OWL-SDA: A Multi-Agent System for Generating Synthetic Data to Support Ontology Evaluation

Maxim Van de Wynckel[1,2,*], Emmelien De Roock[3,4]

[1]*Department of Environment and Spatial Development, Flemish Government, Belgium*
[2]*Web & Information Systems Engineering Lab, Vrije Universiteit Brussel, Belgium*
[3]*UZ Brussel, Belgium*
[4]*Data Analytics Laboratory, Vrije Universiteit Brussel, Belgium*

Abstract

Ontology engineers often need representative RDF instances to validate and document an ontology, but such data may be unavailable, incomplete, too sensitive to share or simply too much work to create manually. We present OWL-SDA, an open-source toolkit that uses worker, supervisor and reviewer AI agents to generate synthetic RDF data from ontology context and SHACL constraints. The workflow combines shape-level task decomposition, direct manipulation of an in-memory triple store, iterative SHACL repair and a final review stage aimed at catching issues that remain possible even after SHACL conformance. We evaluate OWL-SDA on two contrasting ontologies, one for industrial emissions reporting and one for healthcare records. The results show that richer ontology constraints lead to faster convergence and more informative examples, while sparse ontologies require more supervision and still yield less detailed data. We therefore position the toolkit for ontology engineers to help document their ontology, but also to validate if the ontology alone is sufficiently rich to be understood by non-experts.

Keywords

synthetic data, OWL, AI agents, ontology-driven documentation

1. Introduction

Reusability of ontologies depends heavily on documentation and on the availability of representative instance data [1]. However, in emerging or privacy-sensitive domains, such data is often missing, incomplete or difficult to share, which limits empirical ontology evaluation and pushes validation back toward expert judgement [2]. Tool support is already central in ontology engineering [3], and recent LLM-based work has mostly focused on ontology design assistance, reasoning support or taxonomy construction [4–6].

OWL-SDA instead targets ontology evaluation and documentation. It uses AI agents with tools and structured tasks [7–9] to generate RDF examples when real data cannot be shared because of sensitivity, governance or context-window constraints [10–14], or when creating data examples would be too costly or time-consuming. Rather than proposing a new synthetic-data model, OWL-SDA combines shape-level generation, iterative repair and role separation for ontologies expressed in OWL [15]. The approach fits within LLM-based synthetic data generation [16], but prioritises structurally valid and interpretable linked-data examples while

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium
EMAIL: maxim.vandewynckel@vlaanderen.be (M. Van de Wynckel)
ORCID: 0000-0003-0314-7107 (M. Van de Wynckel); 0000-0001-9680-5222 (E. De Roock)

addressing incompleteness and inaccuracy risks [17]. We position OWL-SDA against existing work in Section 2, and detail our workflow in Section 3. Finally, in Section 4, we evaluate OWL-SDA against two contrasting scenarios.

2. Positioning

OWL-SDA is related to, but distinct from, three lines of work. First, recent LLM research in ontology engineering has mainly focused on reasoning support, ontology design assistance or taxonomy construction [4–6]. Our target is narrower and more operational: generating example RDF that helps ontology engineers validate and document an existing ontology to promote its (re)usability without spending too much effort on the creation.

Second, synthetic data generation itself is broader than ontology population. Existing surveys distinguish rule-based, statistical and machine-learning approaches [16]. OWL-SDA belongs to the LLM-based family, but it does not aim to reproduce the full statistical properties of a real dataset. Instead, it prioritises structurally valid and interpretable linked-data examples, while explicitly addressing incompleteness and inaccuracy risks highlighted in existing literature [17]. Our solution is therefore complementary to existing synthetic-data models, and can be used to generate data for ontologies while still in development, when real data is unavailable, incomplete, too sensitive to share, or simply too resource intensive to create manually.

Third, OWL-SDA borrows from multi-agent coordination rather than from single-prompt generation. Hierarchical supervision has already been explored for software-oriented agent systems [18, 19], and tool availability is one of the main distinctions between agents and plain LLMs [7]. Our contribution is to combine these ideas with SHACL-based task distribution, ontology-aware tools, SHACL-based repair and a final reviewer role so that quality control is distributed across the stages where it is most effective.

