PhUSE EU Connect 2018

Paper PP04

Unique Technology for the Future

Catharina Dahlbo, Capish, Malmö, Sweden
Amanuel Workneh, Capish, Malmö, Sweden
Henrik Drews, Capish, Malmö, Sweden

ABSTRACT

Data curation is a very important step in the process to gain effective data explorations and visualizations. The curation step may include data quality controls, translations to a terminology and restructuring the data using an ontology-based approach with a graph database, focused on the nodes instead of relationships, as model. But, to take full advantage of the well-structured data, a new technology has been developed to support the information model. Curated data can be indexed in a way that allows complex queries from the user to be answered in a few seconds, without engaging experts.

This poster will present how a database engine with a method of querying a dataset with an appropriate index structure makes the search function and thereby analytical tools more powerful.

INTRODUCTION

As the amount of data increases, the importance of managing it in the best possible way to get the most out of the collected data also increases. The curation step has a profound impact on how data in the subsequent visualization step can be presented and thereby what knowledge can be gained. This curation step may include data quality controls, translation to a common terminology and restructuring the data according to an ontology-based information model. In conjunction with the right database and other tools that can utilize the features of the model, the full value of the data can be extracted. Preparing the data from disparate sources and domains for analysis given the prerequisites above gives the user flexibility when exploring the data.

Querying and analyzing datasets in existing solutions for data exploration and visualization are often limited on either flexibility or ease of use and are commonly also sensitive to structural changes of the datasets. By developing a new solution based on graph technology, reflective logic and innovative index structures, new opportunities are created in this area. The benefit of this approach does not lie in the individual steps alone but rather in the total concept. This creates possibilities to work with complex cohorts and find previously unknown relationships.

Currently, different data marts are built when data needs to be evaluated in a new context. In most cases this means that information is lost as not all data is integrated in the individual data marts due to the complexity of creating appropriate data models.

Depending on the questions, different vantage points might be necessary. In some cases, a patient perspective is required, in others a study perspective. Data should be accessible from different vantage points while maintaining the ability to connect to any other data point available. By utilizing a new way of searching in data it is possible to control from which vantage point the data should be evaluated, e.g. the patient perspective or the study perspective. A switch between different perspectives can be done by a click of a button in an application utilizing the new approach.

COMPONENTS FOR AN EFFECTIVE VISUALIZATION

In order to make an effective visualization, different useful components can be combined:

- Information Model - How the data is modelled
- Ontology - How the data in the model is described
- Graph Database - How the data is stored
- Reflective Logic - How the graph database is queried and searched

As discussed below, these components in combination create a powerful and flexible solution.

INFORMATION MODEL

The new approach is using an information model consisting of different objects, called Holons, built as a graph.

Each Holon

- contains information that is kept in different fields
- belongs to a Holon type (referred to as Holon definition in the ontology), which covers information of a single concept
- is related directly to at least one other Holon
- can be indirectly related to other Holons by following direct relations through Holons

When data is modelled in a graph with focus on nodes instead of relationships, new and powerful possibilities are created.

ONTOLOGY

An Ontology is holding the rules for the information model and describes the structure of the content. It defines how the Holons relate to each other, which fields are included in each Holon, the datatypes of the fields, units to be used, etcetera. The applied Ontology also provides a terminology controlling the naming of all the entities.

By utilizing an ontology, the data will be easier to access, easier to explore, easier to share, and above all, easier to understand.

GRAPH DATABASE

Since a graph database stores data as a network, where each data point is connected to other data points via relationships, processing of related data is handled in a more natural way. For example, compared to a relational database no joins are needed. By combining a graph database with a powerful conceptual ontology, we are able to create an information network that is very similar to the way the human brain organizes information. This framework also makes it possible to store any data about the integrated data as part of the model, thus combining metadata and data in a single system. A graph database is also less sensitive to structural changes [2].

Modelling the data according to a controlled ontology and storing the data in a graph database is a very good start but could it get even better?

