

ODDToolkit: An Open-Source Toolkit to Help Bridge the Developer Gap in Ontology Driven Design

Dr. Maxim Van de Wynckel &
Geert Van Haute

SEMANTICS 2026

Problem. How do we iteratively engineer an ontology that developers with no knowledge on linked data can validate without spending too much time on documentation? And how do we ensure they do get familiar with linked data?

1

Input from OWL ontology,
SKOS concepts and developer preferences

2

Infer main classes and attributes

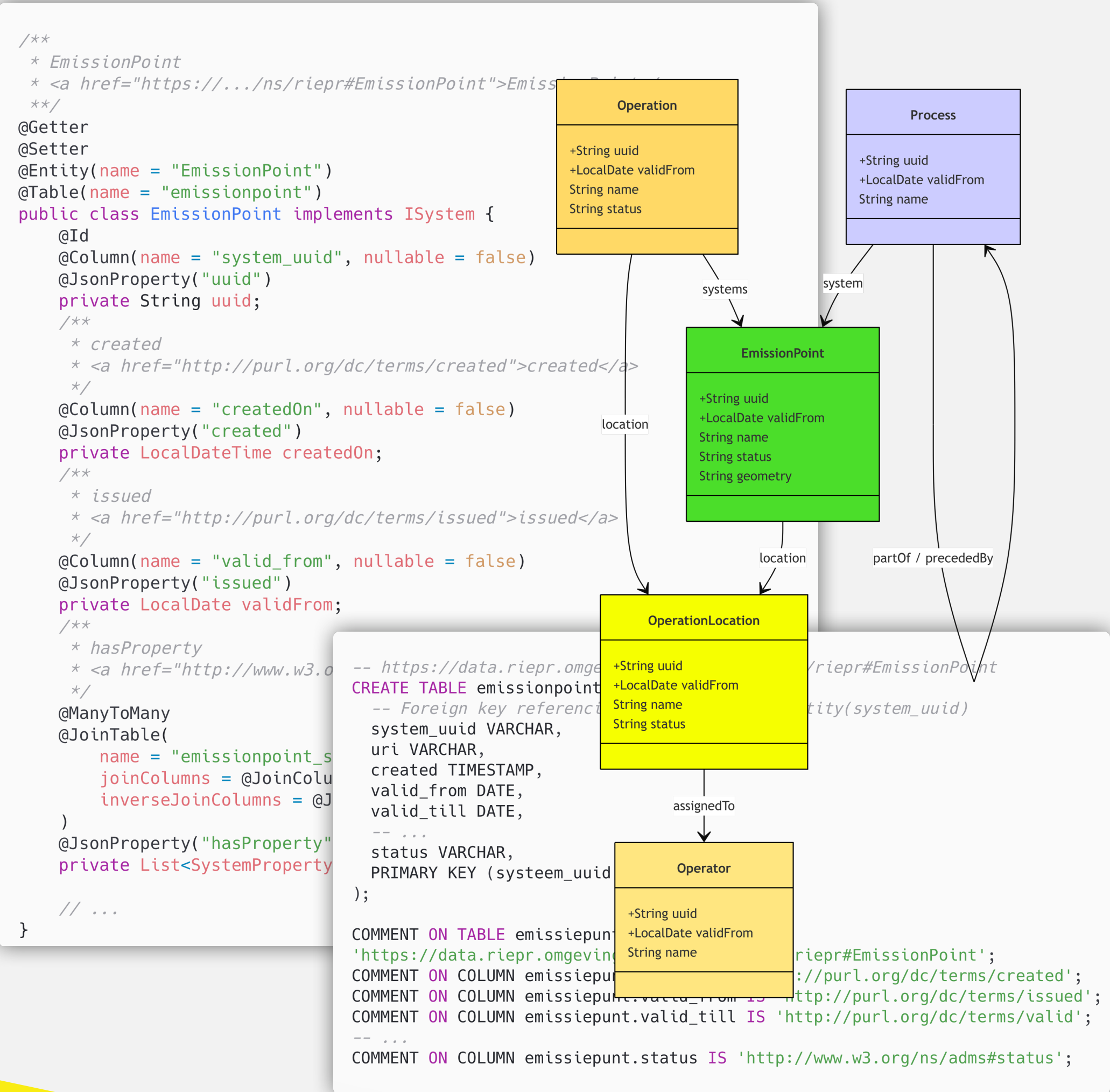
- Concrete classes.** owl:Class from the ontology or external classes used in the SKOS concepts.
- Interfaces.** External RDF classes that are subclasses of more than one concrete class AND used as a relation.
- Cardinality.** combination of owl:*cardinality and owl:inverseOf to determine 1..1, 1..* or *..*
... Other inference and attributes built as modular blocks

3

Map with SKOS concepts and identify
primary keys (e.g., using Hydra)

4

Deterministic output with references to the ontology



Abstract

Linked open data remains **difficult to operationalise** for software teams when OWL-based models are **documented** primarily for **ontology engineers** rather than for implementers. Furthermore, maintaining documentation for implementers becomes time consuming when the ontology is under development. We present ODDToolkit, an open-source Ontology Driven Design toolkit that helps **generate developer-facing artefacts** from three inputs: an OWL ontology, a SKOS concept scheme that provides business-specific names, and a configuration file for developer-specific design choices. The toolkit emits SQL DDL, Java and TypeScript models, SHACL shapes, diagrams, and documentation. Our design goal is not to fully automate translation of arbitrary vocabularies, but to achieve **repeatable regeneration of implementation artefacts as an ontology evolves**. We summarise the pipeline and report its use in a Flemish government environmental reporting project, where the generated artefacts supported implementation, documentation, and validation around a shared ontology as it evolves.

Use Case

ODDToolkit is used in the Flemish Government as a method to communicate the ontology-driven data model for industrial emission reporting (IED-IEPR EU regulation). While the ontology evolves to align with new requirements, developers are minimally impacted while still receiving up-to-date documentation based on the latest ontology design.

ODDToolkit's philosophy is that using linked data should begin in the IDE, not just during publication.

GNU General Public License v3.0