3. Approach

OWL-SDA (OWL Synthetic Data AI-Agent) is an open-source[1] Java toolkit for generating synthetic RDF data from ontology context and user constraints[2]. Early design prototypes of the toolkit showed two recurring problems: direct Turtle editing by agents produced syntax errors and duplicate triples, while delayed validation caused an accumulation of violations. OWL-SDA addresses both by moving edits into an in-memory triple store and by exposing SHACL validation as a continuous feedback loop, which helps mitigate the incompleteness and inaccuracy risks also identified by Hao et al. [17] in the creation of synthetic data.

The final design follows role-based task decomposition [20] and hierarchical supervision patterns [18, 19]. Workers handle narrow shape-level tasks that are sorted based on their complexity and links to other shapes; the supervisor coordinates scope and consistency (e.g., URI consistency or scenario); and the reviewer evaluates the final graph from a read-only perspective. This separation matters because SHACL-valid output can still contain invented vocabulary, inconsistent naming or implausible combinations across shapes.

[1] https://github.com/milieuinfo/owl-sda
[2] https://purl.org/owl-sda

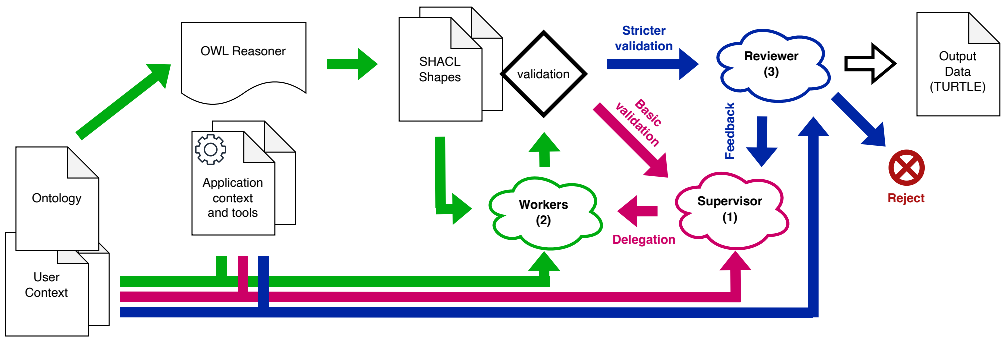


Figure 1: Architecture of the OWL-SDA pipeline

Figure 1 summarises the workflow. During preparation, OWL-SDA loads the ontology, downloads imported vocabularies and derives two SHACL views: one used by workers for strict conformance checks and one used later by the reviewer for broader assessment. The supervisor (1) then assigns shape-level tasks to workers (2), who add and remove triples in a shared concurrent store rather than editing Turtle text directly. Once all shapes are processed, the supervisor (3) performs the final check before sending it to the reviewer (4), who inspects the serialised output.

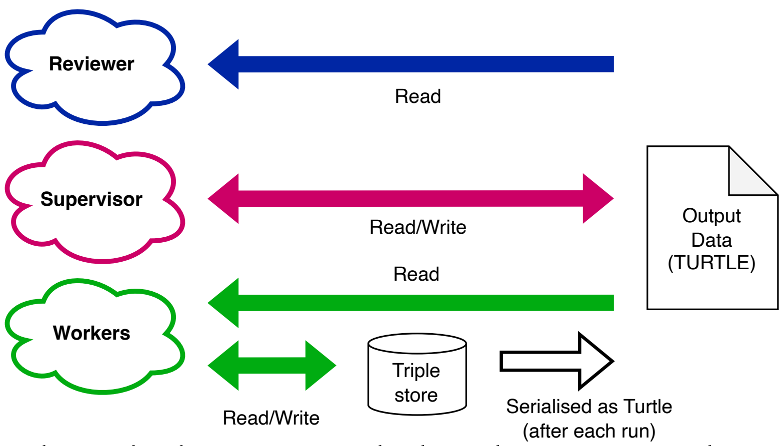


Figure 2: Output data read and write operations by the workers, supervisor and reviewer

