RDF Differ: Configurable Change Reporting for RDF Vocabularies

Rashif Rahman[1,*], Jana Ahmad[1], C. Maria Keet[1] and Eugeniu Costetchi[1]

[1]*Meaningfy SARL, 61 route de Fischbach, Lintgen, Luxembourg*

Abstract

Identifying and interpreting changes in RDF vocabularies is essential for provenance, auditing, and collaborative knowledge engineering. Traditional diffing approaches, often operating at the line level, fail to capture semantically meaningful changes without awareness of schema constraints and application profiles (APs). This paper presents RDF Differ, a diff tool for schema-aware comparison of RDF vocabularies, explicitly designed to leverage configurable templates for APs to maximise portability and precision. The tool is built upon the SPARQL-based semantic change detection algorithms of skos-history, and integrates the template-based, data-driven report generation capabilities of eds4jinja2. Configuration of the AP templates along with their automated SPARQL query synthesis is facilitated by the diff-query-generator (dqgen). This architecture enables dynamic, profile-driven vocabulary diffing, ensuring that comparisons are both scalable and semantically relevant. We introduce the rationale, design, and architecture of RDF Differ, and demonstrate its value by comparing it to other diffing tools using releases of the eProcurement ontology (ePO).

Availability: RDF Differ is open-source and available at https://github.com/meaningfy-ws/rdf-differ-ws

Keywords

RDF, Knowledge Graph Evolution, Linked Data Schema Evolution, Change Tracking

1. Introduction

Change detection or *diffing* of evolving data is a foundational task in IT. Diffing enables change tracking, auditing, and collaborative data management (*a la* Git). However, traditional diffing methods lack a comprehensive change *reporting* mechanism and typically rely on syntactic or line-based comparisons, failing to account for the rich context encoded in RDF-based knowledge organisation systems.

As RDF vocabularies evolve, both tracking and reporting on schema-level changes become essential for collaborative knowledge engineering. Yet, conventional approaches struggle to surface semantically meaningful change reports, especially in contexts guided by schema constraints and Application Profiles (APs). Specifically, while diff tools exist for OWL [1] and RDF instance data [2], they are typically either mostly syntactic or tailored to expressive OWL ontologies, and miss changes that matter in lightweight vocabularies like SKOS, such as label updates, hierarchy adjustments, or annotation changes. An OWL-specific diff tool may not surface label deprecations, while an RDF-specific one may lack certain semantic awareness needed to detect schema-level evolution (see below for a tool comparison). Lightweight ontologies and vocabularies can therefore benefit from a diffing solution that provides a middle ground between line changes and axiomatic differences, with a configurable reporting framework for different APs.

To address this gap, we introduce RDF Differ, a schema-aware RDF diff tool that supports dynamic, profile-driven vocabulary comparisons through a web API and a GUI. It leverages configurable AP templates and SPARQL-based change detection to support dynamic, context-sensitive comparisons. Building on the foundations of skos-history [3] and facilitating report generation (in HTML or JSON) via

Posters and Demos of SEMANTiCS'25, co-located with SEMANTiCS'25: International Conference on Semantic Systems, September 3–5, 2025, Vienna, Austria

*Corresponding author.

rashif.rahman@meaningfy.ws (R. Rahman); jana.ahmad@meaningfy.ws (J. Ahmad); maria.keet@meaningfy.ws (C. M. Keet); eugen@meaningfy.ws (E. Costetchi)

0009-0002-6469-2571 (R. Rahman); 0000-0001-6689-8587 (J. Ahmad); 0000-0002-8281-0853 (C. M. Keet); 0000-0002-9862-5070 (E. Costetchi)

Table 1
Comparison of RDF and OWL Diff Tools. Lang.: language (logic) which for OWL may be a particular species (not verified by us); Profile config.: user-configurable 'profiles' in the sense that the user can freely choose which language features to compare on; Report: whether the computed diff can be saved; l. upd.: last update.

Tool	Lang.	Profile Config.	Diff Algorithm	GUI	Report	Deployment Mode	Technology Stack	Activity
OWL Difference [4]	OWL	No	Structural hierarchy diff	Yes	No	Protégé plugin	OWL API, Java	Inactive (l. upd. ≥10 yrs)
OWLDiff [1]	OWL	No	DL reasoning, justification-based	Yes	No	Standalone (CLI/GUI), plugins	Java, OWL API	Inactive (l. upd. ≥4 yrs)
Ecco [5]	OWL	No	Structural + semantic categorisation	Yes	Yes	Standalone (CLI/GUI)	Java, OWL API	Inactive (last update ≥11 yrs)
Quit Diff [6]	RDF	No	Triple comparison (structural)	No	No	Standalone (CLI), Git plugin	Python, RDFLib	Active (l. upd. ≥10 months)
rdfdiff (Jena) [7]	RDF	No	Triple-level structural diff	No	No	Standalone (CLI)	Java, Apache Jena	Active (l. upd. this month)
skos-history [3]	SKOS	No	SPARQL query-based	No	Possible	Scripts and YASGUI config	Bash	Inactive (l. upd. ≥1 yr)
RDF Differ	*SKOS, RDF*	*Yes (user configurable)*	*SPARQL query-based, fully automated*	*Yes*	*Yes*	*Web-based (self-hosted via Docker or local stack)*	*Python, SPARQL, Jena Fuseki*	*Active (l. upd. this month)*