REFLECTIVE LOGIC

In order to add even more value to the data a new method, reflective logic, has been developed [1]. Reflective logic is used to formulate queries with a centricity, called reflection point, and to find all information for Holons of that centricity that fulfill the query. The method consists of the following steps:

1. Create a subset of data
2. Find all Holons of a specific type that are included in or related to the subset of data
3. Create a new subset of data from the complete dataset that contains the Holons identified in the previous step and all Holons related to those Holons

The subset of data from step 1 above is said to be reflected in the reflection point, which means that data is shown from the perspective of the reflection point, during step 3. 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 [3].

THE TECHNOLOGY BEHIND THE REFLECTIVE LOGIC

On a technical level, an ontology allows the implementation of database indices that result in measurable performance advantages. Particularly the implementation of a relation index, enables the search using indirect relationships.

INDEX STRUCTURES

A couple of different bit vector index structures are created in order to handle both common questions and reflective logic. Each index vector consists of bit vectors where each bit may be either zero or one and each bit is mapped to a specific meaning. The length of the bit vector is equal to the total number of Holons.

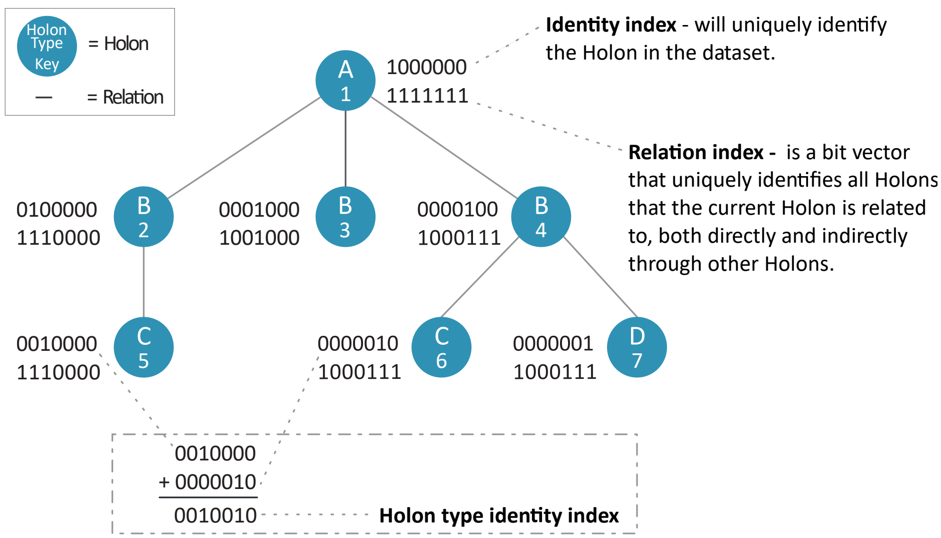

Figure 1: *The diagram shows an example of how the indices, used by the database, may look for each Holon in a dataset.*

There are mainly three kind of indices used:

Identity index - consists of index vectors that each identifies one Holon in the dataset

Relation index - consists of the combination of all identity index vectors of the Holons that the current Holon is related to

Holon type identity index - consists of one index vector for each type of Holon, where each index vector is created from the combination of the identity index vector of all Holons of that type

Using these index structures, it is possible to determine which objects of a specific type that are present in a subset of data.

USE OF INDICES

Questions often need to be asked against datasets in order to obtain knowledge from a dataset. The database engine utilizes the three index structures described to answer questions and process relationships between Holons. How the indices are used can be described as follows:

Identity index - is used to identify a Holon in the dataset

Relation index - is used to identify all data connected to e.g. a particular patient in a single step

Holon type identity index - is used to find which Holons of a specific type that are present in a subset of data

An example of a question could be: Consider a dataset of medical records where the height and weight of patients have been recorded. A physician might be interested in finding all information related to a low weight measurement in order to find out if any actions need to be taken.
This example is described and illustrated below. It shows how to find Holons that should be included in the result by combining relation indices.

1. In Figure 2, the low weight measurement is in the Holon of type D marked green with a dashed line. The Holon of type A marked green with a solid line represents the hospital in which the information was collected.

