Evaluating OpenSPG Engine for Network Operations

Beyza Yaman[1,*], Michael Mackey[2], Peter Cautley[2] and Declan O'Sullivan[1]

[1]ADAPT Centre, Trinity College Dublin, Dublin, Ireland
[2]Huawei Ireland Research Centre, Dublin, Ireland

Abstract

Telecommunication networks pose challenges for root cause analysis due to complexity, heterogeneous data, and high event volumes. Existing models like RDF and LPG provide only partial solutions when used alone. This paper explores the Semantic-enhanced Programmable Graph (SPG) approach via the OpenSPG engine, which combines property graph flexibility with RDF-inspired semantics. Applied to a telecom use case, OpenSPG shows advantages for event-centric modeling but remains limited in maturity, tooling, and semantic interoperability, highlighting directions for future semantic-enabled network management.

1. Introduction

In knowledge representation, two dominant paradigms have emerged: RDF [1] and LPG [2]. Both rely on nodes and edges but differ in modeling style, expressiveness, and applications [3]. RDF, a W3C standard, enables interoperability and reasoning via subject–predicate–object triples, while LPG offers flexible schema design with key-value properties on nodes and edges. RDF is a flexible model but query inefficiencies [4] limit its suitability for scalable graph analytics, whereas LPGs excel in tasks like subgraph matching and real-time querying but lack formal semantics [3]. Each approach has distinct strengths and trade-offs, and the choice between them often depends on the specific requirements of a project, such as the need for reasoning capabilities, the data complexity, or performance needs. Ontologists often prefer RDF, while engineers lean toward LPG [5]. To bridge these strengths, the Semantic-enhanced Programmable Knowledge Graph (SPG) has been proposed [6], combining LPG's structure with RDF-like semantic constraints to better capture causality, hierarchy, and sequence in event-centric graphs. This approach is focused on mostly enterprise causal knowledge graphs (events that lead to another) to prioritize representing logical relationships such as causality, conditionality, hierarchy, and sequence between events, making events the foundational units of such graphs [6].

These modeling paradigms are increasingly relevant for complex, data-rich domains such as modern telecom networks, which struggle with massive data volumes, inconsistent formats, and fragmented repositories. Such challenges hinder correlation, contextualization, and unified monitoring needed for automation. As event-driven systems, telecom networks generate continuous telemetry streams that demand scalable processing and semantic interpretation. The SPG approach, combining LPG efficiency with ontology-based semantics, is well-suited for representing such events and their relations. In this study, we assess its applicability in telecom through the OpenSPG engine.

2. Related Work

To combine the strengths of RDF and property graphs, several mapping approaches have been developed. G2GML [7] converts RDF into property graphs for advanced analytics, while ontology-based methods [8] enable bidirectional transformations to preserve semantics and scalability. Tools such as Gremlinator [9] and Amazon Neptune's OneGraph [5, 10] further bridge SPARQL and openCypher queries, enhancing

Posters, Demos, Workshops, and Tutorials, co-located with SEMANTiCS'25: International Conference on Semantic Systems, September 3–5, 2025, Vienna, Austria
beyza.yaman@adaptcentre.ie (B. Yaman); michael.mackey@huawei.ie (M. Mackey); peter.cautley@huawei.ie (P. Cautley); declan.osullivan@adaptcentre.ie (D. O'Sullivan)
0000-0002-0877-7063 (B. Yaman); 0000-0002-9421-8566 (D. O'Sullivan)

interoperability. A hybrid direction augments property graphs with semantic schemas, integrating LPG's simplicity with RDF's constraints [3, 6]. OpenSPG, the first open-source engine based on the Semantic-enhanced Programmable Graph (SPG) approach, follows this line by enriching property graphs with semantic capabilities for industrial use [11, 6]. In this work, we use OpenSPG to demonstrate a telecom root cause analysis use case, comparing RDF-based modeling and SPG in terms of semantics and applicability. Further details and code are available in our report and submitted ISWC Industry Track paper[1]. The paper and the report compare RDF-based modeling with the Semantic-enhanced Programmable Graph (SPG) approach, implemented via OpenSPG, to evaluate their trade-offs in semantics, and applicability for telecom network root cause analysis. However, in this paper, the OpenSPG engine was chosen to explore our use case. The work is practical and demonstrational, highlighting an industry use case rather than deep formal reasoning.