a SPARQL-enabled Jinja templating extension eds4jinja2,[1] the toolchain enables a modular architecture for vocabulary change detection and reporting.

In this paper, we first assess related work (Section 2) and then describe the tool in Section 3, with the design decisions and architecture (Section 3.1), a use case with the eProcurement ontology, and a comparison with OWL Difference (Section 3.2). Section 4 presents our conclusions.

2. Related Work

A variety of tools and research approaches have been proposed to compute differences between RDF graphs and between OWL ontologies, using tailor-made algorithms, query-based approaches, and syntactic line-by-line comparisons; see Table 1. These solutions address diverse needs such as ontology versioning, synchronisation, and collaborative editing. We briefly summarise key points in this section.

Computing differences for RDF graphs are typically command-line-based (e.g., Quit Diff [6], rdfdiff [7] or integrated in other tools (Quit Store [2] and Jena). Alternatively, they may compute triple-level differences [8], which lack semantic context. Geared towards the SKOS flavour of RDF but adaptable in principle, skos-history [3] describes an approach for producing RDF deltas using SPARQL MINUS operations and named graphs. Tools to compute the difference for OWL are more sophisticated, because they seek to compute also the semantic (entailment) differences to some extent. For instance, Ecco [5] combines structural and semantic analysis, categorising changes like strengthenings, weakenings, and additions, while OWLDiff [1] provides justification-based explanations for changes, emphasising semantic precision. OWL Difference [4], a plugin to the Protégé ontology editor, focuses on detecting structural and hierarchical changes, closer to the basic needs of the general RDF audience.

Most of these tools were easily deployable, with the exception of Ecco and OWLDiff, which required advanced Java software setup know-how due to the lack of release binaries and inconsistent instructions.

[1]https://github.com/meaningfy-ws/eds4jinja2

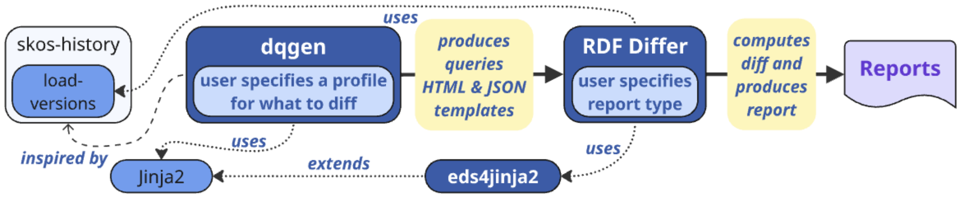

Figure 1: The RDF Differ pipeline architecture (see text for details).

3. The Tool: RDF Differ

RDF Differ offers schema-aware change tracking, application profile-based diffing for RDF vocabularies and ontologies, including SKOS and some OWL features. Inspired by skos-history [3], RDF Differ extends the concept of SPARQL-based change detection to provide template-driven, semantic diffs that reflect meaningful changes in resources, properties and values, enabling structured changelogs and easing change reporting. It is a Python web application supported by additional tools and libraries, collectively forming a pipeline.

3.1. System Design and Implementation

An overview of the RDF Differ pipeline architecture is shown in Fig. 1. The diff-query-generator (dqgen) allows defining templates for APs through CSV files listing all of the resource types (classes and properties) to detect. SPARQL queries, HTML and JSON templates are then automatically generated for each resource and change type. Given an AP, RDF Differ executes the relevant queries, adopting the skos-history technique with the `load-versions` script to populate delta graphs, and automating the diffing logic. RDF Differ allows the user to upload the files to be compared while associating diffing metadata, and generates one or more reports with a desired AP and format. The computed diff results are then made available for download. HTML reports are structured by resource category and change type, offering human-readable summaries, while JSON is intended for machine consumption.

The diff types supported are: addition, deletion, update, move and change. An update is the change of a value of a literal or object, while a move is the shifting of a value from one resource (class or property) to another, and a change refers to the shifting of a value from one property to another within the same resource. Language tags are also accounted for.

