Towards Data Integrity Verification for More Sustainable Petroleum Industry

Yuanwei Qu[1,*], Zhuoxun Zheng[2,1], Baifan Zhou[3,1], Yan Zhou[1], Nicolau Santos[4], Ognjen Savkovic[5], Arild Waaler[1] and David Cameron[1]

[1]*Department of Informatics, University of Oslo, Norway*

[2]*Bosch Center for AI, Germany*

[3]*Department of Computer Science, Oslo Metropolitan University, Norway*

[4]*Federal University of Rio Grande do Sul, Brazil*

[5]*Free University of Bozen-Bolzano, Italy*

Abstract

As a conventional energy industry, the petroleum industry is responsible for supplying over half of the world's energy. Facilitating sustainable development for petroleum energy production remains crucial. Data methods have emerged as powerful tools to advance sustainability by enabling efficient resource management and risk mitigation. However, the reliable implementation of data-driven methods relies on high-quality data, necessitating the verification of data integrity on substantial data volumes. To this end, this poster paper presents our ongoing research, leveraging ontologies and knowledge graphs as shared knowledge representation, and provides preliminary results on data integrity verification. Based on the ontologies, we formulate domain knowledge integrity constraints and test three technologies of integrity verification: Python, PySpark, and SPARQL, for exploring future potential industrial adoption.

Keywords

integrity verification, petroleum industry, sustainability

1. Introduction

Background. With the development of technology, society is becoming increasingly aware of the importance of sustainable development. As a conventional energy industry, the petroleum industry is still responsible for supplying over half of the world's energy [1]. Therefore, it is still extremely important to facilitate sustainable development for petroleum energy production. Being one of the most proactive industries embracing new technologies, the petroleum industry is widely adopting data-driven approaches to enhance energy production efficiency and safety, thereby reducing potential losses and environmental pollution during the production process. Currently, data-driven artificial intelligence (AI) has been widely applied to the petroleum industry to increase production efficiency and safety, from enhanced oil production and recovery to undesired event prediction [2]. This brings benefits such as reducing costly well tests, mitigating risks, increasing safety and operation efficiency. As results, data-driven AI contributes
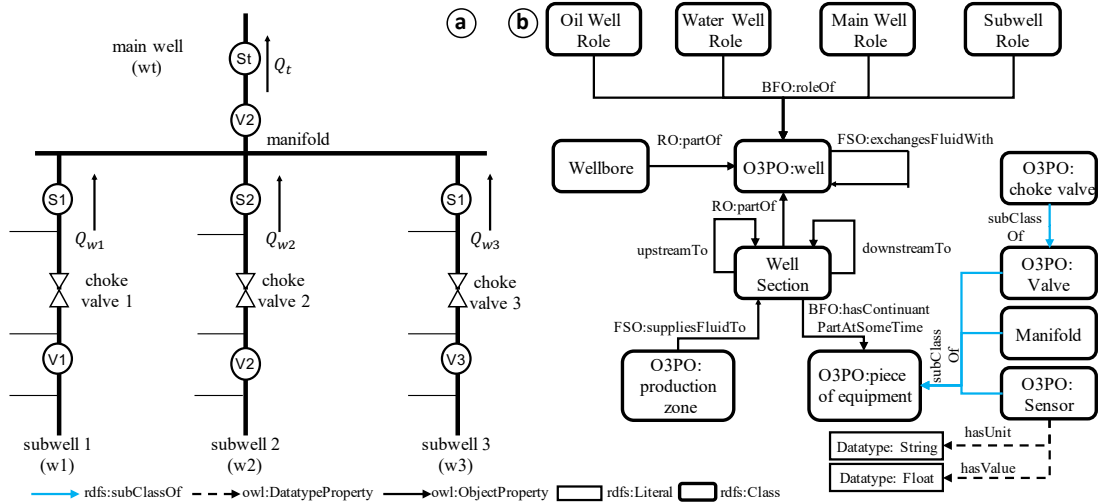
✉ quy@ifi.uio.no (Y. Qu)

Figure 1: (a) a simplified depiction of an oil production well example (S=sensor, V=valve, Q=fluid flow rate). (b) schematic illustration of the proposed ontology (partial) for petroleum production

to sustainability by improving energy production efficiency, and reducing accidents with severe environmental damage, such as oil leaks.