3. Background

This section outlines telecom domain challenges and introduces the OpenSPG engine as a hybrid solution for enterprise use cases.

3.1. Challenges for Network Operations Domain

There are several operational challenges faced by the network telecom systems[2]: Modern network operations face a multitude of data-related challenges that complicate effective monitoring, analysis, and automation. The overwhelming volume of data from various network planes (management, control, and data) combined with inconsistent storage formats and fragmented repositories, makes it difficult to analyse and extract actionable insights. Correlating this data across different sources, understanding service and customer impact, and preserving contextual information such as interface roles or link functions are particularly difficult. Additional complexities arise from varied data collection methods, organizational silos, and conflicting sources of truth (e.g., controllers vs. live network state). Altogether, these issues hinder the development of a unified, machine-readable, and semantically rich view of the network needed for autonomous decision-making and scalable operations.

Recent work has explored semantic solutions for network operations. Martínez *et al.* [12] proposed the YANG Server Ontology for modeling YANG-based servers and NETCONF interactions, with RML-based integration guidelines [13]. They also developed CANDIL [14], a federated data fabric unifying heterogeneous network data into a knowledge graph. Similarly, NORIA-O [15] models network infrastructures and events, though its broad scope challenges consistent multi-domain integration.

3.2. OpenSPG Engine

Released in late 2023, the OpenSPG engine provides an architecture for constructing domain knowledge graphs using a semantic schema, logical rules, and modular operators based on the SPG approach. It integrates LLMs for natural language processing and supports layered graph construction, reasoning, and querying. Data is stored in a graph database (default: Neo4j[3]), with schema definitions in MySQL; alignment between table columns and schema properties is essential for proper mapping. Column names must either be declared as schema properties or mapped to them programmatically. The OpenSPG-Schema defines semantic predicates to enrich LPGs with syntax and semantic constraints. As shown in Fig. 1, it distinguishes three core types: Entity types (circles in Fig. 1) represent system objects (e.g., network elements); Event types (diamonds in Fig. 1) model actions affecting entities over time (e.g., route withdrawal); and Concept types (squares in Fig. 1) classify entities and events (e.g., control plane taxonomy). Distinguishing entity and event types supports semantic graph design by modeling entities as core objects and events as their interactions or state changes over time.

[1]https://github.com/beyzayaman/openspg
[2]https://datatracker.ietf.org/doc/html/draft-mackey-nmop-kg-for-netops-02#name-data-overload-from-network-
[3]https://neo4j.com/labs/neosemantics/4.3/import/.