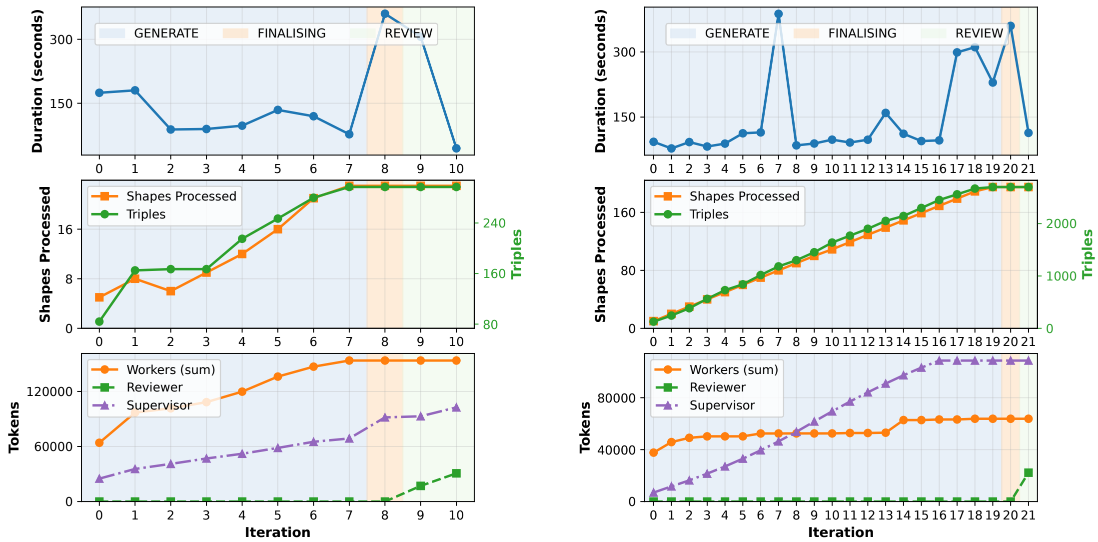
In order to enable collaboration between multiple workers and to facilitate consistency, workers do not directly edit the output data. As illustrated in Figure 2, the workers have access to a shared concurrent triple store that enables them to add, remove and read triples. The triple store provides workers with instant feedback on syntax issues, duplicated triples and helps with the deletion of orphaned blank nodes. Workers can validate the output of the triple store during generation and correct problems immediately.

In a typical run, the supervisor reads the ontology, SHACL shapes and user context; workers generate triples for a small set of target shapes; each worker validates and repairs its own output; the supervisor validates the combined graph and dispatches targeted fixes for cross-shape inconsistencies; and finally a reviewer inspects the serialised output for broader plausibility issues. This keeps workers focused on local conformance while reserving global quality control for the more capable roles. OWL-SDA currently supports GitHub Copilot, Ollama and OpenAI-compatible endpoints, with provider selection configurable globally or per role. In practice, we used smaller, faster models for workers and stronger models for supervision and review, which follows cost-quality trade-offs also observed in multi-agent systems [21].

During its development, we tweaked the workflow to introduce additional guardrails to catch common errors. For example, workers would sometimes create instance data that was using a different URI scheme than the ontology, which was not caught by SHACL validation. Additional checks were added as system prompts to the supervisor and reviewer to help guide workers towards producing more consistent output.

4. Evaluation

We evaluate OWL-SDA on two contrasting ontologies: an industrial emissions ontology that motivated the toolkit and a healthcare ontology with fewer constraints. The evaluation is exploratory: we focus on convergence behaviour, review findings and the practical usefulness of the generated examples rather than on baseline comparisons or statistical similarity to gold-standard datasets. Although OWL-SDA supports multiple backends, including self-hosted and OpenAI-compatible deployments, all runs reported here used `gpt-5-mini` workers, a `gpt-5.4` supervisor and a `claude-4.6-sonnet` reviewer.



