Graph schema validation at last? Revisiting the Stemmarest data model with Neo4J and SHACL

Andrews, Tara Lee
tara.andrews@univie.ac.at
University of Vienna, Austria

In 2020 there was an attempt to create an ontology (Andrews, 2020) for the variant graph, which is a data model used in tools such as CollateX (Haentjens Dekker et al., 2015), Stemmarest (Andrews, 2019), and TraVIZ (Jänicke et al., 2015) to represent and visualize text collations and certain sorts of annotation. The conclusion was that the ontology was not suitable for expressing the full data model employed in variant graphs, not least because all of them (Stemmarest most heavily) rely on property attributes, which RDF and OWL do not handle well and which can therefore not be suitably represented in an ontology definition.

Indeed, although ontologies have in the meantime been proposed for various aspects of digital scholarly edition (NIE-INE, 2019) and specifically for critical apparatus (Giovannetti, 2021) and an opportunity to discuss these together would be very welcome, one major motivation for undertaking the 2020 project at all was to address the concerns, frequently heard by the author in a very heavily XML-based community, that since graph databases do not tend to have schemas and certainly have no way to enforce these schemas, they lack the foundations for sustainability and interoperability that TEI XML schemas or, indeed, relational database schemas provide. Many of those who expressed concerns went on to point to ontologies as examples of schemas that can be used to validate RDF data.

What the 2020 project showed, in the end, was the inadequacy of ontologies and especially OWL as a validation model for graph-based data. While many of the classes used by Stemmarest and their relationships to each other can be described adequately in OWL, the model relies upon many additional constraints that, while seemingly simple, go beyond OWL's capabilities. For example, in the Stemmarest ontology we can say that a Witness can belong to a Tradition and also to one or more Stemmata, and that a Tradition can contain multiple Stemmata. However, one of the important aspects of validation is to prevent inattentive mistakes in the data model. For example, if a witness W appears in a stemma belonging to tradition A, the model software should ensure that W also belongs to tradition A as a witness and not, for example, an unrelated tradition B; moreover, W should not appear in any stemmata of tradition B. Even more intractable problems arise with more complex specifications: for example, the need to ensure that the variant graph is self-consistent, so that a user cannot create a situation in the model where a portion of the edited text loops back onto itself or is omitted altogether. Thus, while the Stemmarest ontology can describe the data reasonably well, it is not an adequate tool for setting constraints; these must still be described in code.

Vogeler (2021, p. 80) notes in his discussion of graph-based models for critical edition that "the formal affordance of OWL is not really exploited in Digital Scholarly Editions". We would argue here that most editors are less interested in the sort of open-world inference for which OWL was intended (though Vogeler makes a case for why it might be interesting), as opposed to the formalisation, description, and consistency of their data. In this sense, the critics of graph-based models do have a point; how do we validate our graphs? Constraint enforcement in Neo4J, perhaps the most popular graph database, is indeed limited strictly to data properties of nodes and relationships (Neo4J, Inc., 2022). The situation is hardly better for LOD stores, which have so far depended more or less entirely on ontology specifications, with limitations as described above, to describe their data.

Recently, however, there have been two intriguing developments in the Semantic Web world that are especially relevant to digital humanities projects, not least Stemmarest. The first of these is RDF*, an extension to RDF that allows an entire triple to be the subject or object in further triple statements (Arndt et al., 2021). While it is still in an early phase of development, RDF* provides the long-awaited ability to set context for LOD statements, and promises to triple stores the same ability to set properties on relationships that graph databases such as Neo4J have always provided; this in turn has opened the way for Stemmarest, and many other projects besides, to take advantage of Neo4J's powerful graph traversal and analysis algorithms while preserving full interoperability with LOD formats.

The second major development is SHACL (Shapes Constraint Language), a language whose purpose is to define a set of constraints against which RDF graphs can be validated (Knublauch and Kontokostas, 2017). This language solves exactly the problem that ontologies do not; specifically, in conjunction with a database that supports transactions and their rollbacks such as Neo4J, a set of SHACL constraints can ensure the consistency of data added to the graph, so that (to take the example above) a witness from one text tradition cannot be included in a stemma belonging to an entirely different tradition.

