Report from Dagstuhl Seminar 22031

Bringing Graph Databases and Network Visualization Together

Karsten Klein*[1], Juan F. Sequeda*[2], Hsiang-Yun Wu*[3], and Da Yan*[4]

1 **Universität Konstanz, DE.** karsten.klein@uni-konstanz.de
2 **data.world – Austin, US.** juan@data.world
3 **FH – St. Pölten, AT.** hsiang-yun.wu@fhstp.ac.at
4 **The University of Alabama – Birmingham, US.** yanda@uab.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 22031 "Bringing Graph Databases and Network Visualization Together". Due to the ongoing restrictions caused by the COVID-19 pandemic, this purely on-site seminar had a reduced number of participants. Twenty-two researchers and practitioners from the Network Visualization and Graph Database communities met to initiate collaborative work and exchange between the two communities. The seminar served to establish a common understanding of the state of the art and the terminology in both communities, and to connect participants to tackle joint research challenges. Survey talks on the first days laid the foundations for subsequent plenary discussions and working groups. Further lightining talks during the next days gave more detailed insight into specific research questions and practical challenges. The contributions of the seminar include bringing the communities together, the identification of the top areas of research interest, and the characterization of research challenges and research questions. As an outcome, a position paper is planned, and further collaborations and joint publications are on the way.

Seminar January 16–21, 2022 – http://www.dagstuhl.de/22031
2012 ACM Subject Classification Information systems → Data management systems; Human-centered computing; Human-centered computing → Visualization
Keywords and phrases graph databases, network visualization, visual analytics
Digital Object Identifier 10.4230/DagRep.12.1.67

1 Executive Summary

Karsten Klein (Universität Konstanz, DE)
Juan F. Sequeda (data.world – Austin, US)
Hsiang-Yun Wu (FH – St. Pölten, AT)
Da Yan (The University of Alabama – Birmingham, US)

Network analytics through interactive network visualization has been essential in many research and application areas, such as bioinformatics, biomedicine, cyber security, e-commerce, social science, and software engineering. A network is often supported by graph databases with advanced query engines and indexing techniques. Graph databases have substantial contributions by academia and gained strong momentum in the industry, where the focus is on scalable systems using graph query languages that require to be learned by users.

* Editor / Organizer

Even though the Graph Database and Network Visualization communities study the same object, a graph/network, albeit from different perspectives, they do not communicate with each other. By bringing both communities together, we aimed to initiate and foster mutual communication and joint work. The goal of this seminar was to initiate collaborative efforts, to increase the mutual awareness of each others' existing concepts and technologies, and to identify new and complementary research challenges that lead to novel scientific outcomes. We have developed the schedule for the seminar based on our experience from previous successful Dagstuhl Seminars with a balance between prepared talks, plenary discussions, and breakout groups for less structured discussions focused on a selection of highly relevant topics.

The organizers envisioned several core topics for discussion at the Dagstuhl Seminar, as outlined in the proposal:

- Integration of fundamental concepts used in the two communities
- Visual scalability and computational performance
- Visual graph query paradigm
- Responsive visualization of graph query results
- (Qualitative) Evaluation
- Domain-oriented applications

During the plenary discussions on the first day, the participants identified several more specific topics for the work in separate working groups, which lead to the following working group titles:

- Evaluation and Usefulness
- Understanding gaps and opportunities between Graph Databases and Network Visualization
- Visual querying and result visualization

Our aim was to have focused discussions on these topics in which we would be able to make significant progress during the seminar, in order to shape a position paper and to lay the foundations for subsequent collaborations. The discussions showed that there indeed is the need for a closer exchange between the communities in order to improve the mutual understanding and practical solutions, but also to identify research questions that can be tackled jointly. They however also showed the great potential in this exchange and the large interest in both communities for joint work.

We have organized the seminar during the COVID-19 pandemic. Due to various regulations and travel restrictions, only roughly half of the usual number of participants attended in person. The meeting was held purely on-site, with the exception of one participant connecting in via video conferencing. We thank Dagstuhl for equipping the seminar rooms with suitable infrastructure and for putting suitable health and safety regulations in place to create a smooth experience and safe environment for all participants.

