Building an Industrial Ontology Engineering Platform

Mikkel Haggren Brynildsen[1,*,†], Claus Jakobsen[2,†], Nicolaj Abildgaard[2,†] and Caitlin Woods[1,*,†]

[1]Grundfos, Poul Due Jensens Vej 7, 8850 Bjerringbro, Denmark
[2]Delegate, Hummeltofevej 49, 2830 Virum, Denmark

Abstract

The Industrial Ontology Engineering Platform (IEOP) is designed to bring ontology engineering (OE) to enterprise data processes while minimising the workload on OE practitioners. Built within an industrial context, IEOP emphasises enterprise data governance and maintenance requirements, while aligning ontologies to existing data sources and industrial upper ontologies. While IEOP has been a success in developing enterprise ontologies, presenting complex ontologies to domain experts remains a challenge.

Keywords

Ontology engineering, Enterprise ontology, Enterprise knowledge graphs, Ontology tool

1. Introduction

Smart manufacturing, digital twins, and Industry 4.0 have seen an explosion in the data volume and complexity of industrial data projects. New and existing data projects should use common models and definitions to control this growing complexity. Ontologies are a mechanism to implement controlled data models that leverages open-source, community-driven models. However, an evident skills shortage in Ontology Engineering (OE) limits industrial organisations' ability to implement ontologies. We have built the Industrial Ontology Engineering Platform (IOEP) to support domain experts in using and implementing ontologies while reducing the workload for OE specialists.

IOEP is built in collaboration with a manufacturing company and a software engineering company and is designed to meet the following goals: **(G1)** Position domain experts as the primary owners and contributors of new ontological models. **(G2)** Reduce the workload for OE practitioners. **(G3)** Meet enterprise data governance requirements throughout the OE process. **(G4)** Support both data creation and data ingestion from existing sources.

This work is inspired by a pipeline presented by DNV [1] that uses the Reasonable Ontology Templates (OTTR) framework [2]. IOEP similarly uses the OTTR framework, which informs many aspects of the platform's design. While IEOP is not an open source platform, we present the achievements and challenges in building and using this platform in an industrial organisation. We also highlight opportunities for future work.

ISWC'23: International Semantic Web Conference, November 06–10, 2023, Athens, Greece
*Corresponding author.
†These authors contributed equally.
mbrynildsen@grundfos.com (M. H. Brynildsen); cja@delegate.dk (C. Jakobsen); nab@delegate.dk (N. Abildgaard); cwoods@grundfos.com (C. Woods)
0009-0000-2521-1021 (M. H. Brynildsen); 0009-0007-5097-8364 (C. Jakobsen); 0000-0001-7829-7674 (C. Woods)

2. Implementation

The IOEP tool, pictured in Figure 1, provides a user interface for domain experts manage their data, and export that data to an ontology. Domain experts open a *workspace* where they can make edits to an ontology (that is designed by an OE practitioner in collaboration with domain experts). They are presented with a list of *source tables* containing the ontology's classes, individuals, properties and axioms in a tabular format. Data is added to an ontology by adding new rows to these source tables (using the "add row" button), or importing data from an external source such as Snowflake. When the domain expert clicks "build", a *build service* is invoked that uses OTTR templates (designed by the OE practitioner) to generate an OWL ontology using the tabular data. The core features of IOEP, and the goals addressed by each feature (e.g., G1) are:

1. Auto-generated Internationalized Resource Identifiers (IRIs): IOED assigns all new entities an immutable, unique IRI. The IRI is a prefix with a GUID appended on the end. Each entity also has a label column containing the human-readable name of the entity. Using system-controlled IRIs, IOED reduces the workload for OE practitioners because new entities use legal IRIs that follow the same conventions (G2) and the tool enforces consistent labels across workspace tables and rows.

2. Auto-generated governance fields: Governance fields are also automatically generated for all newly created entities. The governance fields *modifiedBy*, *modifiedDate*, *approvalStatus*, *modelStatus* and *changeNote* are enforced for every source table in IOEP (G3).