IED-IEPR benchmark

OpenCareOntology benchmark

Figure 3: Benchmark comparison for the two evaluation scenarios.

Figure 3 shows the per-iteration processing times. The first iterations are consistently slower because the agents load the ontology and task context, after which later batches benefit from narrower scopes and lower-complexity shapes. For both runs, the LLM messages and their results are available as benchmarks in the OWL-SDA repository.

4.1. Industrial Emissions Ontology

The first scenario uses a domain-specific ontology for Industrial Emissions Directive reporting [22]. It models industrial systems and measurement setups as `ssn:System` instances and relates them to operational plans and steps from P-Plan, namely `p-plan:Plan` and `p-plan:Step`. The ontology was one of the catalysts for OWL-SDA and was selected because it combines rich constraints with a concrete documentation need. The ontology used in this evaluation is available on GitHub but was translated to English for the purposes of this paper[3]. This scenario produced the strongest results: richer constraints and property descriptions enabled SHACL-conformant output with limited supervisor steering, although qualitative inspection still exposed underspecified ontology ranges such as `prov:Agent` that were later used as feedback to further document the ontology.

```
<https://example.org/id/installation/glass-former>
  a :Installation ;
  rdfs:label        "Glass Forming Machine"@en ;
  dc:created        "2024-06-15T09:30:00"^^xsd:dateTime ;
  dc:issued         "2021-06-15"^^xsd:date ;
  dc:modified       "2024-06-15T09:30:00"^^xsd:dateTime ;
  adms:status       concept:active ;
  sosa:isHostedBy   <https://example.org/id/location/myGlassFactory> ;
  ssn:hasProperty   <https://example.org/id/installationproperty/glassformer-power> .

<https://example.org/id/installationproperty/glassformer-power>
  a :InstallationProperty ;
  dc:type             concept:installedPower ;
  qudt:numericValue   "2.5"^^xsd:decimal ;
  qudt:hasUnit        unit:MegaW ;
  ssn:isPropertyOf    <https://example.org/id/installation/glass-former> .
```

Listing 1: Partial AI-generated output for the industrial emissions ontology.

Listing 1 shows a partial example of the generated output. The `Installation` instance is linked to a `InstallationProperty` instance that describes its installed power. While this is a simple example, producing a larger set of such instances would be time consuming for a human.

4.2. Healthcare Ontology

The second scenario uses the OpenCareOntology[4] [23] to generate synthetic patient history records. This scenario is relevant because using real patient data as prompt context is often undesirable or prohibited [12, 13]; self-hosted alternatives can mitigate that, but may be too costly in

[3] https://github.com/milieuinfo/RIE-IEPR/
[4] https://bioportal.bioontology.org/ontologies/OCO

smaller settings [24]. Moreover, pseudonymisation does not remove all re-identification risk [25], while full anonymisation may remove information needed for correct relation generation [26]. Compared with the industrial ontology, OpenCareOntology provides broader healthcare coverage but fewer constraints and less in-ontology documentation. As a result, OWL-SDA required more iterations and more supervisor intervention, and the generated instances had fewer properties on average, although the output remained useful as documentation.

This difference between scenarios was also visible in review feedback. In the industrial case, most fixes concerned missing links or underspecified ranges in an otherwise rich graph. In the healthcare case, the reviewer more often had to judge whether the produced combinations were merely SHACL-conformant or also clinically plausible, which reinforces the need for a separate review stage beyond validation alone.

5. Conclusion and Future Work

OWL-SDA provides ontology engineers with a practical way to generate synthetic RDF examples for validation and documentation when real data cannot be shared or is unavailable. Across both evaluation scenarios, richer ontology constraints and documentation led to faster convergence and more useful output, whereas sparse ontologies demanded more supervisor intervention and still produced less detailed instances [1, 6]. Our toolkit helps both to produce example data and to expose where an ontology remains underspecified. Because the generation does not require user supervision, it can run alongside ontology design to help evaluate its fitness for purpose.