Acknowledgments

We would like to thank all participants of the seminar for their contributions and lively discussions, and the scientific directorate of Dagstuhl for providing us with the opportunity to organize this seminar. Finally, the seminar would not have been possible without the amazing support by the staff of Dagstuhl.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

3 Overview of Talks

3.1 Lightning Talk on GraphPolaris Visual Graph Analytics System and Research Challenges

Michael Behrisch (Utrecht University, NL & GraphPolaris.com Analytics Platform)

GraphPolaris is a no-code analytics platform for graph analysis. It enables non-data scientists to analyze large and complex datasets without the typically required query scripting and allows exposing the gathered analytical insights directly through effective visualizations. Our inductive exploration workflow engages the user into a visual data analysis dialog, in which an intuitive drag and drop user interface guides the construction of complex analytical queries visually and expressive visualizations give access to on-the-fly result interpretations, thus making graph databases and graph analytics accessible for a wide commercial and research audience.

In this talk, I demonstrated, on the one hand, the current approach towards developing a visual graph analytics platform in GraphPolaris and, on the other hand, contrasted it with the current research challenges.

3.2 Visual Graph Query and Analysis for Tax Evasion Discovery

Walter Didimo (University of Perugia, IT)

We briefly report on a 3-years collaboration with the Italian Revenue Agency, about the design of a decision support system for tax evasion discovery. The system combines network visualization techniques with graph databases and exploits some key ingredients: 1) An intuitive and powerful visual language that allows analysts to define suspicious patterns related to fraudulent schemes; the language also handles temporal information and does not require the knowledge of native GDBMS query languages. 2) A graph pattern matching engine, built on top of the Neo4J graph database, which efficiently retrieves and ranks subgraphs from a large network of taxpayers, based on the previously defined schemes. 3) An interactive environment which makes it possible to visualize the results returned by the graph pattern matching engine and to incrementally explore the network of taxpayers starting from them. Additionally, the system adopts artificial intelligence and information diffusion techniques to automatically assign each taxpayer a risk level based on both classical network centrality indices and on new centrality indices that estimate the level of involvement of a taxpayer in suspicious activities.

3.3 Overview of Graph Query Language Part 1 – Foundations

George Fletcher (TU Eindhoven, NL)

We gave an overview of graph data models with particular attention to RDF and Property Graphs. We then highlighted two basic ingredients used in the design of graph query languages: subgraph pattern matching and path querying. In the first class the languages of conjunctive queries and unions of conjunctive queries were formally defined and illustrated with examples. In the second class reachability, label constrained reachability, and regular path queries were defined and illustrated. We then defined and illustrated languages which combine both ingredients: unions of conjunctions of regular path queries and the regular queries. We concluded with an overview of the complexity of query evaluation and the complexity of query containment for each language.

3.4 Lightning Talk on Visualization and Query Optimization

Pavel Klinov (Stardog – Arlington, US)

The idea of the talk is that there might be an overlap between the graph visualization area and the graph query optimization area. Both graph visualization tools and query optimizers often require graph pre-processing, such as summarization, algorithms to figure out the structural properties of the graph. Visual rendering algorithms require it to highlight the most salient nodes and patterns in the graph while the optimizers use it to enable cardinality estimations for generating efficient query plans. It remains to be seen whether similar summarization or sampling algorithms can turn out useful for both kinds of tools.

3.5 Visualizing large graphs: a brief overview and some research experience

Fabrizio Montecchiani (University of Perugia, IT)

Visualization is a very popular and central task in graph processing pipelines. When the input is a large and complex network, computing an effective visualization is very challenging. The main steps involved in the creation of large-scale graph visualizations are usually simplification, layout, rendering, and interaction. We gave a brief overview of the methods and techniques used to address each of these steps. Particular attention has been paid to node-link layouts and to force-directed methods for computing such layouts. We also presented a vertex-centric multilevel force-directed algorithm to compute node-link layouts on a cloud computing platform.