Challenge. The performance of data-driven approaches heavily relies on the data quality. Thereby, it presents an important challenge in ensuring the integrity of the data, which is crucial for data-driven solutions to deliver reliable predictions. In petroleum production, the purpose of collecting data from various sensors is to allow domain experts to analyse and make informed decisions based on their knowledge and experience. In this context, we discuss one of the issues of data integrity: the data should follow certain constraints of physical laws. This is in addition to other issues, such as missing values, sensor precision errors, etc. For verifying the constraints of physical laws, domain knowledge plays an important role, and should be incorporated in the integrity checking. Semantic technology is suited here due to its transparency, which the domain experts tend to have a high chance to trust because they can observe that their domain knowledge is used and how it is used.

Our approach. In this poster paper, we present our ongoing research on a semantic solution for data integrity verification for petroleum industry. We develop a draft of a petrolum ontology aligned with upper ontologies as shared representation for data integration and knowledge representation; we construct knowledge graphs for transparent and unified human understanding; we experiment technologies for data integrity verification, including Python, PySpark and SPARQL. We provide preliminary experiment results and discussion of adoption.

2. Data and Knowledge Representation

Ontology for petroleum production. In the petroleum domain, ontologies for petroleum exploration [3], reservoir [4], offshore production plant [5], subsurface fault [6], and petroleum risk assessment [7]. To meet our needs of verifying data integrity for the petroleum industry,

we develop a draft of an ontology for petroleum production wells. Fig. 1a presents a simplified visual depiction of oil production wells. The fluid flow depicted in the figure originates from subsurface sub-wells, then merges towards the main well, and in the end reaches the offshore production platform. Fig. 1b depicts our petroleum ontology written in OWL 2, which includes 15 classes, 11 object properties, and 2 datatype properties. It contains several classes that are essential to the industry, including *well section* and *subwell role*. The ontology further includes core relations such as *upstreamTo* and *downstreamTo*, which facilitate the representation of the spatial locations of each *well section* and indicate the flow direction of the fluid supplied from the production zone. To ensure compatibility and interoperability, our ontology has been aligned with a domain ontology: the Offshore Petroleum Production Plant Ontology (O3PO) [8], which is built upon the Basic Formal Ontology (BFO) [9], Relation Ontology (RO) [10], and Industrial Ontologies Foundry Core [11], and Flow System Ontology (FSO) [12]. By using these classes and relations, our ontology provides a structured framework for integrating data, renaming the variables, and capturing and organising relevant knowledge and data, and supporting the integrity verification.

Data from petroleum production wells. The data collected from the sensors in the production wells are typically presented in relational tables, including various sensor measurements on the range of well sections from the bottom-hole to the wellhead. These measurements include parameters such as pressure, temperature, and flow rate of each well section. Additionally, the collected data can contain other important equipment information, such as the ratio between the choke opening rate and flow coefficients. These relational tables provide a structured format for data-driven approaches to make predictions.

Knowledge graphs for petroleum wells. Based on the proposed ontology, we construct knowledge graphs (KG) (Fig. 2) with domain experts to illustrate the production wells, well sections, well sensors, and their relations. These KGs serve as a flexible foundation to formalise the domain knowledge , and to support a transparent shared understanding between the domain experts, semantic experts, data scientists, etc. The KGs can also have the potential for sophisticated reasoning, for example, using domain-knowledge-based constraints to detect subtle anomalies, identify potential erroneous data, and ensure the consistency of the delivered data.

3. Integrity Verification with Preliminary Evaluation

Integrity constraints. The integrity constraints play a crucial role in ensuring the quality of petroleum data. After renaming the features, we can formulate these constraints based on domain knowledge for verifying the data integrity. This allows validation of the sensor measurements, ensuring that they align with physical laws and empirical expectations. Here we give three examples (Fig. 2b):

Example 1: the flow rate at any position within a well must consistently equal the flow rate of its upstream or downstream locations.

Example 2: the total flow rate of the main well should precisely match the sum of the flow rates of all merged (or split) wells.

