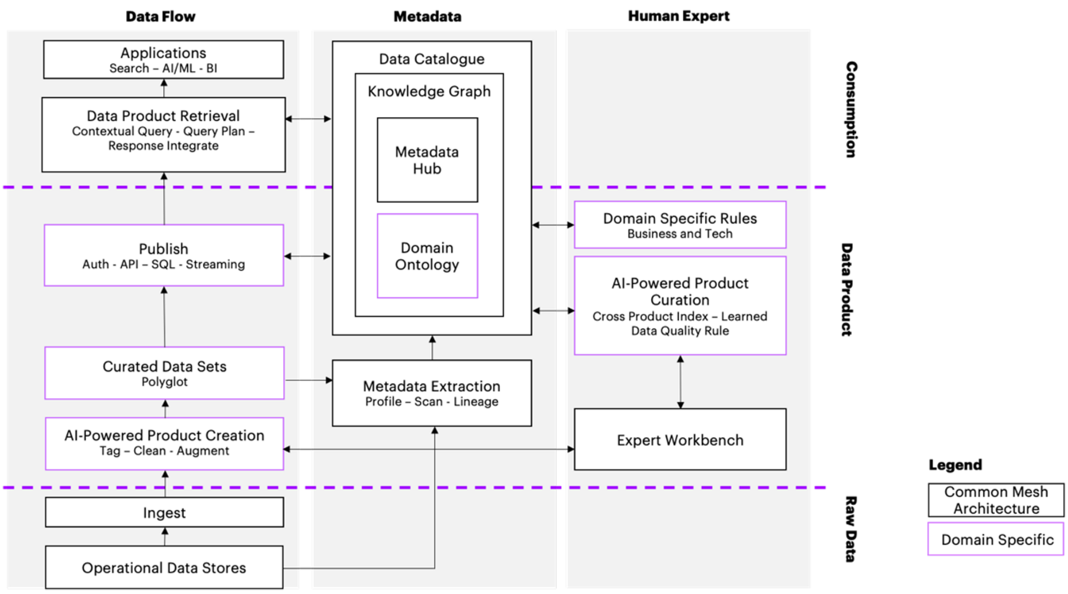


Figure 7: Knowledge graph supported Data Mesh architecture.

Table 2: OSDU concepts and Data Mesh concepts alignment.

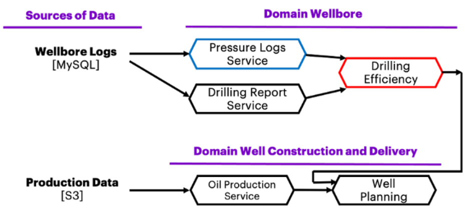
OSDU Concept	Data Mesh Concept
Domain	Domain
Work Product Component	Data product
Comment	Data product semantic
SecurityClassification in AbstractCommonRules	Security of data products and domains
LogVersion in Wellbore-Log, version of each class, ResourceLifeCyleStatus in AbstractCommonResources	Temporal aspect of data products
AbstractAccessControlList	Computational governance and code for enforcing access control
AbstractQualityMetrics	Quality of data products

imputation, and augmentation. Following this, the datasets are published as polyglot curated datasets, with different access points for different data consumption platforms. For the creation of AI-powered data products, an Expert Workbench is required to provide a user-friendly interface for Human (Domain) Experts to validate data quality rules. In the knowledge graph-based catalogue, metadata can be pulled from various generated data products. The knowledge graph also reflects all the learned data quality and cross product indices.

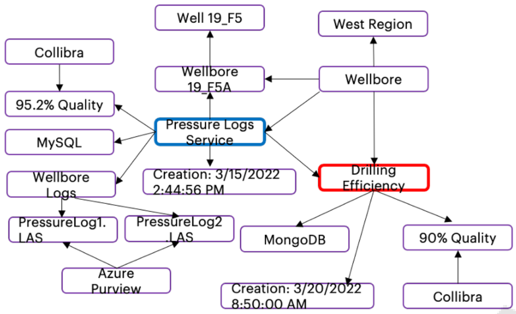
As shown in Figure 7, on the Consumption part of Data Flow, the end user can search and retrieve the data products of interest using contextual queries. Results are generated based on SPARQL queries submitted to the knowledge graph-based catalogue.