We model change as a state transition operator ($\Rightarrow$) between old (on the left) and new (on the right), and either old or new is checked, or both are. Note that the changes are instances of patterns, but not instances in the sense of an individual in the RDF graph. To indicate this distinction in the inventory of types of changes (see Table 2), we use ι (subject) and ν (object/value) in the domain and range position/filler of property ρ, which may be subscripted if more than one appears in the pattern, with $1 \leq i \leq n$ (and $n \in \mathbb{N}$). In addition, ρ' denotes the secondary predicate in a property chain ('reified' in SKOS terms) that is indicated with a / between properties, $@l$ the language tag of the value, if present, and $\emptyset$ denotes the null (i.e., absent before or after the change).

The system supports running in a microservice setting using Docker. The stack includes: a Flask API and server-rendered UI, which are served by Gunicorn; Celery for asynchronous task management; an RDF triplestore (Fuseki); Redis for asynchronous task queues; Docker Compose for microservices orchestration; and Traefik for production deployments.

3.2. Use and Evaluation

The aims of this brief evaluation are to 1) show that it works for a business case-based evaluation, 2) compare it against the identified case's current practices, and 3) compare RDF Differ against the relative closest other diff tool while demonstrating differences.

Table 2

Change type inventory. Abbreviations: Elem.: element; no lang tag: no language tag; lang. tag: language tag; reif p val: value of a reified/secondary property; reif: change detection of the reified/secondary object; l/old chk: Left condition checking and r/new chk: Right condition checking, i.e., whether an existence or non-existence check applies on that side of the transition operator.

Change pattern	Elem.	no lang. tag	lang. tag	reif p val	lang. tag	reif	l/old chk	r/new chk
Addition	$\emptyset \Rightarrow \iota$	$\emptyset \Rightarrow \iota\rho\nu$	$\emptyset \Rightarrow \iota\rho\nu@l$	$\emptyset \Rightarrow \iota\rho/\rho'\nu$	–	–	–	x
Deletion	$\iota \Rightarrow \emptyset$	$\iota\rho\nu \Rightarrow \emptyset$	$\iota\rho\nu@l \Rightarrow \emptyset$	$\iota\rho/\rho'\nu \Rightarrow \emptyset$	–	–	x	–
Update	–	$\iota\rho\nu_1 \Rightarrow \iota\rho\nu_2$	$\iota\rho\nu_1@l \Rightarrow \iota\rho\nu_2@l$	$\iota\rho/\rho'\nu_1 \Rightarrow \iota\rho/\rho'\nu_2$	$\iota\rho/\rho'\nu_1@l \Rightarrow \iota\rho/\rho'\nu_2@l$	–	x	x
Move (subject)	–	$\iota_1\rho\nu \Rightarrow \iota_2\rho\nu$	–	$\iota_1\rho/\rho'\nu \Rightarrow \iota_2\rho/\rho'\nu$	–	–	x	x
Change (property)	–	$\iota\rho_1\nu \Rightarrow \iota\rho_2\nu$	–	$\iota\rho_1/\rho'\nu \Rightarrow \iota\rho_2/\rho'\nu$	–	–	x	x

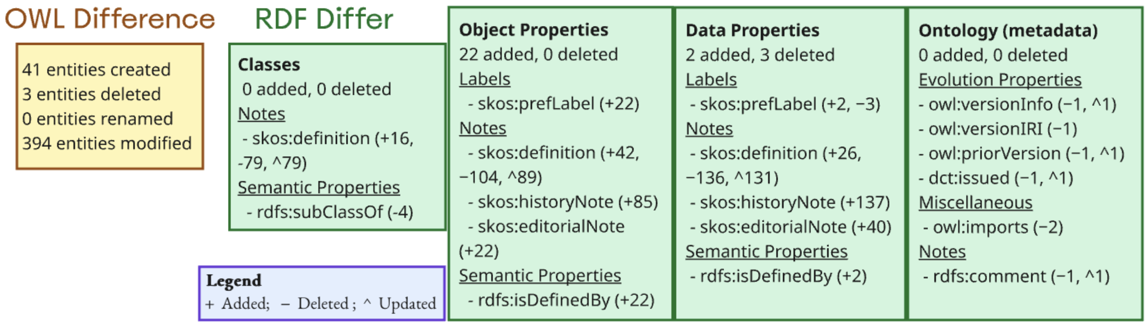

Figure 2: Comparison of RDF Differ and OWL Difference aggregate outputs on ePO v4.1.0 against v4.2.0.