The current evaluation does not claim statistical fidelity to real-world data. Repeated runs may differ because LLM behaviour is non-deterministic, and our results are limited to two ontologies and to the specific model combination used. Most importantly, SHACL conformance is necessary but not sufficient for quality: data can satisfy all shapes while remaining semantically implausible, for example when a clinically unlikely diagnosis and treatment combination is represented with valid classes and properties. This is precisely why OWL-SDA retains a separate review stage. In the industrial emissions reporting use case, generated examples are already being used to help illustrate the ontology while the ontology is still under development.

Future work should add human-in-the-loop steering [27] and broaden the evaluation with expert assessment and, where legally possible, comparisons against reference data. More generally, the current reviewer and supervisor routine could be extended with human checkpoints and alternative validation strategies, so that domain experts can assess usefulness, plausibility and fitness for purpose in addition to SHACL conformance.

Supplemental Material. OWL-SDA and the benchmark results from the initial release are available in the project repository [28].

Declaration on Generative AI. During the creation of OWL-SDA, the authors used GitHub Copilot to help with the documentation and development. The authors take full responsibility for the content in the repository.

References

[1] R. Alharbi, Assessing Candidate Ontologies for Reuse, in: Proceedings of the Doctoral Consortium at ISWC 2021, 2021, pp. 1–12.

[2] J. Brank, M. Grobelnik, D. Mladenic, A Survey of Ontology Evaluation Techniques, in: Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005), volume 17, 2005, pp. 166–170.

[3] P. Hitzler, K. Janowicz, T. Tudorache, Ontology Engineering: Current State, Challenges, and Future Directions, Semant. Web 11 (2020) 125–138. doi:10.3233/SW-190382.

[4] E. Damiano, F. Orciuoli, Evaluating Large Language Models on OWL Lite Reasoning, in: Proceedings of the Second Workshop on Knowledge Graphs and Neurosymbolic AI (KG-NeSy), co-located with SEMANTiCS'25, 2025, pp. 1–12.

[5] H. Yang, J. Chen, U. Sattler, Large Language Model for OWL Proofs, arXiv preprint arXiv:2601.12444 (2026).

[6] F. M. Soares, A. M. Saraiva, L. F. Pires, L. O. B. d. S. Santos, D. d. A. Moreira, F. E. Corrêa, K. R. Braghetto, D. P. Drucker, A. C. B. Delbem, Exploring a Large Language Model for Transforming Taxonomic Data into OWL: Lessons Learned and Implications for Ontology Development, arXiv preprint arXiv:2504.18651 (2025).

[7] A. Yehudai, L. Eden, A. Li, G. Uziel, Y. Zhao, R. Bar-Haim, A. Cohan, M. Shmueli-Scheuer, Survey on Evaluation of LLM-based Agents, arXiv preprint arXiv:2503.16416 (2025).

[8] P. Vaithilingam, T. Zhang, E. L. Glassman, Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models, in: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems Extended Abstracts, CHI EA '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 1–7. doi:10.1145/3491101.3519665.

[9] B. Webber, N. Badler, B. Di Eugenio, C. Geib, L. Levison, M. Moore, Instructions, Intentions and Expectations, Artificial Intelligence 73 (1995) 253–269. doi:10.1016/0004-3702(94)00013-Q, computational Research on Interaction and Agency, Part 2.

[10] B. Yan, K. Li, M. Xu, Y. Dong, Y. Zhang, Z. Ren, X. Cheng, On Protecting the Data Privacy of Large Language Models (LLMs) and LLM Agents: A Literature Review, High-Confidence Computing 5 (2025) 100300. doi:10.1016/j.hcc.2025.100300.

[11] Z. Wang, H. Yuan, W. Dong, G. Cong, F. Li, CORAG: A Cost-Constrained Retrieval Optimization System for Retrieval-Augmented Generation. CoRR abs/2411.00744 (2024), arXiv preprint arXiv:2411.00744 (2024).

[12] V. B. Vallevik, A. K. C. Befring, S. Elvatun, J. F. Nygaard, Processing of Synthetic Data in AI Development for Healthcare and the Definition of Personal Data in EU Law, arXiv preprint arXiv:2508.08353 (2025).

