Graph Database API for Variant Texts and Scholarly Editions

Tara L. Andrews and Sascha Kaufmann
University of Bern
firstname.lastname@dh.unibe.ch

The Digital Humanities group at the University of Bern currently hosts two projects of a rather different character, but relying on a common underlying model for the data. The first, Stemmaweb, is a collection of web services for scholars to investigate the transmission of complex text traditions. The second project, the SNF-funded "Chronicle of Matthew of Edessa Online" which runs from 2015–2018, aims to produce a technologically innovative digital critical edition of the twelfth-century Armenian chronicle written by Matt'eos Uṙhayec'i (Matthew of Edessa), a priest who lived in the Crusader county of Edessa in the early twelfth century (Uṙhayecʿi 1898).

The interesting challenge common to both these projects is to develop a platform for both philologists and historians that allows study of the text and its transmission, as well as annotation and presentation of the text in terms of its content (e.g. specific annotations for persons, locations and dates in the case of a historical text). One critical point is to conceive of the data model of a text separately from its presentation in any particular data format. Our texts are modelled internally in the form of a graph, in close affinity with previous work on this topic (e.g. Schmidt and Colomb 2009, Andrews and Macé 2013, Dekker et al. 2014) and the aim is to provide the ability through API functionality to generate a view of the data in a number of formats, including web page display or TEI export.

Having learned from our experiences in the Stemmaweb project[1], we store the texts and their annotations in Neo4J, a graph-database. This has several advantages:

1. The internal data structure is itself a graph structure.
2. Many essential traversal functions are already built into Neo4J, which saves coding time and confers a large performance advantage.
3. Various functions (e.g. plausibility checks for variant relationships) are easier to implement and more efficient.

This choice has required a migration of large parts of Stemmaweb's functionality, originally written in Perl, into Java. We have also adapted and improved the Web-API to enable users to access the service directly and more easily. This has the advantage of modernizing Stemmaweb's backend engine and making it more efficient, while the text model remains interoperable with 'the old' Stemmaweb.

In this paper we give a little insight into the Neo4J graph-database schema. We distinguish the database objects into 'System' and 'User' objects. While the platform's core, largely informed by the needs of Stemmaweb, is built by using 'System' objects only (e.g. TRADITION, SECTION, and READING nodes),

[1] Stemmaweb currently stores its information in a relational database as Perl object serializations.

'User'-objects are more flexible. They allow the user to define customized objects to store individual information, such as annotations, that are needed for a scholarly edition of a text.

Figure 1 illustrates a TRADITION, modelled as a node that consists of one or more SECTIONS. These SECTIONs should be interconnected by WITNESS_ORDER relationships to indicate their arrangement in individual text witnesses; they can also be interconnected via NEXT-relationships, to indicate a canonical sequential arrangement. Each SECTION-node is connected to a collation graph composed of READING nodes, which are also connected to each other to explicitly indicate the text sequence. These connections also come in two varieties: LEMMA_TEXT relationships are set by the editor and build the 'canonical'-text, while SEQUENCE-relationships indicate the text as it has been transcribed from the individual witnesses.

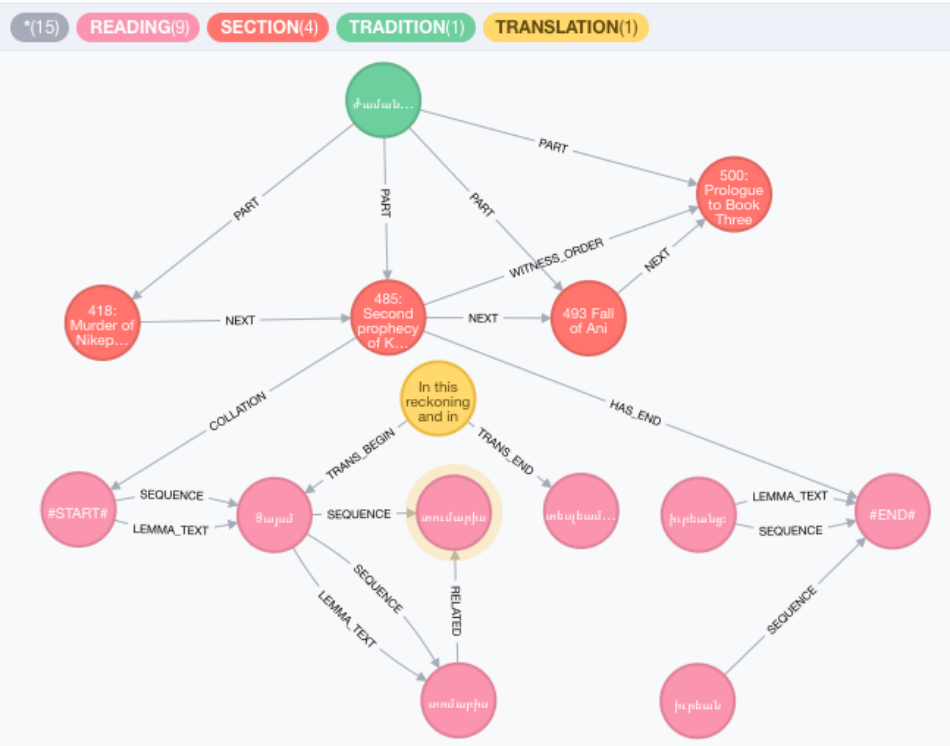

Figure 1: Database structure excerpt

This example also includes a TRANSLATION node that stores an editor's translation of a sequence of READING nodes. This is an example of a so-called 'User' object, which functions as an extension to the core created by the owner of the TRADITION. Thus the user is able to enter translations for words, sentences, paragraphs or whole sections; s/he is free to choose the granularity of any given translation. The definition of the 'core' system objects allows us to provide a REST API for common queries that takes advantage of the efficient traversal provided by the graph database, such as extracting the text of a particular witness in whole or in part; allowing recollation of the witnesses at a particular point in the graph; defining variation relationships (and thus more closely defining what a 'reading' is); extracting all annotations of a given class; and so on. Apart from these core text functions, we aim to make the API lightweight enough that a user may extract the data into any form required – for example an eBook, a web page for presentation, a TEI document hierarchy according to a defined schema, or an RDF document employing the CIDOC-CRM model.

References

Andrews, T. L. and Macé, C. (2013) 'Beyond the tree of texts: Building an empirical model of scribal variation through graph analysis of texts and stemmata.', *Literary and Linguistic Computing (10.1093/llc/fqt032).* 28.4, pp. 504–5021.

Dekker, R. H. et al. (2014) 'Computer- supported collation of modern manuscripts: CollateX and the Beckett Digital Manuscript Project.', *Literary and Linguistic Computing, fqu077. 10/1093/llcfqu007*.

Schmidt, D. and Colomb, R. (2009) 'A data structure for representing multi-version texts online.', *International Journal of Human-Computer Studies* 67, pp. 497–514.

Uŕhayecʿi, M. (1898) *The Chronicles of Matthew of Edessa*, Žamanakagrutʿiwn. Vałaršapat.