PHUSE US Connect 2023

Paper DV13

Navigating in an Information Graph

Eva Kelty, Capish, Malmö, Sweden

ABSTRACT

When a patient enrolled in a clinical study experiences a serious adverse event related to the study treatment it is important to find out more about the event as well as more about the patient. Also, have other patients experienced the same adverse event, in the same study or in other studies with the same treatment or with other treatments? Did the patients have other adverse events, treatments, or something else in common? How do you easily find all these answers quickly? Modern technology makes this possible. In the presentation we will discuss how to explore data by navigating an information graph and how complex questions can be answered instantly when working with a reflective logic implementation in a graph database.

INTRODUCTION

The R&D performance of pharmaceutical companies is dependent on a good understanding of all relevant data, not only for each study, but also for entire study programs. This sets requirements on *flexibility* and *scalability* of a database solution, to avoid the need to remodel the data repository as new data is to be incorporated [1]. Interoperability is an important component to achieve this e.g., by using a consistent approach to harmonization and integration of data, through standardization and implementation of common terminologies [2].

Furthermore, data needs to be modelled in an objective manner that maintains the *context* in which the data was collected. The context carries the meaning or the purpose of the data, which enables users to properly communicate, analyze, draw conclusions and make decisions [3].

NoSQL (not only SQL) solutions and especially graph databases have become more common when it comes to managing large amounts of data.

GRAPH DATABASE

Graph databases, as the name implies, represent data in a graph format, i.e., a network of information. The main difference to traditional relational databases is that each data point (node, vertex) is explicitly connected to other data points, via relationships (edges) [1]. How the nodes and their relations are modelled and stored depends very much on the purpose of the database.

A property graph database has a possibility to hold an internal structure of its components, achieved by adding properties, types and labels to the nodes and relations. How the nodes and their relationships are modelled in a property graph depends very much on the purpose of the database. While nodes and relationships are equal partners in a graph database, models can choose to focus on either nodes or relationships. The model to choose depends on the questions you want to ask and whether the aim is to analyze the graph as a whole for analytical purposes or whether local transactions are the primary concern.

ONTOLOGY

An ontology is a formal representation of a knowledge domain, describing its entities (e.g. events and processes) as well as the relationships connecting these entities. Ontologies are often used to add semantics to the data in a graph. Semantics in this context refer to the meaning and understanding of the data. No graph database can in itself provide semantics to its components. Instead, semantics must be provided explicitly, often by the use of an ontology that describes the components of the graph [1].

The Capish ontology consists of an information model, defining how the nodes and relations of the graph should be modelled, as well as a terminology, describing and defining all the components of the data. This model is based on a network of information units, so-called “Holons”, and their relations. Data that is tightly connected are stored together

in the Holon. Examples of such data could be variables belonging to an individual event, process, or entity, e.g., a measurement result stored together with its unit and the date it was measured.

KNOWLEDGE GRAPH/INFORMATION GRAPH

The populated Holons and the relations between them make up a knowledge/information graph that can be used for data exploration of the full database or of an individual patient.

In order to add even more value to the data, reflective logic is used to formulate queries with a centricity, called reflection point, and to find all information for Holons of that centricity that fulfil the query. In clinical trial data a typical reflection point is the patient, but the study or site could also be of interest. The result from a search with reflective logic will contain all Holons that are related to all Holons of the Holon type specified as the reflection point that match the query criteria [4].

Setting a reflection point in, for instance, the 'Patient' Holon will identify all data connected to a particular patient, in one single step. The possibility to add a reflection point makes it possible to explore the database from different perspectives and easily create data subsets.

CONCLUSION

Knowledge/Information graphs is a good way to work with life science data and provides great flexibility. With an interactive web-based interface to a graph database utilizing an ontology makes for a solution that is fit for FAIR principles; Findable, Accessible, Interoperable and Reusable data.

REFERENCES

References go at the end of your paper. This section is not required.

1. A Berg, H Drews and C Dahlbo, "PP05- New Approach to Graph Databases", PhUSE EU Connect 2018
2. A.E. Thessen, and D.J. Patterson, "Data issues in the life sciences". ZooKeys, 2011: p. 15-51.
3. I. Fleming and J. Leveille, *"TT10 – The Technology of Context"*, presented at the annual PhUSE conference, 2015.
4. C Dahlbo and E Kelty, "PP14 - A Two-Step Procedure – Select and Act – How Does It Work?", PhUSE EU CONNECT 2020

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the author at:
Eva Kelty
Capish Nordic AB
Lokgatan 8
211 20 Malmo, Sweden

Email:eva.kelty@capish.com