3.6 Lightning Talk: Case Study: yFiles layout improvements for Graph Database Visualizations

Sebastian Müller (yWorks GmbH – Tübingen, DE)

The presentation was about what improvements were made in the graph drawing library "yFiles" in response to requirements by real world users working with graph database visualizations. With the help of conceptually simple layout techniques, the usefulness of visualizations for graph database exploration purposes was improved dramatically. By incorporating information about different semantic types of nodes and edges in the diagram as stored in the graph database, the algorithms were able to improve the arrangement in the diagram. Also, as an additional improvement, certain patterns that frequently appear in query results were specifically highlighted in the diagram, matching the users' expectation of the query results. For this, paths, stars, chains, and parallel structures were detected and treated especially by the layout algorithm. These improvements are available for various common layout algorithm implementations in yFiles.

3.7 What do Knowledge Graphs, Data Catalogs and Network Visualization have to do with each other?

Juan F. Sequeda (data.world – Austin, US)

Data catalogs are metadata and data management systems that inventory and organize data within an organization. Knowledge Graphs are a means of integrating data and knowledge at scale where concepts and relationships of a domain are manifested in the form of a graph. Thus a data catalog can be powered by a knowledge graph.

A question is how can Network Visualizations help accomplish common tasks in a data catalog such as business glossary definition, ontology engineering, impact analysis, root cause analysis, sensitive data impact. In this presentation, I highlight challenges and opportunities when attempting to integrate Network Visualization with a data catalog powered by a knowledge graph.

3.8 Lightning Talk on Neo4j Bloom

Hannes Voigt (Neo4j – Leipzig, DE)

We gave a brief ad-hoc presentation/demo of Neo4j Bloom, a domain-agnostic, low-code, ad-hoc graph visualization and exploration tool for data experts, data scientists, and data analysts.

3.9 Overview of Graph Query Language Part 2 — Practice

Hannes Voigt (Neo4j – Leipzig, DE)

We gave an overview of three graph query languages used in practice: SPARQL, Cypher, and GQL. The talk illustrated how the basic ingredients for graph query language discussed in Part 1 manifest in these graph query languages. This demonstrated that these languages – even if designed for different data models – are very similar in their capabilities. Still, we pointed out corners in which the languages differ slightly in their capabilities or put different emphases. Specifically for property graph query languages, we highlighted how they make use of the visual benefits of the ascii-art approach used in the graph pattern sublanguage and how it contributes to the adoption of these languages.

3.10 Qualitative Evaluation: Opportunities and Pitfalls

Tatiana von Landesberger (Universität Köln, DE)

This talk introduces evaluation studies in graph visualization. It focuses on qualitative evaluation with users. Evaluation studies with users are important when evaluating real value of application-motivated visualizations such as medicine, biology, finance. The visualizations need to be of high quality both from perceptual side and need to fit to the user's task and experience. Based on author's experience, and related literature in conducting evaluations of visualizations for real applications, the talk presents main steps, guidelines and pitfalls in conducting the studies. The talk discusses how the choice of tasks influence the evaluation, the pros/cons of methodological choices such as think-aloud protocols, which measures should be used for the evaluation, what is the value of free feedback for the evaluation. Finally, the talk presents how pitfalls from ill-posed evaluation questions can be turned into interesting research questions.

References

1 Archambault, Daniel, Helen Purchase, and Tobias Hoßfeld. "Evaluation in the Crowd." Crowdsourcing and Human-Centered Experiments: Dagstuhl Seminar 15481. Vol. 10264. Springer, 2015.

2 Purchase, Helen C. Experimental human-computer interaction: a practical guide with visual examples. Cambridge University Press, 2012.

3 Isenberg, Tobias, et al. "A systematic review on the practice of evaluating visualization." IEEE Transactions on Visualization and Computer Graphics 19.12 (2013): 2818-2827.