Perhaps the most intriguing feature of SHACL is the ability to include "constraint components" expressed in SPARQL or Javascript, providing more or less limitless possibilities for formalising complex models. In a sense, this affordance challenges the idea of the separation of program code and data that is the usual gold standard for sustainability efforts, insofar as constraints can be expressed as arbitrarily complex programming logic, embedded in RDF or Turtle files and saved as a schema alongside the data itself.

The SHACL model thus provides a fascinating way to bring the intellectual contributions inherent in codework (Zundert, 2016; Antonijevic Ubois, van Zundert, and Andrews, 2018) across the "sustainability divide" that allows them to be properly valued and conserved as scholarship; at the same time, it points to a promising future model of digital work in the humanities, in which domain-specific code can live alongside the data, to be processed by more generic (and thus, hopefully, more long-lived) software platforms.

Bibliography

Andrews, Tara L. (2019). Critical Edition as Process: A Digital Model. Presented at the European Society for Textual Scholarship, Málaga. https://stemmaweb.net/?p=74.

Andrews, Tara L. (2020). An Ontology for Critical Editions of Variant Text. In *Digital Humanities 2020: Book of Abstracts*. Presented at the Digital Humanities 2020, Ottawa. https://stemmaweb.net/?p=89.

Antonijevic Ubois, Smiljana / Zundert, Joris van / Andrews, Tara. (2018). Unwrapping Codework: Towards an Ethnography

of Coding in the Humanities. In *Digital Humanities 2018: Conference Abstracts*. https://dh2018.adho.org/en/unwrapping-code-work-towards-an-ethnography-of-coding-in-the-humanities/.

Arndt, Dörthe / **Broekstra, Jeen** / **DuCharme, Bob** / **Lassila, Ora** / **Patel-Schneider, Peter F.** / **Prud'hommeaux, Eric** / **Thibodeau Jr., Ted** / **Thompson, Bryan.** (2021, December 17). RDF-star and SPARQL-star. *W3C*. https://www.w3.org/2021/12/rdf-star.html (accessed 5 November 2022).

Haentjens Dekker, Ronald / **Van Hulle, Dirk** / **Middell, Gregor** / **Neyt, Vincent** / **Zundert, Joris van.** (2015). Computer-Supported Collation of Modern Manuscripts: CollateX and the Beckett Digital Manuscript Project. *Literary and Linguistic Computing*, 30(3), pp. 452–70. 10.1093/llc/fqu007.

Giovannetti, Francesca. (2021). The Critical Apparatus Ontology (CAO): Modelling the TEI Critical Apparatus as a Knowledge Graph. In Spadini, E., Tomasi, F., and Vogeler, G. (eds.), (Vol. 15). Norderstedt: BoD, pp. 125–39. http://www.uni-koeln.de/ (accessed 5 November 2022).

Jänicke, Stefan / **Geßner, Annette** / **Franzini, Greta** / **Terras, Melissa** / **Mahony, Simon** / **Scheuermann, Gerik.** (2015). TRAViz: A Visualization for Variant Graphs. *Digital Scholarship in the Humanities*, 30(suppl_1), pp. i83–99. 10.1093/llc/fqv049.

Knublauch, Holger / **Kontokostas, Dimitris.** (2017, July 20). Shapes Constraint Language (SHACL). *W3C*. https://www.w3.org/TR/shacl/ (accessed 5 November 2022).

Neo4J, Inc. (2022). Constraints - Neo4j Cypher Manual. *Neo4j Graph Data Platform*. https://neo4j.com/docs/cypher-manual/4.4/constraints/ (accessed 5 November 2022).

NIE-INE. (2019). Nationalen Infrastruktur Für Editionen - Infrastructure Nationale Pour Les Éditions. *Nationalen Infrastruktur Für Editionen*. https://e-editiones.ch/about (accessed 4 November 2022).

Vogeler, Georg. (2021). "Standing-off Trees and Graphs": On the Affordance of Technologies for the Assertive Edition. In Spadini, E., Tomasi, F., and Vogeler, G. (eds.), (Vol. 15). Norderstedt: BoD, pp. 73–94. http://www.uni-koeln.de/ (accessed 5 November 2022).

Zundert, Joris J. van. (2016). Author, Editor, Engineer — Code & the Rewriting of Authorship in Scholarly Editing. *Interdisciplinary Science Reviews*, *40*(4): 349–75. http://dx.doi.org/10.1080/03080188.2016.1165453.