

From: Joshua Shinavier joshsh@uber.com 
Subject: position statement for the W3C Workshop on Graph Data
Date: 22 January 2019 at 04:01
To: group-data-ws-pc@w3.org



First of all, thank you for inviting me to serve on the PC. However, I can not have been of much help since I am apparently not on the PC mailing list. If you add me, I try to make myself useful.

My background:

I am one of the three co-founders of TinkerPop, now Apache TinkerPop. Most developers do not know that early TinkerPop had its roots in the Semantic Web, and was not entirely focused on property graphs. Before Gremlin, we used an RDF-based language called Ripple [1, 2]. Before Neo4j, we used AllegroGraph, and one of the first things we did with Neo4j was to build an RDF adapter [3]. Almost by definition, that was the first PG-RDF mapping. We added several others in the following years: SailGraph [4], GraphSail [5], and PropertyGraphSail [6], each with distinct though related functions (mapping from PG to RDF, from RDF to PG, and RDF storage in a graph database, similar to the original Neo4j wrapper). We never formally described the data model nor the mappings, but they were simple and useful. With TinkerPop3, the focus shifted from model-level mappings to query-level ones, and RDF support was dropped. In addition to a SPARQL-like "match" step, Apache TinkerPop supports RDF at the query level by mapping SPARQL queries to Gremlin queries.

Much more recently, at Uber, the knowledge graph effort has had need of a schema framework for property graphs, as well as mappings to RDF. I described an early version of this framework in a talk at Graph Day Seattle 2017 [7]. The framework builds upon some of the common themes that have emerged from the disparate schema languages of Neo4j, JanusGraph, and other NoSQL graph- or graph-like backends, while also providing easy interoperability with OWL and an eventual path to integration with external vocabularies such as schema.org. As the original developer of the DCAT-inspired Dataset extension [8,9] to schema.org during my time as graduate student at RPI, I took inspiration from schema.org while designing a framework and a process for developing a core common vocabulary at Uber. This vocabulary is endorsed at the highest level of tech leadership here, officially for use in RPC services, but it is in fact also used for streaming and storage. In Q1 of this year, we will seek the same level of endorsement for the logical schema language itself, for which we will soon release some open-source tooling (I will also describe the framework in a public talk [10] later this week at Data Day Texas).

Goals for participation in the workshop

Uber has a business interest in bringing our internal standards into alignment with external standards, including emerging ones like the subject of this workshop. In the case of property graph - RDF - relational mappings, we also have techniques and experience to contribute in this area. Personally, I would like to make deep support for RDF a priority for TinkerPop4, as well. More than that, I would like us to have a solid foundation for defining and reasoning about transformations between data models, which I see in the language of category theory and the practices of functional programming.

As a cautionary tale, a previous attempt was made to standardize the property graph data model, in the form of the Property Graphs working group [11]. From what I gathered at the time, this never reached critical mass due to resistance from the graph database community, and despite pressure from the Semantic Web community. A property graph standard must be a pragmatic and flexible standard, allowing at least the degrees of freedom of TinkerPop's Graph.Features.

I look forward to contributing to this broad discussion, and to the shared understanding that will emerge.

Best regards,

Joshua Shinavier

[1] <https://pdfs.semanticscholar.org/c966/f8321060ef86af51827755377aa4887c5acb.pdf>

[2] <https://cdn.rawgit.com/joshsh/ripple/develop/doc/screencast/index.html>

[3] <https://github.com/neo4j-contrib/neo4j-rdf-sail>

[4] <https://github.com/tinkerpop/blueprints/wiki/sail-implementation>

[5] <https://github.com/tinkerpop/blueprints/wiki/sail-ouplementation>

[6] <https://github.com/tinkerpop/blueprints/wiki/PropertyGraphSail-Ouplementation>

[7] <https://drive.google.com/file/d/0B9GF5nhNxdyvc2ZMMkJSRF9cUlmeE04Qm1KX291ejl2V2pz/view>

[8] <https://www.w3.org/wiki/WebSchemas/Datasets>

[9] <https://developers.google.com/search/docs/data-types/dataset>

[10] <https://datadaytexas.com/2019/sessions#shinavier>

[11] <https://www.w3.org/community/propertygraphs/>

[12] <http://tinkerpop.apache.org/javadocs/3.1.3/core/org/apache/tinkerpop/gremlin/structure/Graph.Features.html>