The use case selected is the eProcurement Ontology (ePO)[2], an ontology being developed as part of the Digital Europe Project, and used in several projects concerning EU public procurement. It has multiple versions, extensive documentation, and a lightweight 'core' profile among its artefacts. They originate from a UML model converted by model2owl[3] into OWL (RDF files in Turtle format) of varying OWL expressivity, of which the *ePO-core* makes barely any use of OWL-specific constructs. To show the working of RDF Differ, we compare the last two stable minor versions of *ePO-core*, being v4.1.0 and v4.2.0, which have a sufficient changeset to compare. We also compare the output against the official changelog[4] and note evident differences. Based on our practical review (recall Table 1), we chose the Protégé plugin OWL Difference [4] as the best contender to compare with. We ran the aforementioned selected ePO versions, and compared the two tools on their usage, time, and output.

The results concerning the reporting of the changes at the level of the language features are shown in Fig. 2 and the qualitative comparison in Table 3. Both tools detected the same number of added object and data properties (24). OWL Difference provided only aggregated entity-level statistics, whereas RDF Differ offered more detailed insights at the property level and for metadata. For example, RDF Differ detected ontology 'header' changes, such as for `owl:versionInfo` that OWL Difference did not appear to track. Further, OWL Difference overreported (by 17) the number of entities as it double-counts data property updates. The lack of categorisation and ability to copy or locate results also makes it harder to audit changes in OWL Difference. Although RDF Differ is slow to produce results, this is offset by a much higher granularity of diffs and usability.

Comparing RDF Differ's output to the official ePO changelog, the latter underreported several changes;

[2]https://docs.ted.europa.eu/epo-home/index.html; last accessed 4-6-2025.
[3]https://github.com/OP-TED/model2owl; last accessed 4-6-2025
[4]https://docs.ted.europa.eu/EPO/4.2/release-notes.html#_epo_core; last accessed 4-6-2025.

Table 3
Summarised evaluation results. For the Reporting Granularity comparison, refer to Fig. 2.

Tool	Exportable Results	Diffing Time	Strengths	Limitations
OWL Difference	No, also cannot copy	0.032s	• Fast	• No grouping or categorisation • No copy, export, search or sort • No metadata changes
RDF Differ	Yes, in JSON & HTML	90s	• Clear grouping and categorisation • Export, search, and sort functionality	• Slow • No summary of modifications

e.g., it reports only 9 object properties added versus 22 detected by RDF Differ. The changelog also misclassified epo:isSMESuitable as newly added, even though it was present in previous versions. Conversely, the manual ePO changelog lists the cardinality constraints and domain/range for every change, whereas OWL Difference and RDF Differ do not report them at all.

4. Conclusions

We introduced RDF Differ, a semantic diff tool tailored to RDF vocabularies. By enabling profile-based configuration and semantic-aware queries, it fills a longstanding gap in RDF and lightweight ontology versioning. The demo will showcase RDF Differ's API, UI, and usage on real-world vocabularies.

Future work includes a more user-friendly interface for the dqgen component, additional predefined reporting templates, support for the OWL 2 EL profile, and possible performance optimisation via caching and parallelisation.

Declaration on Generative AI

During the preparation of this work, two of the four authors used Windows Copilot and Writefull in order to: Grammar and spelling check, Paraphrase and reword. After using these services, those authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] P. Kremen, M. Smíd, Z. Kouba, OWLDiff: A practical tool for comparison and merge of OWL ontologies, in: Proc. of DEXA'11, IEEE Computer Society, 2011, pp. 229–233. Toulouse, France, Aug. 29 - Sept. 2, 2011.

[2] N. Arndt, P. Naumann, N. Radtke, M. Martin, E. Marx, Decentralized collaborative knowledge management using git, Journal of Web Semantics 54 (2019) 29–47.

[3] J. Neubert, skos-history: Exploiting web standards for change tracking in knowledge organization systems, in: Proc. of WWW'16, WWW '16 Companion, 2016, p. 275–276.

[4] T. Redmond, J. Vendetti, M. Horridge, OWL Difference Plugin, 2015. URL: https://github.com/protegeproject/owl-diff, last accessed: 11-6-2025.

[5] R. S. Gonçalves, B. Parsia, U. Sattler, Ecco: A hybrid diff tool for OWL 2 ontologies, in: Proc. of OWLED'12, volume 849 of *CEUR-WS*, 2012. Heraklion, Crete, Greece, May 27-28, 2012.

[6] N. Arndt, N. Radtke, Quit diff: Calculating the delta between rdf datasets under version control, in: Proc. of SEMANTiCS'16, ACM, 2016, p. 185–188.

[7] Apache Jena, rdfdiff, 2025. URL: https://jena.apache.org/documentation/tools/#other-handy-command-line-tools, last accessed: 3-7-2025.

[8] J. Ahn, D.-H. Im, J.-H. Eom, et al., G-Diff: A Grouping Algorithm for RDF Change Detection on MapReduce, in: Semantic Technology, Springer, 2015, pp. 230–235.