Handling Multiple Foci in Graph Databases

Jaudete Daltio[1,2] and Claudia Bauzer Medeiros[1]

[1] Institute of Computing - UNICAMP, Campinas, SP, Brazil
[2] Brazilian Agricultural Research Corporation's - EMBRAPA, Brazil
{jaudete,cmbm}@ic.unicamp.com

Abstract. Scientific research has become data-intensive and data-dependent, with distributed, multidisciplinary, teams creating and sharing their findings. Graph databases are being increasingly considered as a computational means to loosely integrate such data, in particular when relationships among data and the data itself are at the same importance level. However, a problem to be faced in this context is that of multiple *foci* – where a *focus*, here, is a perspective on the data, for a particular research team and context. This paper describes a conceptual framework for the construction of arbitrary foci on graph databases, to help solve this problem. The framework, under construction, is illustrated using examples based on needs of teams involved in biodiversity research.

Keywords: eScience, Graph Database, Focus, Views

1 Introduction and Motivation

eScience, sometimes used as a synonym for data-intensive science [9], is characterized by joint research in computer science and other fields to support the whole research cycle – from data collection, mining, and visualization to data sharing. Biodiversity research – our target domain – is a good example of eScience. It is a multidisciplinary field that requires associating data about living beings and their habitats, constructing models to describe species' interactions and correlating different information sources. Such data includes information on environmental and ecological factors, as well as on species, and includes images, text, video and sound recordings [5], in multiple spatial and temporal scales.

Sharing and reuse of data are hampered by the heterogeneity of data and user requirements inherent to such domains. Each community applies different data extraction and processing methodologies and has distinct research perspectives and vocabularies. Several researchers have adopted graph representations (and graph database systems) as a computational means to deal with such integration challenges [11], especially in situations where relations among data and the data itself are at the same importance level [1].

However, graph database systems present limitations when it comes to creating and processing multiple perspectives of the underlying data. This paper presents our approach to these issues, which consists of a conceptual framework that allows experts to specify and construct arbitrary perspectives on top of

graph databases. This framework, under construction, takes advantage of some of our previous implementation work, in particular concerning ontology management [6]. Informally, the idea is to support a notion similar to that of database views, constructed on top of graph databases. However, our constructs go beyond standard database views.

Here, we follow the terminology we introduced in [13], and use the term *focus* for such views. Intuitively, a *focus* is a perspective of study of a given problem, where data can be restricted to one specific scale/representation, or put together objects from distinct scales. Moreover, given the same set of data, distinct foci will arise when the data is analyzed under different models, processed using focus-specific algorithms, or even visualized with particular means.

This paper has two main contributions. The first is to explore the notion of views on graph database systems, which is not yet supported in such systems. This requires extending the traditional specification of views, while at the same time maintaining the same principles. The second contribution is to show, via the running example, how to model and create multiple foci, for biodiversity research, thereby allowing experts to manage and analyze the same underlying datasets under arbitrary perspectives.

2 Theoretical Foundations and Related Work

2.1 Graph Databases

Graph databases allow to represent information about the connectivity of unstructured data – a recurrent scenario in scientific research. The interpretation of scientific data usually requires the understanding about linked data, interactions with other data and topological properties about data organization.

The formal foundation of all graph data structures is based on the mathematical definition of graphs and, on top of this basic layer, several graph data structures were proposed [1,12], including features such as directed or undirected edges, labeled or unlabeled edges and hypernodes. One of the most popular structures supported by many graph database systems is the *property graph*. It tries to arrange all the features that these graph types express in a single and flexible structure through key-value pairs to describe vertex and edge characteristics, such as type, label or direction.

To manipulate these data, graph query languages can be used to [14]: (i) find vertices that satisfy a pattern; (ii) find pairs (x, y) of vertices such that there is a path from x to y whose sequence of edge labels matches some pattern; (iii) express relations among paths; (iv) compute aggregate functions based on graph properties; and (v) create new elements. Each query language has its own syntax and considers its own data structure to represent a graph.