2. The two filtering criteria are defined to be combined using AND
The information should be collected at a certain hospital
The weight measurement should be considered low

3. By combining the relation indices of the two criteria in step 2 a new identity index is composed. This index identifies the Holons that either matches the criteria or is related, directly or indirectly, to one that does.

4. The Holons with the dotted line are the result of the query

Holon type specification
A = Hospital
B = Patient
C = Height
D = Weight

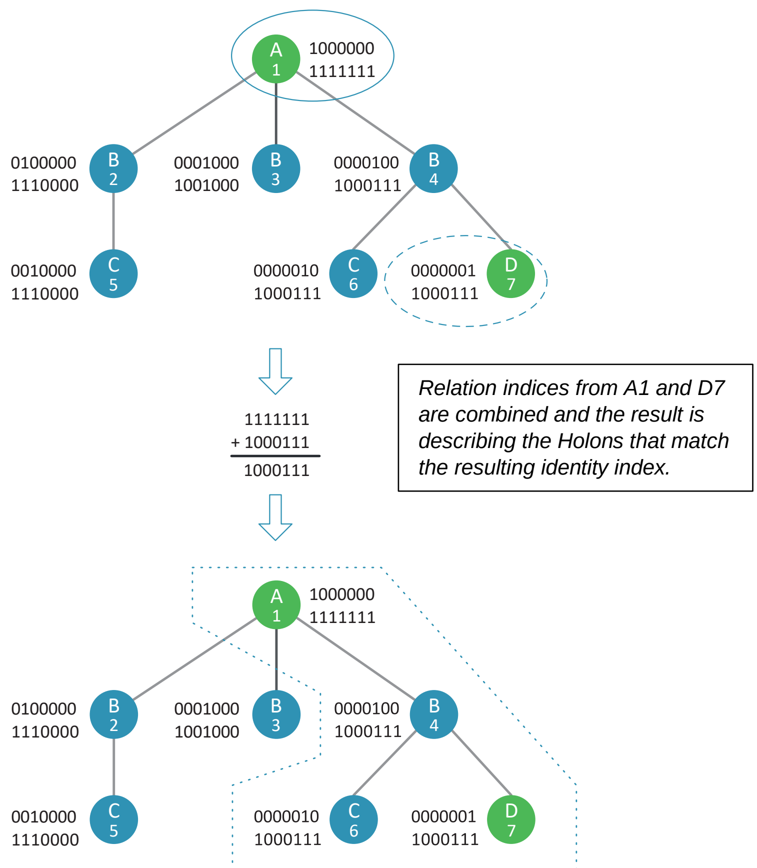

Figure 2: *The figure shows an example of how the database can utilize the relation index to give the physician all information, related to a low weight, that might be of interest.*

DESCRIPTION OF REFLECTION

By utilizing the power of the ontology-based information model, the resulting graph database can be searched in different ways and not only in the mentioned way. The illustration below shows the difference between searching with and without reflection.

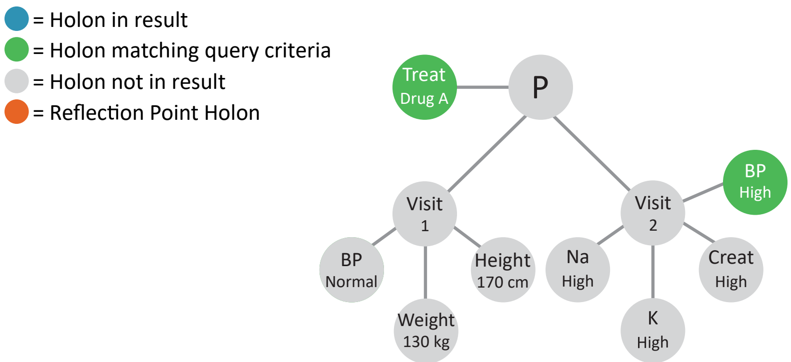

Figure 3: *The query criteria is set to high blood pressure and treatment with Drug A*