4 Sedlmair, Michael, Miriah Meyer, and Tamara Munzner. "Design study methodology: Reflections from the trenches and the stacks." IEEE transactions on visualization and computer graphics 18.12 (2012): 2431-2440.

5 Ballweg, Kathrin, et al. "Visual Similarity Perception of Directed Acyclic Graphs: A Study on Influencing Factors and Similarity Judgment Strategies." J. Graph Algorithms Appl. 22.3 (2018): 519-553.

6 Kochtchi, Artjom, T. von Landesberger, and Chris Biemann. "Networks of Names: Visual Exploration and Semi-Automatic Tagging of Social Networks from Newspaper Articles." Computer Graphics Forum. Vol. 33. No. 3. 2014.

3.11 An Overview of Graph Analytics

Da Yan (The University of Alabama – Birmingham, US)

Main reference Da Yan, Guimu Guo, Md Mashiur Rahman Chowdhury, M. Tamer Özsu, Wei-Shinn Ku, John C. S. Lui: "G-thinker: A Distributed Framework for Mining Subgraphs in a Big Graph", in Proc. of the 36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020, pp. 1369–1380, IEEE, 2020.
URL https://doi.org/10.1109/ICDE48307.2020.00122

A typical data analytics workflow includes (1) data retrieval, which falls in the domain of Data Engineering for data maintenance and querying; and (2) data analytics, which falls in the domain of Data Analytics. This is of no exception when we consider graph data, where the relevant graph data can be first retrieved from a graph database using query languages such as SPARQL, Neo4j Cypher, and GQL; the obtained graph can then be analyzed using methods in data mining and knowledge discovery, machine learning, and data visualization. Oftentimes, such retrieval and analytics algorithms can be programmed using a graph-parallel programming framework where users only need to specify some important user-defined functions based on application needs, rather than learning and using existing query languages and analytics libraries.

This talk introduces two such programming framework paradigms: (1) think like a vertex (TLAV) which aims to output a value for each vertex, as presented by Google's Pregel; and (2) think like a task (TLAT) for mining subgraphs, as presented by G-thinker, which allows compute-intensive analytics to scale performance with the number of CPU cores in contrast to existing data-intensive analytics tools. This talk also reviews a list of graph analytics tasks: (T1) Graph Traversal for Node Labeling, (T2) Random Walks for Node Scoring/Embedding, (T3) Graph Neural Networks, (T4) Frequent Subgraph Mining, (T5) Dense Subgraph Mining, and (T6) Subgraph Matching. Among them, (T1)-(T3) can be addressed by TLAV systems, while (T4)-(T6) can be efficiently addressed by TLAT systems such as G-thinker and PrefixFPM both published in ICDE 2020.

4 Working groups

4.1 Working Group 1: Evaluation and Usefulness

Juan F. Sequeda (data.world – Austin, US), Walter Didimo (University of Perugia, IT), Nadezhda T. Doncheva (University of Copenhagen, DK), George Fletcher (TU Eindhoven, NL), Stephen G. Kobourov (University of Arizona – Tucson, US), Giuseppe Liotta (University of Perugia, IT), Catia Pesquita (University of Lisbon, PT), and Tatiana von Landesberger (Universität Köln, DE)

This group discussed evaluation and the notion of usefulness of network visualizations. We observe a socio-technical phenomenon when it comes to the wow factor vs usefulness of network visualizations. Our discussions led us to the following realizations:

- We need to understand the diversity of roles around the work of network data management and network visualization.

- Bridging the gap between network visualization and graph database communities should be a win-win for both communities. Network visualization should support graph databases. Graph databases should support network visualization
- It is important to study how much network visualization can help in each role.
- There are specific interactions between the roles, which are not fully understood. How can network visualization help for those interactions?
- The network visualization space needs to be more fully studied, in the light of evaluation and usefulness.