2.2 Views

In the context of relational databases, a *view* can be regarded as a temporary relation against which database requests may be issued [7]. Views are widely used to restrict, protect or reorganize relational data. Views are built by a com-

bination of operations applied on the underlying relations, creating alternative or composite representations of existing database objects. The sequence of operations that creates a particular view is called *view generating function*.

The concept of view is used in many data management contexts. A *view of an ontology* is a subset of the original ontology, built by the extraction of some relevant parts thereof. Tools and languages for ontologies usually take advantage of their graph structure; vertices represent classes and instances and edges represent properties, relations and class hierarchies. There are different approaches to create ontology views [10]. Some are based on query languages and others are based on guidelines to navigate through ontology concepts, using the notion of *central concept* – a class around which the view is built and that defines which elements must be part of a view. Different from databases in which a query always results in an instance set, a query on an ontology can result in a partial schema (classes, relations), an instance set or a combination of both [6].

2.3 Multifocus Research

The notion of *focus* (a perspective of study of a given problem) appears naturally in eScience. The idea behind a focus is similar to the idea of an application – each application has its own perception of the world, goal, complexity and specific requirements. For the same underlying datasets, each focus represents a perception of the data, how it can be analyzed, visualized and interpreted.

A focus allows to restrict data, manage spatial and temporal scales thereof (multiple representations) and create distinct scenarios, including the vocabulary, constraints, process and rules that should be applied to the dataset [13,15]. The same data item can be interpreted in distinct ways – a species observation, for example, could represent an organism to be analyzed in a small level of detail or, in a macro perspective, a feature of a biome.

One important problem in focus-related research is how to improve data semantics, increasing its understanding and removing ambiguity. The use of ontologies has been pointed out as a means to deal with some of these issues and used to drive data management. This notion, known as *"ontology-driven information systems"* [8], uses ontologies as a central role with impact on the main components of the system and providing multiple perspectives of the data.

3 A Framework to Generate Foci

The goal of our research is to specify and implement a framework to build and explore arbitrary foci. To achieve this purpose, we extend the traditional definition of views to represent a focus, providing a reorganization of the original data or part thereof. The framework uses graph databases as the basis of data management, taking advantage of their ability to deal with highly connected datasets, a common scenario in eScience. Since graph databases do not implement the view concept, the framework introduces extensions to existing systems.

Figure 1 gives a general overview of the framework. The interface receives a focus specification as input and provides the focus as output. Both focus and

underlying databases are represented as graphs (a focus may be built combining one or more graphs). The focus specification is a text file whose content and format are still under definition, using existing graph query languages (e.g. Cypher, SPARQL [12]) and the parameters of graph algorithms. Following the figure, step (1) decomposes the focus specification to define the focus generation strategies, operators and parameters. Next, the focus is created using either a query view mechanism (2); a central concept view mechanism (3); or a combination of both.

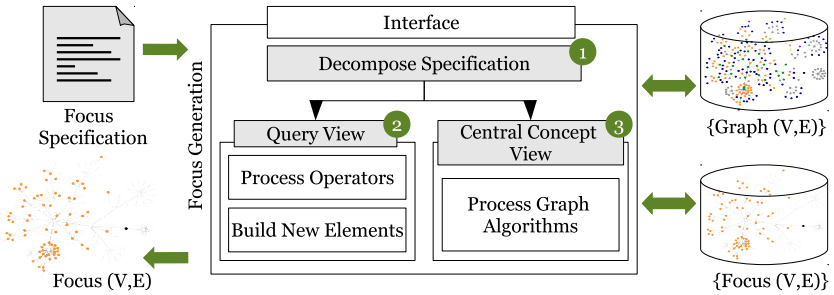

Fig. 1. Overview of the Focus Generation Process

The query view approach (2) adopts concepts from relational databases. Here we have two tasks: processing the operators that compose the query and creating new elements that do not belong to the original graph. Part of the focus specification is used to create the *"view generator function"*, the sequence of operators to be applied to the database. The traditional operators are adapted by the framework: (i) selection: to filter parts of the graph applying predicates; (ii) projection: to restrict parts of the original graph; (iii) join: to combine two or more graph databases via join conditions; (iv) aggregate functions: to provide graph summarizations, extracting vertex and edge properties.

