

**From:** Tobias Lindaaker tobias.lindaaker@neo4j.com  
**Subject:** Position Statement for W3C Workshop on Web Standardization for Graph Data, Tobias Lindaaker  
**Date:** 11 January 2019 at 12:05  
**To:** group-data-ws-pc@w3.org

---



## 1. About me

### 1.1 Background and Biography

Tobias is a systems and language architect at Neo4j, and has worked with Graph Databases at Neo4j for more than the past decade. He played and continues to have a central role in the design of the Property Graph Data Model (PGM); for instance, designing and introducing into the PGM the notion of labels. As the first employee at Neo4j, Tobias has been and continues to be a key figure in shaping the evolution of Neo4j. Tobias is one of the principal designers of the Cypher query language for property graphs and currently serves as Neo4j representative in ISO as well as the US standards body.

### 1.2 Goals for this workshop

My ambition for this workshop is to find alignment and synergies between the GQL effort in the area of Property Graph systems and the future directions of RDF. Personally I would like to learn and understand the reasons for particular designs and features rather than just the functionality. I am also keen on getting to know more people working in this space, forming healthy future working relationships.

### 1.3 Interests

My primary interest in Graph Data Management is from a systems perspective, building efficient and capable data management systems that solve real problems. This has lead me to be interested in the capabilities of data management languages as well as data representation formats for graphs. In this area path algebra is of particular interest to me, as well as the balance between efficient querying and computation in the database.

## 2. Proposed topics

### 2.1 Path Pattern Language

Through the years there have been multiple approaches to expressing path queries over graphs. It is useful to consider and compare these different approaches in order to learn from them going forward. A good start for understanding the evolution of path languages is G [1] and its successors G+ [2, 3] and GraphLog [4]. After that, Gram [5] and GraphDB [6] provide interesting extensions by allowing patterns to range over node types as well, and not only edge types. SPARQL 1.1 [7] added property paths for supporting this kind of CRPQs. Further extensions come from GXPath [8] which allows predicates on data values in addition to type tests.

DataLog is another important language where many advances have been made that are of significance for path querying. Similar contributions come from the general field of semi-structured data, with query languages such as UnQL [9] with its support for structural recursion.

I would also like to bring attention to the ongoing industry efforts in GQL and the property graph querying extension to SQL, inspired by Cypher [10] and PGQL [11].

### 2.2 Pattern Matching and Morphism

It is a proven fact that isomorphic matching of Regular Path Queries is intractable in the general case [12], yet experience shows that in many cases isomorphism is more intuitive for expressing real world queries. For many queries that have intractable worst case performance, it is often the case they perform perfectly fine on real world data. It has also been shown that the most commonly used path patterns in real query logs belong to a tractable subclass of Regular Path Queries [13]. This begs the question as to what kinds of morphism make sense for graph querying, and how to clearly and succinctly express the difference between different morphisms, and the scope of morphism (such as within a single pattern or across a whole query).

### 2.3 Additional Topics

In addition to the topics above that I am proposing to introduce and lead a conversation about, there are a few other topics that I think would be valuable to cover during the workshop. Some of these I know that my colleagues are proposing to introduce.

It would be a good idea to take a look at the landscape of existing graph query languages. Some of the people interested in GQL have come together into an "Existing Languages Working Group", their research would be a useful input into such a landscape survey discussion.

The topic of combining data from multiple graphs in one query, would be useful to cover. In particular comparing different approaches taken by different languages and systems.

Schema in Graphs is another topic that would be valuable. I have heard from several sources that RDF ontologies are too heavy and difficult to work with, while Property Graphs are too relaxed in their schema. Therefore it would be useful to discuss how these two schools of thought can be bridged.

The notion of supporting both sophisticated data retrieval as well as analytics in the same query language or query processing

system, and how to balance the tradeoffs between the two kinds of workload from a language design point of view is very interesting to me, and I would think a relevant topic for this workshop.

### 3. References

- [1] I. F. Cruz, A. O. Mendelzon, and P. T. Wood. "A graphical query language supporting recursion." In Proc. ACM SIGMOD, pages 323–330, 1987.
- [2] I. Cruz, A. Mendelzon, and P. Wood. G+: Recursive Queries without Recursion. In Proc. 2nd Expert Database Systems Conference, pages 645–666, 1989.
- [3] A. O. Mendelzon and P. T. Wood. Finding regular simple paths in graph databases. In Proc. of the 15th Int. Conf. on Very Large Data Bases, pages 185–193, 1989.
- [4] M. P. Consens and A. O. Mendelzon. Expressing structural hypertext queries in GraphLog. In Proc. 2nd Annual ACM Conference on Hypertext, pages 269–292, 1989.
- [5] B. Amann and M. Scholl. Gram: A graph data model and query languages. In Proc. ACM Conference on Hypertext, pages 201–211, 1992.
- [6] R. H. Güting. GraphDB: Modeling and querying graphs in databases. In Proc. 20th Int. Conf. on Very Large Data Bases, pages 297–308, 1994.
- [7] A. Seaborne and S. Harris(Editors). SPARQL 1.1 Query Language, W3C Recommendation 21 march 2013.
- [8] Libkin, L., Martens, W., and Vrgoč, D. Querying Graphs with Data. ACM Journal, pages 1-53, 2016.
- [9] P. Buneman, M. Fernandez, and D. Suciu. UnQL: a query language and algebra for semistructured data based on structural recursion. The VLDB Journal, 9(1):76–110, Mar. 2000.
- [10] openCypher. Cypher Query Language Reference, Version 9, Mar. 2018.  
<https://github.com/opencypher/openCypher/blob/master/docs/openCypher9.pdf>.
- [11] van Rest, Oskar, et al. "PGQL: a property graph query language." Proceedings of the Fourth International Workshop on Graph Data Management Experiences and Systems. ACM, 2016.
- [12] Mendelzon, Alberto O., and Peter T. Wood. "Finding regular simple paths in graph databases." SIAM Journal on Computing 24.6 (1995): 1235-1258.
- [13] Martens, Wim, and Trautner, Tina. "Enumeration problems for regular path queries." arXiv preprint arXiv:1710.02317 (2017).