The main topics we discussed were the following:

What is special about graphs/networks? Graphs are "natural" as they often capture the underlying model/problem well: If you need to model objects and relationships between them. A graph captures this with vertices and edges. Furthermore, a node-link diagram is a "natural" representation of a graph because the objects are the nodes, the relationships are the edges/links. There are many variations (edges can be directed/undirected, nodes can have attributes). The takeaway is that the data model, the visualization, and the interaction and meta-modeling paradigms are all the same: a graph. This is what makes graphs/networks special.

On the varieties of "Usefulness". What does it mean for a network visualization to be useful? Let's start with an example for the use case of ontology matching in life science. In the absence of context, we might infer that two references to the term "Gum" might match with high confidence. However, given the context of a dental ontology (network) visualization, we might note that one reference is in the context of the human mouth while the other is in the context of a type of candy, arising during different tasks by different researchers. This greatly lowers the subject matter expert's confidence in the match. This simple example illustrates that the usefulness of visualizations is very multifaceted.

We discussed and itemized several of these facets: the phases of usefulness and the lifecycle of network visualization (from engagement to final outcome); each of access for experts versus non-experts; aesthetics and enjoyability of network visualizations; trustworthiness and interpretability; small versus big data; loose versus highly structured data; data versus metadata; human context and human factors (e.g., profession); cost of interaction.

Roles in graph data and visualization. The final major topic discussed was that of roles around network data work. We identified some major roles, as a starting point to studying their interaction and how these interactions can be better supported by network visualization solutions.

- Domain Practitioner: Has expertise in a specific domain and has a question that needs to be answered. This role is the ultimate motivator (has the $), e.g., Doctor, CEO, Journalist.
- Analyst: Translates the question from the domain to the other roles. Probably the role responsible to provide the answer (or some means to get the answer such as data, visualization, etc) to the Domain Practitioner, e.g., Business Analyst, Data Scientist.
- Knowledge Scientist: Gather knowledge from the domain in order to understand what data should be used in order to answer the question.
- Visualization Scientist: Knows the space of visualization paradigms/metaphors and is able to suggest the best way of visualizing a certain type of data.
- Visualization Engineer: Is in charge of integrating or implementing visualization solutions and algorithms.

- Graph Database (GDB) Admin: Is responsible for the graph database, making sure the infrastructure is up and running, and provides access to the data.
- GDB Engineer: Is responsible to develop the graph database system itself; this role is typically filled in the context of a graph database vendor.
- Data Engineer: Is responsible for bringing in the data, integrating the data following requirements and loading them into a graph database.
- Data Steward: Is responsible for input data sources.
- Data Governance: Is responsible for understanding what data exists, how it is used and that it satisfies organizational requirements (regulations, policies, security, legal)

Some remarks. First, this list is not exhaustive. Second, one person can assume multiple roles, and multiple people can assume one role. Finally, we do not use the word "user" as it is ambiguous/confusing and also has negative connotations in some domains.

The following two additional roles are considered, but they are separated from the others, because they are studying the phenomena that occurs with the previous roles (their are not part of the evaluation process):

- Visualization Researcher: Provides innovation in the visualization field, by studying and experimenting new visualization metaphors/paradigms, layout algorithms, proof-of-concepts, prototype systems, quantitative and qualitative evaluations
- GDB Researcher: Provides innovation in the graph database field, by studying and experimenting new languages, data structures, efficient algorithms for graph queries.

These roles are fulfilled by professors, PhD students, industry researchers, etc.

Challenges and Opportunities. A major topic for further study is to deeply understand the interactions between roles and how we can better support these interactions with network visualization and graph data management solutions. A further general challenge is to study the different roles of visualization in the context of graph databases:

- Use visualization as a communication media between the different actors of the data production chain.
- Use visualization to design user-centric applications for the domain practitioners who need to explore the data and elaborate new information.
- Graph Layout Recommendation based on role, task, data, etc.

Within each of these challenges lies the deeper study of the definition(s) of usefulness, how we might quantify these definitions, and use these metrics for better evaluation of more effective network visualization and graph data management methods.