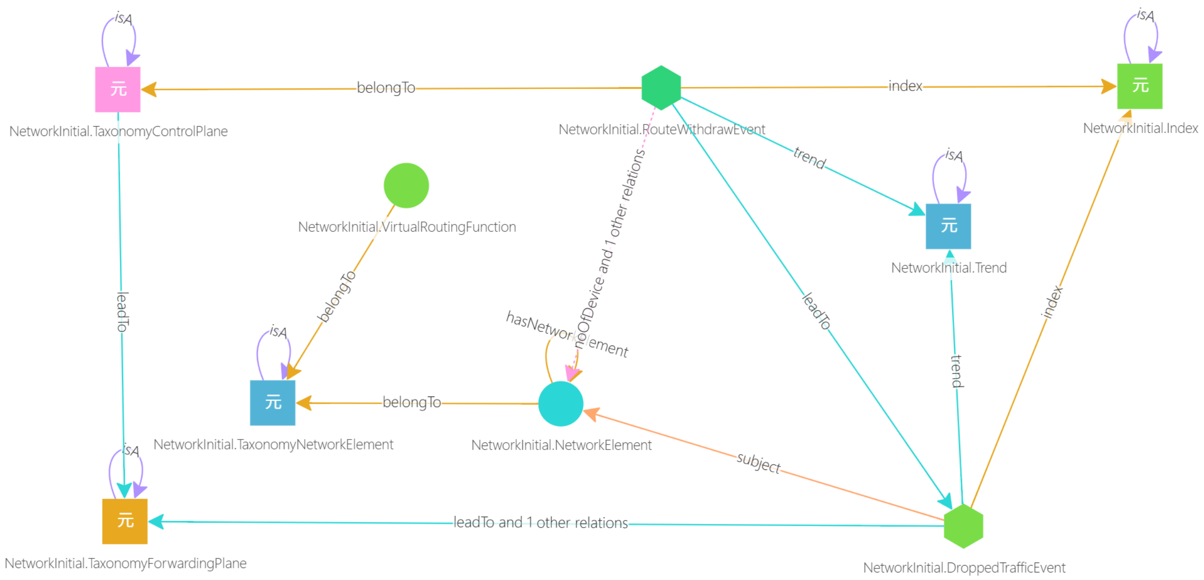

Figure 1: OpenSPG Network Schema Diagram (Circle: Entity; Diamond: Event; Square: Concept)

4. Application Scenario

This section presents a network systems use case implemented with OpenSPG, illustrating schema modeling for causal event relationships. In a network environment, each router first receives advertised routes, then decides whether to accept them based on rules (like who sent them), and may pass them on to other peers. These interactions help routers build paths to reach different networks, enabling internet traffic to flow efficiently. A Border Gateway Protocol (BGP) router may lose connection to one of its upstream peers, triggering a route withdrawal. This means it no longer advertises certain network prefixes it previously learned from that peer. As a result, neighbouring routers must quickly find alternative paths or may momentarily lose access to those networks. During this convergence period, packets destined for the withdrawn prefixes may be dropped, causing a temporary traffic outage or degradation in service until stable routes are re-established. Thus, it is imperative to identify the network elements which first causes the problem in the network.

Listing 1: OpenSPG schema for Network Operations

```
namespace NetworkInitial
TaxonomyNetworkElement(TaxonomyNetworkElement): ConceptType
        hypernymPredicate: isA
NetworkElement(NetworkElement): EntityType
        properties:
                neID(neID): Text
            index: TextAndVector
                neName(neName): Text
                hasNetworkElement(hasNetworkElement):NetworkElement
            constraint: MultiValue
                IND#belongTo(belongTo): TaxonomyNetworkElement
TaxonomyControlPlane(TaxonomControlPlane): ConceptType
        hypernymPredicate: isA
        relations:
                CAU#leadTo(leadTo): TaxonomyForwardingPlane
RouteWithdrawEvent(RouteWithdrawEvent): EventType
        properties:
                subject(subject): NetworkElement
                index(index): Index
                trend(trend): Trend
                time(time): STD.Date
                neID(neID): Text
                routeDistinguisher(routeDistinguisher): Text
                peerAddress(peerAddress): Text
                isAdjRIBin(isAdjRIBin): Text
                isAdjRIBout(isAdjRIBout): Text
                IND#belongTo(belongTo): TaxonomyControlPlane
        relations:
                CAU#leadTo(leadTo): DroppedTrafficEvent
```

Listing 1 presents a small script of the OpenSPG schema we created to represent relationships between different events. `RouteWithdrawEvent` is an event type which belongs to the event concept class `TaxonomyControlPlane` and it leads to the event type `DroppedTrafficEvent`. The subject of `RouteWithdrawEvent` is a network element. Designing a hybrid schema involves assigning labels to properties and defining schema constraints for classes. In the schema, it can be seen that NetworkElement has an index of on `neID` and `hasNetworkElement` property has a multivalue constraint.