Example 3: for each well, both the flow rate and pressure consistently adhere to the principles outlined in the Bernoulli function.
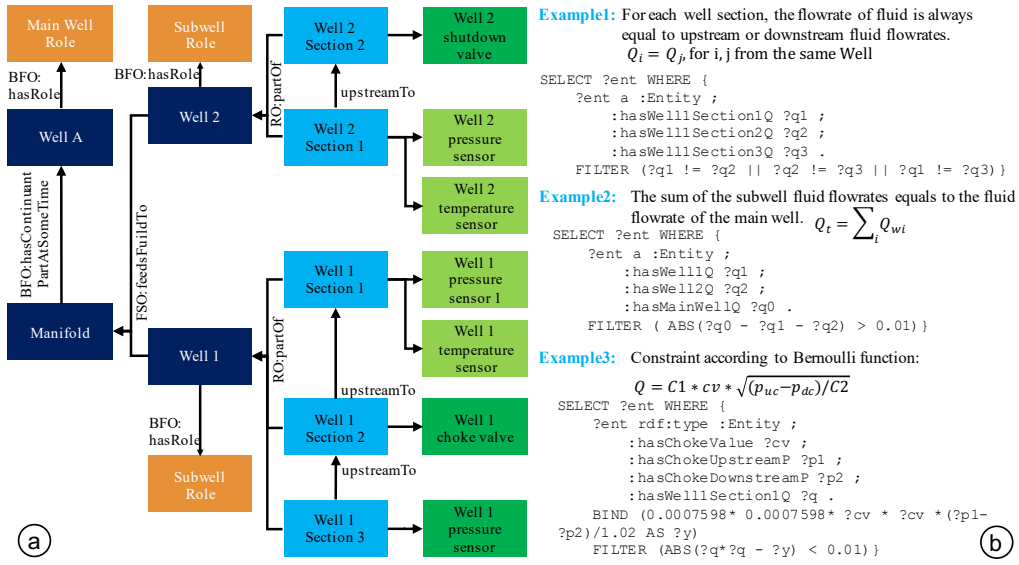
Figure 2: (a) Schematic illustration of the production KG, (b) example constraints formulated both in equations and SPARQL queries. The "strict equal" is relaxed to account for calculation errors. The missing values are handled by other queries, and we assume all values exist here. C1, C2 in Example 3: constants in Bernoulli function.
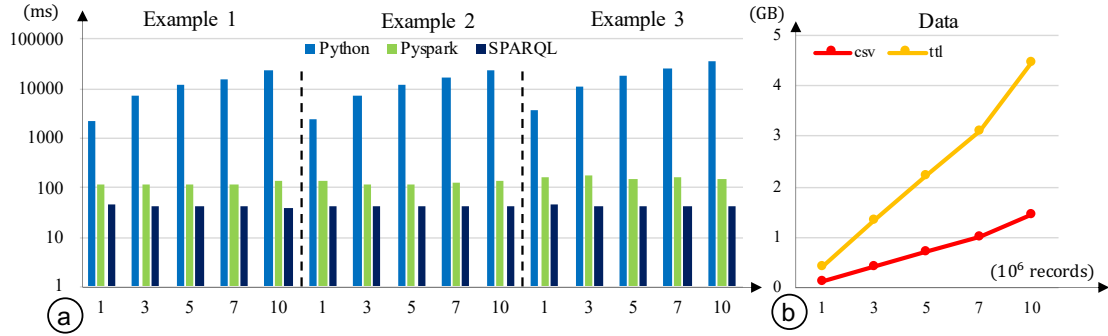
The figure (b) contains:

Example1: For each well section, the flowrate of fluid is always equal to upstream or downstream fluid flowrates.
$$Q_i = Q_j, \text{ for } i, j \text{ from the same Well}$$

```
SELECT ?ent WHERE {
    ?ent a :Entity ;
        :hasWellSection1Q ?q1 ;
        :hasWellSection2Q ?q2 ;
        :hasWellSection3Q ?q3 .
    FILTER (?q1 != ?q2 || ?q2 != ?q3 || ?q1 != ?q3)}
```

Example2: The sum of the subwell fluid flowrates equals to the fluid flowrate of the main well. $Q_t = \sum_i Q_{wi}$

```
SELECT ?ent WHERE {
    ?ent a :Entity ;
        :hasWell1Q ?q1 ;
        :hasWell2Q ?q2 ;
        :hasMainWellQ ?q0 .
    FILTER ( ABS(?q0 - ?q1 - ?q2) > 0.01)}
```

Example3: Constraint according to Bernoulli function:
$$Q = C1 * cv * \sqrt{(p_{uc} - p_{dc})/C2}$$

```
SELECT ?ent WHERE {
    ?ent rdf:type :Entity ;
        :hasChokeValue ?cv ;
        :hasChokeUpstreamP ?p1 ;
        :hasChokeDownstreamP ?p2 ;
        :hasWellSection1Q ?q .
    BIND (0.0007598* 0.0007598* ?cv * ?cv *(?p1-
?p2)/1.02 AS ?y)
    FILTER (ABS(?q*?q - ?y) < 0.01)}
```


Figure 3: (a) Running time of verification (x-axis: million records) (b) data size along number of records.