Let's illustrate the creation of a new data product using a simple example and show how a knowledge graph-based data catalog powered by the OSDU ontology enables it. Figure 8 shows an Oil & Gas enterprise following the above architecture, which is composed of several decentralized domains like Wellbore, Well Construction and Delivery, etc. Creating data products that are easy to discover and consume is the responsibility of each domain. As illustrated in Figure 8a, domain Wellbore generates a data product called Drilling Efficiency.

In order to calculate the value of a data product like Drilling Efficiency, it must have multimodal access to different data sources, semantics, and quality metrics. In addition, it must be computed and logged at constant intervals due to its temporal nature. For example, when a data scientist searches for Pressure Logs in order to predict Drilling Efficiency, the search engine authenticates the user and checks



(a) An example of domains and data products in an Oil & Gas enterprise.



(b) Knowledge graph-based data catalogue.

Figure 8: Domain decentralization and data products interoperability and a high-level reflection of their instances in the knowledge graph.

access control before presenting all the relevant Data Products from different domains. It will display various metadata related to Pressure Logs, including address, API endpoint, quality, semantics, lineage, and versions. In the backend, the knowledge graph is queried to retrieve the most relevant Pressure Log data products. When the data scientist is done with creating the Drilling Efficiency data product based on all the retrieved metadata of Pressure Log Products from different domains, the new data product, namely, Drilling Efficiency, gets indexed in the knowledge graph along with all its metadata as shown in Figure 8b. Prior to confirming its reconciliation in the knowledge graph, the domain expert needs to evaluate the generated metadata of the Drilling Efficiency data product.

New data products can also be created by accessing a data product from another domain. From the Well Construction and Delivery domain, for example, the Drilling Efficiency data product can be accessed and combined with the Oil Production Service data product from the Well Construction and Delivery domain to produce a new data product called Well Planning, as shown in Figure 8a. According to the mappings between Data Mesh concepts and OSDU data platform concepts, the OSDU ontology describes the location of data products and their domains. Taxonomies for quality metrics, versioning, domain semantics, and tools generating work product components are also included. Oil & Gas industry semantics and Data Mesh concepts are not covered by any of the previously defined domain ontologies. In addition, OSDU ontology is favored by energy companies who are interested in building a Data Mesh architecture for their enterprise data.

6 CONCLUSIONS

OSDU ontology is an abstraction layer aligning operational data with business concepts for subsurface energy data. As part of the Data Mesh trend in analytical data architecture, OSDU ontology classes and properties align with the new trend. Metadata and relationships are modeled between various Oil & Gas entities and attributes. Using OSDU data platform standards, we designed an ontology for modeling energy data. Currently, the platform covers information about exploration, development, production, and drilling of wells. As a result of OSDU ontology, discoverability was improved in OSDU platform by improving semantic search, user experience by enabling intuitive and efficient access to relevant data, data quality by removing redundancy and disambiguation during product creation in the AI-powered supply chain, and data quality by disambiguating data. The OSDU ontology has been introduced to the OSDU community in their biweekly forum meetings and has been well received by cloud providers and subsurface energy companies for building domain knowledge graphs based on standard Oil & Gas data models. In the future, OSDU plans to release data on solar, windfarms, hydrogen, and geothermal. Based on the rules that we defined for schema JSON file translation to OWL ontology, we can expand the scope of OSDU ontology according to new releases of the data platform to domains other than Oil & Gas. In light of the widespread adoption of OSDU standards, OSDU ontology is a natural choice as many Oil & Gas companies are also embracing the new paradigm of data architecture. The value of knowledge graphs and semantic technologies increases with the new Industry 4.0 trend, Digital Twin. The digital twins that are enabled with knowledge graphs are more integrated and provide better decision-making capabilities. Oil & Gas manufacturers can benefit from these digital twins by improving their operational efficiency, reliability, and agility. For energy industry-related digital twins, the OSDU ontology can be an invaluable resource.