Listing 2: Logical concept rule for Network Operations

```
‘TaxonomyControlPlane‘/‘RouteWithdraw‘:
    rule: [[
        Define (e:RouteWithdrawEvent)-[p:belongTo]->(o:‘TaxonomyControlPlane‘/‘RouteWithdraw‘) {
            Structure {
            }
            Constraint {
                R1: e.index == 'route'
                R2: e.trend == 'withdraw'
            }
        }
    ]]
```

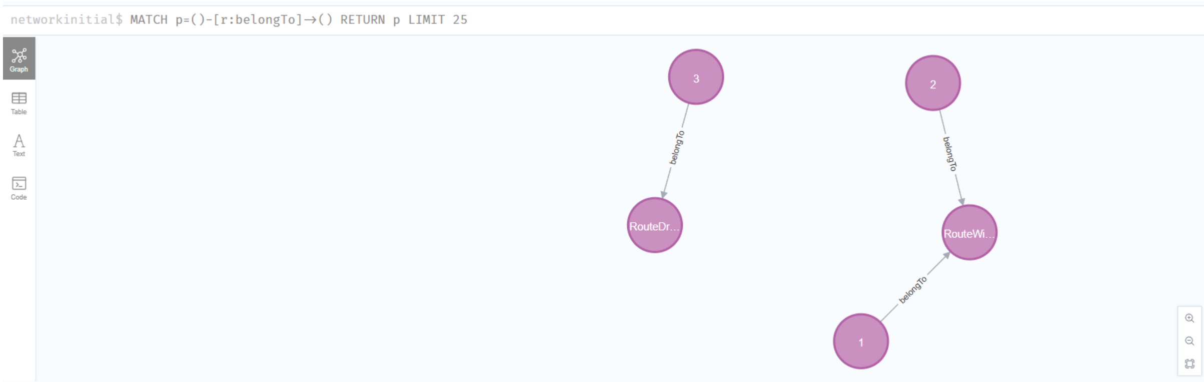

Figure 2: belongTo relationship

There are two ways of creating the relationships, one option is to introduce a new relation type in the schema which will require a manual upkeep of the schema and bring with it an associated risk of inconsistencies between the new data and the existing knowledge graph. A second approach is to create attributes and relationships through logical rules, with the goal of enabling more consistent and automated modeling. Listing 2 presents the rule for `belongTo` relationship where only index is route and trend is withdraw then the *instances* of `RouteWithdrawEvent` will belong to the concept of `RouteWithdraw` in the `TaxonomyControlPlane`. As a result of the rule given in Listing 2, we were able to classify different type of instances to their given classes dynamically. A Cypher query along with its corresponding result is shown in Fig. 2. For property graph management, OpenSPG relies on Neo4j as its underlying database.

5. Conclusions

This paper presents a work in progress investigation on the state-of-the-art methods, as part of their contribution to the ISWC industry track paper, reflecting both technical competence and a clear understanding of the underlying concepts in both OpenSPG and RDF approaches.

Switching from RDF-based modelling to the OpenSPG approach offers flexibility with schema definitions but comes with notable trade-offs in semantics and interoperability. Without RDF's formal ontologies and reasoning capabilities, automated inference and standards-based validation (e.g., SHACL) are limited, increasing reliance on manual interpretation. Interoperability also suffers, as integration with external linked data sources, reuse of public RDF datasets, and portability between systems become more complex due to the lack of shared vocabularies and standard query protocols like SPARQL. While these drawbacks may be less critical in performance-driven, internal domains such as telecom event analysis, they reduce the universality and cross-system compatibility that are hallmarks of semantic web solutions. Our use case illustrates how causal relationships between network events can be presented by employing SPG for modeling complex, event-driven data in network operations. This early exploration highlights the potential of SPG-based systems in operational domains and lays the groundwork for future research in semantic modeling for large-scale, event-driven environments.