Experiment dataset. To test the verification performance, we generate a large number of data with a random ratio of integrity violations regarding examples 1-3 in Fig. 2b (real data contain few violations) based on real production data, provided by two world-leading energy companies. In total, we generated five such tables with sizes ranging from 145MB to 1.45GB, containing from 1 million to 10 million records. In addition, we generate corresponding KGs (Fig. 2a) following our ontology. These KGs are saved as Turtle files ranging in size from 440MB to 4.4GB.

Implementation. We implement the constraints in Example 1-3 with (a) Python, because it is relatively easy to learn and it is popular among the petroleum domain experts; (b) PySpark, for its similarity to Python and that it unlocks the potential of parallelizable computation;

(c) and SPARQL, for its popularity in the semantic community. The Python implementation uses common libraries such as Pandas, Numpy. PySpark is the Python API for Apache Spark, which is a distributed computing framework that enables parallel computation for dealing with large-scale datasets. The SPARQL is implemented with Jena and Fuseki. Jena is an open-source Java framework for semantic applications, while Fuseki is for setting SPARQL endpoint.

Results and discussion. From the results (Fig. 3) we can see that the Python running time increases significantly when the data size grows, while the running time for PySpark and SPARQL changes insignificantly. We postulate that the reason is that the data size is under a certain threshold so that the most consumed time for PySpark and SPARQL is used for loading the environment, not for querying. The results indicate that both PySpark and SPARQL have the potential for verifying large datasets. Yet, note that for generating the ttl files for SPARQL to query, it takes a large amount of time (some minutes to some hours). Besides, many domain experts are familiar with Python, but unfamiliar with Jena Fuseki and SPARQL. We expect they tend to learn writing constraint queries in PySpark than in SPARQL. All these factors need to be taken into account in considering industrial adoption.

References

[1] H. Ritchie, et al., Energy, Our World in Data (2022). Https://ourworldindata.org/energy.

[2] L. Kuang, et al., Application and development trend of artificial intelligence in petroleum exploration and development, Petroleum Exploration and Development 48 (2021) 1–14.

[3] J. Ge, Z. Li, T. Li, A novel chinese domain ontology construction method for petroleum exploration information., J. Comput. 7 (2012) 1445–1452.

[4] F. Cicconeto, L. V. Vieira, M. Abel, R. dos Santos Alvarenga, J. L. Carbonera, L. F. Garcia, Georeservoir: An ontology for deep-marine depositional system geometry description, Computers & Geosciences 159 (2022) 105005.

[5] N. Santos, et al., O3po: A domain ontology for offshore petroleum production plants, SSRN 4280151 (2022).

[6] Y. Qu, M. Perrin, A. Torabi, M. Abel, M. Giese, Geofault: A well-founded fault ontology for interoperability in geological modeling, arXiv preprint arXiv:2302.07059 (2023).

[7] P. F. Silva, L. F. Garcia, G. Figueiredo, R. J. de Moraes, R. K. Romeu, How do specialists express risks: an applied ontology for the oil & gas domain., in: ONTOBRAS, 2021, pp. 114–125.

[8] N. O. Santos, M. Abel, F. H. Rodrigues, D. Schmidt, Towards an ontology of offshore petroleum production equipment, CEUR-WS, 2022.

[9] R. Arp, et al., Building ontologies with basic formal ontology, MIT Press, 2015.

[10] B. Smith, et al., Relations in biomedical ontologies, Genome biology 6 (2005) 1–15.

[11] B. Smith, et al., A first-order logic formalization of the industrial ontologies foundry signature using basic formal ontology., in: JOWO, 2019.

[12] V. Kukkonen, et al., An ontology to support flow system descriptions from design to operation of buildings, Automation in Construction 134 (2022) 104067.