[13] B. Kaabachi, J. Despraz, T. Meurers, K. Otte, M. Halilovic, B. Kulynych, F. Prasser, J. L. Raisaro, A Scoping Review of Privacy and Utility Metrics in Medical Synthetic Data, npj Digital Medicine 8 (2025) 60. doi:10.1038/s41746-024-01359-3.

[14] S. Naik, Cloud-Based Data Governance: Ensuring Security, Compliance, and Privacy, East Asia South Institute of Computer Science 1 (2023) 69–87. URL: https://esj.eastasouth-institute.com/index.php/esiscs/article/view/452. doi:10.58812/esiscs.v1i01.452.

[15] G. Antoniou, F. v. Harmelen, Web Ontology Language: OWL, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 91–110. doi:10.1007/978-3-540-92673-3_4.
[16] M. Goyal, Q. H. Mahmoud, A Systematic Review of Synthetic Data Generation Techniques Using Generative AI, Electronics 13 (2024). URL: https://www.mdpi.com/2079-9292/13/17/3509. doi:10.3390/electronics13173509.
[17] S. Hao, W. Han, T. Jiang, Y. Li, H. Wu, C. Zhong, Z. Zhou, H. Tang, Synthetic Data in AI: Challenges, Applications, and Ethical Implications, arXiv preprint arXiv:2401.01629 (2024).
[18] B. Banihashemi, G. De Giacomo, Y. Lespérance, et al., Hierarchical Agent Supervision, in: Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, volume 2, International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2018, pp. 1432–1440.
[19] Y. Long, AI-Supervisor: Autonomous AI Research Supervision via a Persistent Research World Model, arXiv preprint arXiv:2603.24402 (2026).
[20] G. Li, H. Hammoud, H. Itani, D. Khizbullin, B. Ghanem, CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023), volume 36, Curran Associates, Inc., 2023, pp. 51991–52008. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/a3621ee907def47c1b952ade25c67698-Paper-Conference.pdf.
[21] L. Chen, M. Zaharia, J. Zou, Less is More: Using Multiple LLMs for Applications with Lower Costs, in: Proceedings of the Workshop on Efficient Systems for Foundation Models at ICML 2023, 2023, pp. 1–9. URL: https://openreview.net/forum?id=TkXjqcwQ4s.
[22] European Parliament and Council of the European Union, Regulation (EU) 2024/1244 of the European Parliament and of the Council of 24 April 2024 on Reporting of Environmental Data from Industrial Installations, Establishing an Industrial Emissions Portal and Repealing Regulation (EC) No 166/2006, https://eur-lex.europa.eu/eli/reg/2024/1244/oj/eng, 2024. Official Journal of the European Union, OJ L, 2024/1244, 2 May 2024.
[23] B. Dutta, D. Bain, OpenCareOntology, 2024. URL: https://bioportal.bioontology.org/ontologies/OCO, nCBO BioPortal entry for OpenCareOntology (OCO), version 1.1.
[24] M. Potluri, LOIS: Learning-Optimized Inference System for Cost-Efficient Large Language Model Deployment, Available at SSRN 6118086 (2026).
[25] K. El Emam, E. Jonker, L. Arbuckle, B. Malin, A Systematic Review of Re-Identification Attacks on Health Data, PLOS ONE 6 (2011) 1–12. doi:10.1371/journal.pone.0028071.
[26] F. J. Sarmin, A. R. Sarkar, Y. Wang, N. Mohammed, Synthetic Data: Revisiting the Privacy-Utility Trade-Off, International Journal of Information Security 24 (2025) 156. doi:10.1007/s10207-025-01072-6.
[27] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, L. He, A Survey of Human-in-the-Loop for Machine Learning, Future Generation Computer Systems 135 (2022) 364–381. doi:10.1016/j.future.2022.05.014.
[28] M. Van de Wynckel, E. De Roock, OWL-SDA - GitHub Repository, 2026. doi:10.5281/zenodo.21489630.