The central concept view approach (3) is inspired by approaches to construct views on ontologies. Here, just one task is executed: processing of graph algorithms, starting from a central concept, namely a vertex defined in the focus specification. This graph algorithm can provide, for instance, the neighborhood, the shortest path to another vertex, the maximum clique, and so on [3]. The combination of these approaches allows expressiveness higher than graph query languages alone, usually untyped [4], based on triple patterns [12] and without native graph algorithms. Besides that, graph languages have limitations to create temporary elements without altering the original database and the result of a query is not necessarily a graph.

Graph databases and the foci created on the top of them are stored in a persistence layer, so that a focus can be reused. Moreover, since a focus is represented as a sub-graph, it can be used to construct other foci. We also keep the specification that originates a focus for provenance information – e.g., to

describe the perspective materialized in the focus and to allow to update a focus when the graph databases used to generate it are updated.

4 Running Example

Our running example concerns biodiversity studies of animal species, concentrating on observation metadata. In particular, we deal with observations of animal vocalizations, motivated by the challenges faced by the Fonoteca Neotropical Jacques Vielliard (FNJV) at the University of Campinas (UNICAMP) [3]. FNJV has a large collection of animal sound recordings (about 30 thousand observations), whose metadata is stored in a relational database [5]. Observation metadata include information about the species, the place where the sound was recorded, the recording devices, date and time of the observation, and so on.

Although the metadata is, currently, structured as a relational database, it can be directly converted to a property graph database [12], applying straight formal approaches, e.g. [2,11]. Each row of each table can be modeled as a vertex, using the column names as attributes, and each foreign key can be modeled as an edge. Altogether, an observation has 54 metadata attributes, which can be combined in different ways to determine the edges of the graph database. Figure 2 shows one possible graph database denoted by G_{obs}. In the figure, vertices 1 through 6 represent the taxonomic hierarchy of the observed species, and vertices 8 through 11 characterize an observation, represented by vertex 7.

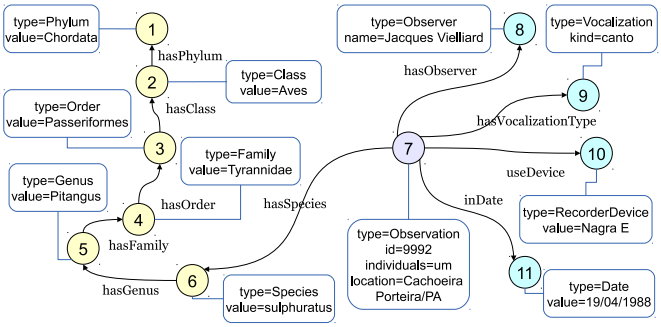

Fig. 2. Partial Metadata Graph Database of FNJV Observations - G_{obs}

G_{obs} can be integrated with many additional information sources, such as biological and environmental variables to describe the context in which vocalizations were recorded. Distinct pieces of information can be used to produce specific analyses and to build foci. A focus may concern, for example, a geographical scale or a group of species of interest. The following examples describe some use scenarios of foci for this graph database.

[3] http://proj.lis.ic.unicamp.br/fnjv

4.1 Example Focus 1: Location and Biomes

An example of focus which changes the perspective of analysis is defined as: *"Set of all locations in which observations were made, summarizing the number of distinct species observed at each location, and connecting the locations that belong to the same biome"*. This kind of focus can be helpful to analyze the biological and environmental characteristics of locations that were targets of study. To process this focus, it is necessary to aggregate the observation data to generate new information (here, the number of distinct species) and to link the original data with biome information (graph external to our database).