Acknowledgement

This research received funding from Research Ireland [grant number 24/IRDIF/13114] and ADAPT Researchers are partially funded by the European Regional Development Fund through the ADAPT Centre for Digital Content Technology [grant number 13/RC/2106].

Declaration on Generative AI

During the preparation of this work, the authors used Chat-GPT-4 in order to: Grammar and spelling check. After using these tool(s)/service(s), the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] F. Manola, E. Miller, B. McBride, et al., Rdf primer, W3C recommendation 10 (2004) 6.
[2] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, A. Taylor, Cypher: An evolving query language for property graphs, in: Proceedings of the 2018 international conference on management of data, 2018, pp. 1433–1445.
[3] S. Purohit, N. Van, G. Chin, Semantic property graph for scalable knowledge graph analytics, in: 2021 IEEE International Conference on Big Data (Big Data), IEEE, 2021, pp. 2672–2677.
[4] Z. Zhao, X. Ge, Z. Shen, C. Hu, H. Wang, S2ctrans: Building a bridge from sparql to cypher, in: Int. Conference on Database and Expert Systems Applications, Springer, 2023, pp. 424–430.
[5] W. Broekema, M. Elzarei, O. Lassila, C. López-Enríquez, M. Neyman, F. Schmedding, M. Schmidt, A. Steigmiller, G. Varkey, G. T. Williams, A. Xiang, opencypher queries over combined RDF and LPG data in amazon neptune, in: Proceedings of the ISWC 2024 Posters, Demos and Industry Tracks, volume 3828, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3828/paper44.pdf.
[6] L. Liang, M. Sun, Z. Gui, Z. Zhu, Z. Jiang, L. Zhong, P. Zhao, Z. Bo, J. Yang, et al., Kag: Boosting llms in professional domains via knowledge augmented generation, arXiv:2409.13731 (2024).
[7] H. Chiba, R. Yamanaka, S. Matsumoto, G2gml: Graph to graph mapping language for bridging rdf and property graphs, in: International Semantic Web Conference, Springer, 2020, pp. 160–175.
[8] S. Das, J. Srinivasan, M. Perry, E. I. Chong, J. Banerjee, A tale of two graphs: Property graphs as rdf in oracle., in: EDBT, 2014, pp. 762–773.
[9] H. Thakkar, D. Punjani, J. Lehmann, S. Auer, Two for one: Querying property graph databases using sparql via gremlinator, in: Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems and Network Data Analytics, 2018, pp. 1–5.
[10] O. Lassila, M. Schmidt, O. Hartig, B. Bebee, D. Bechberger, W. Broekema, A. Khandelwal, K. Lawrence, C. M. Lopez Enriquez, R. Sharda, et al., The onegraph vision: Challenges of breaking the graph model lock-in, Semantic Web 14 (2022) 125–134.
[11] P. Yi, L. Liang, D. Zhang, Y. Chen, J. Zhu, X. Liu, K. Tang, J. Chen, H. Lin, L. Qiu, et al., Kgfabric: A scalable knowledge graph warehouse for enterprise data interconnection, Proceedings of the VLDB Endowment 17 (2024) 3841–3854.
[12] I. D. Martínez-Casanueva, L. Bellido, D. López, Declarative construction of knowledge graphs from netconf data sources, Semantic Web Journal, Under Review (2025).
[13] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, R. Van de Walle, Rml: A generic language for integrated rdf mappings of heterogeneous data., Ldow 1184 (2014).
[14] I. D. Martinez-Casanueva, L. Bellido, D. González-Sánchez, D. Lopez, Candil: A federated data fabric for network analytics, Future Generation Computer Systems 158 (2024) 98–109.

[15] L. Tailhardat, Y. Chabot, R. Troncy, Noria-o: an ontology for anomaly detection and incident management in ict systems, in: European Semantic Web Conference, Springer, 2024, pp. 21–39.