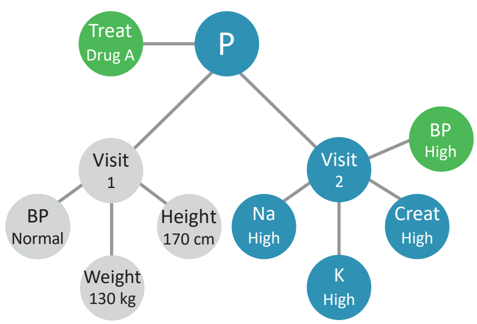

Figure 4: *No Reflection*

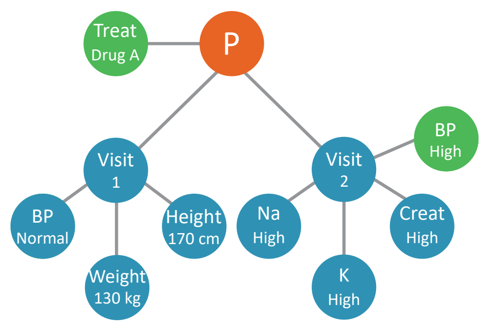

Figure 5: *Reflection*

In the examples above one patient is selected to illustrate the effect of different approaches. Figure 1 shows the query criteria containing a patient that has had high blood pressure and been treated with Drug A. The two other figures show the result sets for two different approaches that answers two different questions.

The first example, Figure 4, answers the question “What results are generated when a patient had high blood pressure and was treated with Drug A?”. The result set shows data related to the time (Visit 2) when the patient had high blood pressure.

The second example, Figure 5, answers the question “What results are connected to a patient with high blood pressure and has been treated with Drug A?” The result set shows here all events related to the patient independently of when they happened. This approach enables searching in the database in an easier, more flexible and responsive manner that give the user a broader result set. This example operates on both direct and indirect relationships in the datasets.

In the second example, Figure 5, patient is used as reflection point which means that data is shown from a patient perspective. All data connected to a patient, directly or indirectly, will then be captured. Other reflection points e.g. the study or the site can also be of interest in an analysis of a clinical study. This possibility to change reflection point adds flexibility and value to the dataset.

MORE USE CASES

A search for a value could potentially not only retrieve all the information units containing that value but could also be configured to retrieve all directly or indirectly related information.
Examples of other questions than mentioned above are:

>What other adverse events have been reported for patients with elevated liver values who received drug A?

>Which female patients that were given a dose of drug A have had high blood pressure measurements during episodes of severe headache?

CONCLUSION

One advantage of this new technology is for instance, that the user can specify what kind of information that is of interest and what criteria that information must fulfill, without knowing how the datasets are structured. Another advantage is that queries over complex data, that would normally require deep graph traversal, are quickly answered using the index structures. Yet another advantage is, by utilizing reflective logic, it is possible to control from which vantage point the data should be evaluated, e.g. patient perspective or study/site perspective.

The new paradigm on how to effectively query and retrieve the information from a dataset in a way that exploits the existence of direct and indirect relationships in the ontology, has been taken advantage of in a new data exploration tool, Reflect. The indexed datasets in Reflect allows the end-user to ask and retrieve answers to complex questions promptly and returns a broader result set depending on the query.

The advantage of this graph retrieval approach increases with the complexity of the data and the questions.

REFERENCES

1. S. Gestrelius and H. Drews, "*Patent US20130246438 – Reflective logic unlocks knowledge in datasets*", 2013
2. A. Berg, H. Drews and C.Dahlbo, "*PP05 – New Approach to Graph Databases*", Presented at PhUSE EU Connect 2018
3. C. Dahlbo, E. Kelty and M. Rubison, "*PP20 - Understanding the Power of Information Modeling for Efficient and Effective Visualization*", Presented at PhUSE US Connect, 2018

ACKNOWLEDGMENTS

I would like to thank all my colleagues at Capish for having come with valuable input in this work.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Catharina Dahlbo
Capish Nordic AB
Carlsgatan 3
211 20 MALMÖ, Sweden

+46 (0)40 10 88 80
catharina.dahlbo@capish.com
www.capish.com