Let us first consider just the first part of the focus: *"Set of all locations in which observations were made, summarizing the number of distinct species observed at each location"*. This kind of focus can also be processed by the query view approach (2) of the framework, combining: (i) "build new element" operator, to create the set of vertices with type **Location** from the attribute *location* of vertices of type **Observation** in G_{bio}; (ii) "aggregate function" operator, to count the number of distinct species observed in each **Location** and store the value in *numberOfSpecies* attribute; (iii) "projection" operator, to filter the vertex and edge types that should be part of the focus (in this case, **Location**).

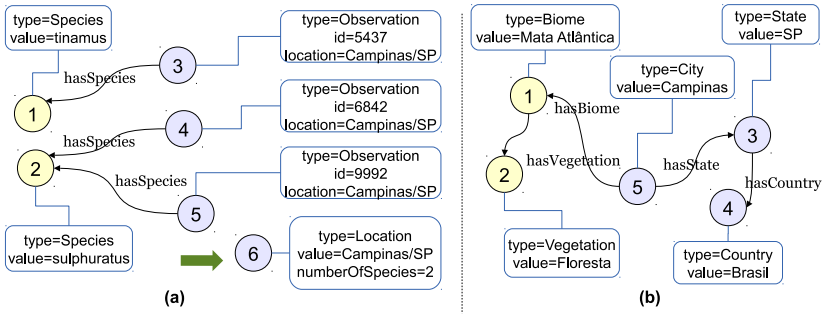

Fig. 3. Focus: (a) location and number of distinct species and (b) Partial Biome Graph Database - G_{bio}

Figure 3 (a) presents a portion of G_{obs} and explains these steps, with the creation of vertex 6 **Campinas SP** of type **Location** and *numberOfSpecies* (here, set to value 2). To connect the locations of the same biome, it is necessary to add biome information not available in G_{obs}. Figure 3 (b) shows a partial biome graph database (here shortened to G_{bio}), which is used to integrate this information, using the join operator. In this case, the focus specification combines: (i) "join" operator, to link each vertex with type **Location** in G_{obs} with the corresponding vertex of type **Biome** in G_{bio}, creating an edge (**hasBiome**) between **Location** and **Biome**; (ii) "build new element" operator, to create the set of edges with type **sameBiome** beetween the **Locations** connected to the same **Biome**; (iii) "projection" operator, to filter the vertex and edge types that

should be part of the focus (vertices of types **Location** and **Biome**). A partial view of the result focus is shown in Figure 4 (a).

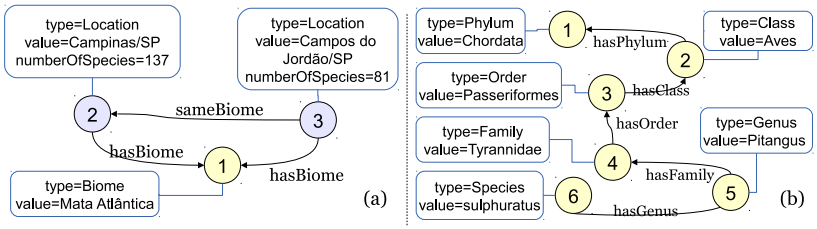

Fig. 4. (a) Query View Focus: Observation Locations and Biomes (b) Central Concept Focus: species closest to **Tinamus tao**

4.2 Example Focus 2: Species "Closely Related" to Tinamus tao

Another possible scenario builds the focus from a central concept. Here, an example would be: *"Which are the species closest in the taxonomy to the species **Tinamus tao**"*. This kind of focus can be helpful to analyze the diversity of the species observed according to the "closeness" to other species within a taxonomic level (e.g. genus, family or order). This focus can be processed by the central concept view approach (3) of the framework, starting from species *Tinamus tao* in G_{obs}. The graph for this focus is built considering only edges related with taxonomic classification levels. The notion of closeness here is defined considering the distance between the vertices in G_{obs}: closest mean shortest paths.