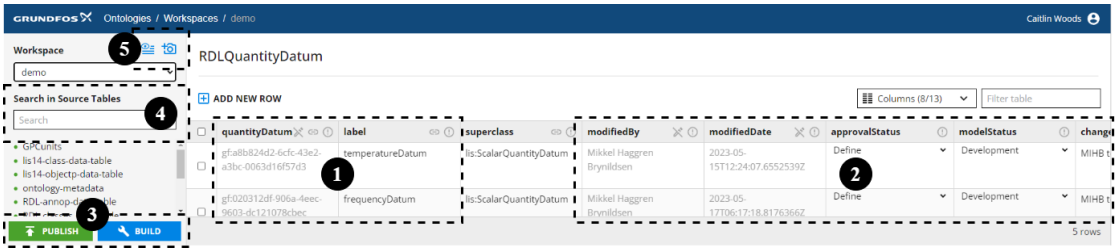

Figure 1: Data Entry Interface for IOEP (numbers correspond to features presented in Section 2)

3. Automated ontology building and publishing: The IOEP interface features "build" and "publish" buttons. Using the "build" button, domain experts can generate an ontology from their data (using the build service) and check that ontology in Protégé without the intervention OE expert (G2). Using the "publish" button, the source tables are pushed to source control for change tracking. Future work involves creating live linked data endpoints when the source tables are published.

4. Search: A search field supports the findability of entities in IOEP. In the example shown in Figure 1, searching IRI "gf:a8b824d2-dcfc-43e2-a3bc-0063d16f57d3" will filter the source table list and highlight references to that IRI in the source tables. This feature helps users understand the context of their data in the ontology and the implications of any changes they make (G1).

5. Workspace Snapshots: Users can create and reinstate snapshots of their workspaces. For example, after some changes a user may click the "build" button and realise there are issues in the generated ontology in Protégé. The user can roll-back the changes that they made using an old snapshot. This feature gives users confidence to experiment with changes to their data (G1).

Another feature that is currently under development is an *external tables* feature, allowing users to see tables from Snowflake in the tool and use those tables in generating the ontology. When a table is selected, IOEP will create a unique IRI for each entity in the table. By reading data from Snowflake (G4), this data can remain under governance in Snowflake without being duplicated in the tool (G3).

3. Successes, Challenges and Future Work

IOEP is currently in-use to develop three ontologies for three different project teams. A notable success of IOEP is its use of OTTR to improve its *scalability*. We can reuse existing projects' OTTR templates that describe ontology design patterns (ODPs). This re-usability reduces ontology design efforts and supports a common language for similar concepts across ontology projects. So far, two OE practitioners have been sufficient to oversee the ontological design, OTTR template development and maintenance processes of all three projects.

Another success of IOEP is the ability to abstract ontological complexity away from the domain expert. This abstraction is also largely due to our use of OTTR templates. We use the ISO15926-14 TR/CD upper ontology to ensure that the generated ontologies have the same conceptual underpinnings. Domain experts need a basic understanding of the concepts in the upper ontology (i.e. to assign entities to appropriate types and super-classes) to contribute to an IOEP project. However, they are not required to learn its object properties and axioms. This responsibility remains with the OE practitioner.

Some challenges in building and using IOEP are:

1. Representing complex ODPs in a tabular format: It is simple to explain to a domain expert how to model a simple class hierarchy in a tabular format. However, some ODPs require complex axioms such as property chains. Sometimes complex ODPs need additional columns in the data tables that are difficult to justify to domain experts. Future work involves finding a way to hide or automatically generate these columns.

2. Open standards for ontology governance: In our projects, we ground ontologies in open standards to improve the robustness of our designs. However, there are no commonly-used open standards for ontology governance. Thus, our governance fields are decided within our organisation. Future work involves developing open standards for ontology governance that we can use across different projects.

References

[1] J. W. Klüwer, READI for the CFIHOS RDL: An OTTR use case, 2021. URL: https://www.ottr.xyz/event/2021-01-28-user-forum/slides-johan.pdf.

[2] M. G. Skjæveland, D. P. Lupp, L. H. Karlsen, J. W. Klüwer, Ottr: Formal templates for pattern-based ontology engineering., in: WOP (Book), 2021, pp. 349–377.