REFERENCES

Beheshti, A., Benatallah, B., Nouri, R., and Tabebordbar, A. (2018). Corekg: a knowledge lake service. *Proceedings of the VLDB Endowment*, 11(12):1942–1945.

Booshehri, M., Emele, L., Flügel, S., Förster, H., Frey, J., Frey, U., Glauer, M., Hastings, J., Hofmann, C., Hoyer-Klick, C., et al. (2021). Introducing the open energy ontology: Enhancing data interpretation and interfacing in energy systems analysis. *Energy and AI*, 5:100074.

Dehghani, Z. (2020). Data mesh principles and logical architecture. *martinfowler. com*.

Dehghani, Z. and Fowler, M. (2022). *Data Mesh: Delivering Data-driven Value at Scale*. O'Reilly Media.

Delva, T., Van Assche, D., Heyvaert, P., De Meester, B., and Dimou, A. (2021). Integrating nested data into knowledge graphs with rml fields. In *KGWC2021, the Knowledge Graph Construction*, volume 2873.

Evans, E. (2004). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley.

Garcia, L. F., Abel, M., Perrin, M., and dos Santos Alvarenga, R. (2020). The geocore ontology: a core ontology for general use in geology. *Computers & Geosciences*, 135:104387.

Guan, Q., Zhang, F., and Zhang, E. (2019). Application prospect of knowledge graph technology in knowledge management of oil and gas exploration and development. In *2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages 161–166. IEEE.

Gulla, J. A., Tomassen, S. L., and Strasunskas, D. (2006). Semantic interoperability in the norwegian petroleum industry. In *Information Systems Technology and its Applications, 5th International Conference ISTA 2006*. Gesellschaft für Informatik eV.

Hlomani, H. and Stacey, D. (2014). Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, 1(5):1–11.

Huang, S., Wang, Y., and Yu, X. (2020). Design and implementation of oil and gas information on intelligent search engine based on knowledge graph. In *Journal of Physics: Conference Series*, volume 1621, page 012010. IOP Publishing.

Landre, E. (2021). Data mesh tale of two views of a thing. https://community.opengroup.org/osdu/platform/domain-data-mgmt-services/home/-/wikis/Data-Mesh---Tale-of-two-views-of-a-thing-(Einar-Landre). Accessed: 2023-04-30.

Nargesian, F., Zhu, E., Miller, R. J., Pu, K. Q., and Arocena, P. C. (2019). Data lake management: challenges and opportunities. *Proceedings of the VLDB Endowment*, 12(12):1986–1989.

Norheim, D. and Fjellheim, R. (2006). Aksio-active knowledge management in the petroleum industry. In *Proceedings of the ESWC 2006 Industry Forum*.

Oprea, M., Marcu, M., and Coloja, M. P. (2006). Smartwellonto: An ontology for smart wells. In *2006 International Multi-Conference on Computing in the Global Information Technology-(ICCGI'06)*, pages 36–36. IEEE.

OSDU™(2023). The open group: The open group guide osdu™system concept. https://osduforum.org/. Accessed: 2023-04-30.

POSC (2020). Posc caeser association. https://www.posccaesar.org/. Accessed: 2023-04-30.

Sbai, S., Louhdi, M. R. C., Behja, H., and Chakhmoune, R. (2019). Jsontoonto: building owl2 ontologies from json documents. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 10(10).

Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., and Aleman-Meza, B. (2005). OntoQA: Metric-based ontology quality analysis. In *Proceedings of IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*.

Thorsen, K. A. H. and Rong, C. (2008). Data integration in oil and gas at norwegian continental shelf. In *22nd International Conference on Advanced Information Networking and Applications-Workshops (aina workshops 2008)*, pages 1597–1602. IEEE.

Yao, F., Müller, H.-G., and Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. volume 100, pages 577–590. [American Statistical Association, Taylor and Francis, Ltd.].