The generating function combines: (i) "projection" operator, to filter from G_{obs} the set of vertex and edge types that should be part of the focus (in this case, vertex types related to taxonomic level); (ii) "central concept", in this case, the vertex of type **Species** that represents the species *Tinamus tao*; (iii) the graph algorithm to be applied, in this case, shortest path. The focus result contains all species vertices in the graph for which the paths to species *Tinamus tao* are minimal. A partial result focus is shown in Figure 4 (b).

This focus can be further restricted to *"Species closest in taxonomy to **Tinamus tao**, observed in the same locations"*. This can be helpful to understand the similarity among environments where "closely related" species are observed. In this case, specification of focus 2 should be extended, including a "selection" operator to filter only species observed in the same locations. This focus demands a combination of all functionalities available in the focus generation module.

5 Conclusions and Ongoing Work

This paper presented the specification of a framework to to build and explore arbitrary foci in scientific databases, using graph databases as the basis of data management. The approach extends the traditional definition of views in relational databases to represent a focus, combining graph query languages with

graph algorithms to build customized foci. The internals of the framework were explained via examples in biodiversity data management, pointing out some of challenges to be faced. The implementation of the framework will take advantage of previous work of ours in ontology management [6].

The first challenge involves extending the concept of view of relational databases to graph databases. Another challenge is related to the specification of a focus. At the moment, we assume that a focus is specified by indicating a suite of operations to be applied to the underling graph databases. This, however, will need to be improved once we formalize focus construction operators.

Acknowledgements Work partially financed by FAPESP/Cepid in Computational Engineering and Sciences, MSR FAPESP Virtual Institute (NavScales), CNPq (MuZOO), FAPESP-PRONEX (eScience), and grants from CNPq.

References

1. R. Angles and C. Gutierrez. Survey of graph database models. *ACM Comput. Surv.*, 40(1):1:1–1:39, February 2008.
2. C. Bizer. D2rq - treating non-rdf databases as virtual rdf graphs. In *In Proceedings of the 3rd International Semantic Web Conference*, 2004.
3. U. Brandes and T. Erlebach. *Network Analysis: Methodological Foundations (LNCS)*. Springer-Verlag New York, Inc., Secaucus, USA, 2005.
4. D. Colazzo and C. Sartiani. Typing query languages for data graphs. *I. W. on Graph Data Management: Techniques and Applications*, 2014.
5. D. C. Cugler, C. B. Medeiros, and L. F. Toledo. An architecture for retrieval of animal sound recordings based on context variables. *Concurrency and Computation: Practice and Experience*, pages 1–17, 2011.
6. J. Daltio and C. B. Medeiros. Aondê: An ontology web service for interoperability across biodiversity applications. *Information Systems*, 33(7-8):724–753, 2008.
7. A. Furtado, K. Sevcik, and C. Santos. Permitting updates through views of databases. *Informations Systems*, 4:269–283, 1979.
8. N. Guarino. Formal ontology and information systems. In *Proceedings of Formal Ontology in Information System*, pages 3–15. IOS Press, 1998.
9. T. Hey, S. Tansley, and K. Tolle, editors. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, Redmond, Washington, 2009.
10. N. F. Noy and M. A. Musen. Specifying ontology views by traversal. In *International Semantic Web Conference*, volume 3298 of *LNCS*, pages 713–725, 2004.
11. Y. Park, M. Shankar, B. Park, and J. Ghosh. Graph databases for large-scale healthcare systems: A framework for efficient data management and data services. *I. W. on Graph Data Management: Techniques and Applications*, 2014.
12. I. Robinson, J. Webber, and E. Eifrem. *Graph Databases*. O'Reilly Media, Incorporated, 2013.
13. A. Santanche, C. B. Medeiros, J. Jomier, and M. Zam. Challenges of the Anthropocene epoch - Supporting multi-focus research. In *Proc XIII GeoINFO*, 2012.
14. P. T. Wood. Query languages for graph databases. *SIGMOD Rec.*, 41(1):50–60, April 2012.
15. S. Zhou and C. B. Jones. A multi-representation spatial data model. In *I. S. on Advances in Spatial and Temporal DBs*, volume 2750 of *LNCS*, pages 394–411, 2003.