4.2 Working Group 2: Understanding Gaps and Opportunities Between Graph Databases and Network Visualisation

Karsten Klein (Universität Konstanz, DE), Henry Ehlers (TU Wien, AT), Oliver Kohlbacher (Universität Tübingen, DE), Sebastian Müller (yWorks GmbH – Tübingen, DE), Falk Schreiber (Universität Konstanz, DE), Hannes Voigt (Neo4j – Leipzig, DE), and Markus Wallinger (TU Wien, AT)

Originally constituted as a group to work on use cases and applications, our initial discussions quickly showed that before we could talk about concrete use cases, we needed to lay foundations for our common understanding of the potential interplay between graph databases

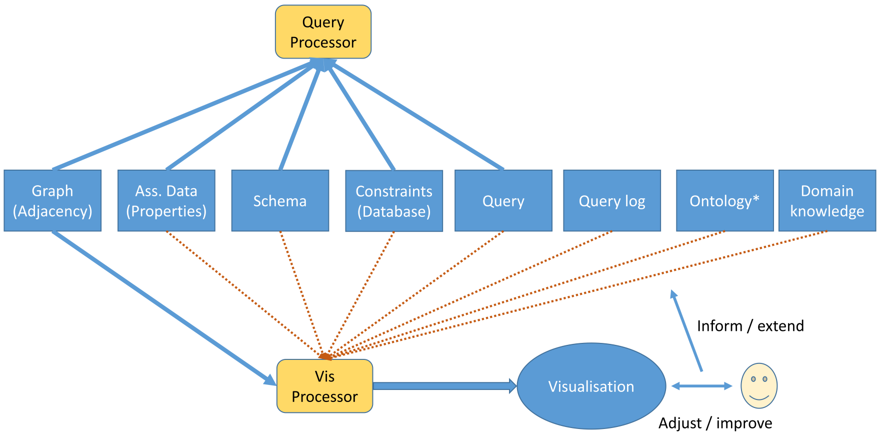


Figure 1 A conceptual diagram of the interplay between data assets, processing entities, and users in interactive graph database visualisation. It shows which sources of information could be used for query and visualisation processor, and how the user would interact with the system. Red arrows indicate currently unused but available information.

and network visualisation. While large potential for synergy and cross-pollination between the two areas is quite evident, we began by structuring the involved aspects and expectations in order to derive a conceptual framework based on which we could better identify gaps and opportunities of this interplay.

Our discussions quickly converged to a network visualisation in which we tried to cover the interplay between data assets and involved stakeholders and entities when graph visualisation is used in interactive interfaces for graph databases. Figure 1 shows the intermediate result that we came up with and which we used for the further discussions. While probably omitting some relevant aspects, it helped us to structure the discussion and to create a common mental map of the interactive graph database visualisation process.

With this conceptual model, we could now analyze the role as well as the aims of the user. The conceptual model then reveals requirements as well as the potential impact of graph visualisations, which in turn led us to a first characterisation of challenges and opportunities. As main gaps and challenges we identified

- A lack of appropriate graph visualisation and navigation methods that are tailored towards users of graph databases
- A lack of methods for projection in graph database systems (i.e., mapping from data assets to graph visualisation), incl. aggregation / abstraction of the graph structure
- Missing integration of concepts from graph database and visualisation communities into one coherent concept
- A definition of "usefulness" for graph visualisation in the graph database context and measures to evaluate it, and in general evaluation of metaphors for specific data/use cases/tasks
- Accessibility of meta-data available in the database system for the visualisation tool, and a systematic understanding of visualisation patterns for meta-data integration
- A lack of annotations for useful domain knowledge about graph topology in database schemas

As opportunities we see both significant potential improvements in practice, in particular regarding data handling, user experience, and more direct interfaces, as well as space for methodological work and new applications for research. These opportunities might concern different stakeholders differently depending on their role – user, developer, product owner, vendor – and area – visualisation, graph databases:

- Development of new graph visualisation and interaction techniques tailored towards graph databases. These can target the layout, encoding, and navigation, but also abstractions, e.g. for overview visualisation and subgraph comparison.
- Availability of currently untapped data sources and use cases for research, as there are huge and diverse graph databases with context available.
- Users can start with an improved out-of-the-box visualisation and apply ready-made templates to common use-cases with subsequent incremental improvement.
- New queries can be automatically derived by interacting with the visualisation as a more intuitive interface than current solutions.
- Query result visualisation has a large potential for improvement (leveraging meta-data and additional data stored with and in the database).
- An efficiency dividend can be achieved through simplified analysis processes.
- Visualisation system developers or vendors can increase the degree of automation and abstraction of their visualisation tool box and simplify its usage.
- Visualisation domain developers can produce domain-specific visualisations in shorter time due to an improved visualisation tool box.
- DB system developer/vendor can improve functional support of visualisation applications (by offering e.g. more powerful graph projection or additional graph schema annotation), and improve performance of query processing for visualisation applications (by leveraging extra knowledge about the application).

In order to structure our discussion on use cases, we made high-level distinctions of these with respect to the following questions: 1) Who tells the story – user or system? 2) What is the user's level of knowledge on the database content? 3) What is the goal – exploration, answering specific questions, or creating visualisations to tell a story?

Building on these discussions, we identified connections to the discussion topics in other groups, for example for the definition and evaluation of usefulness, the roles and types of audience involved, possible exploration patterns, as well as visual query support and corresponding visualisation metaphors. We plan to put our model to the test by creating instantiations of it for specific application use cases that we are familiar with, and to refine it according to the experience we gain in that process.

4.3 Working Group 3: Visual Querying and Result Visualization

Hsiang-Yun Wu (St. Pölten University of Applied Sciences, Austria, hsiang-yun.wu@fhstp.ac.at), Da Yan (The University of Alabama at Birmingham, USA, yanda@uab.edu), Michael Behrisch (Utrecht University, The Netherlands & GraphPolaris.com Analytics Platform, m.behrisch@uu.nl), Carsten Goerg (University of Colorado, USA, carsten.goerg@ucdenver.edu), Katja Hose (Aalborg University, Denmark, khose@cs.aau.dk), Pavel Klinov (Stardog, USA, pavel@stardog.com), and Fabrizio Montecchiani (University of Perugia, Italy, fabrizio.montecchiani@unipg.it)

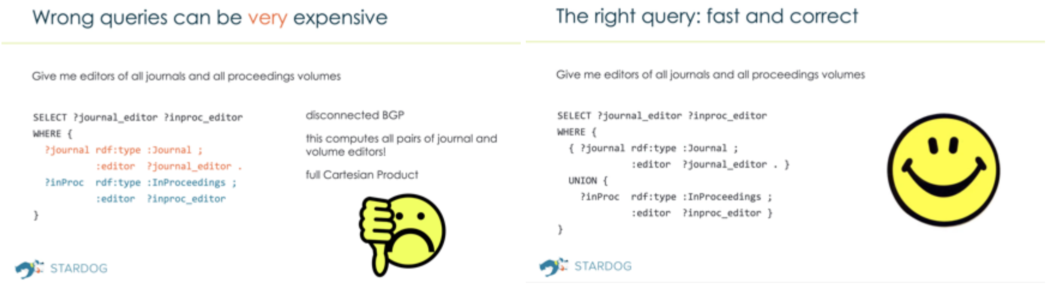
Working group 3 studied (1) the problem of **visual graph querying** which aims to assist user to formulate effective and efficient graph queries, and (2) the problem of **visualizing graph query results** for effective summarization, aggregation and human comprehension.

Visual Graph Querying (VGQ). VGQ is in contrast to the traditional graph database query languages such as SPARQL, Neo4j's Cypher, and GQL. Dagstuhl Seminar 22031 was fortunate to involve industrial attendees from Neo4j, Stardog, GraphPolaris, yWorks and data.world, which are startups and companies with graph database products already integrated with some simple frontend visualization tools. These participants have given in-depth demonstration of their products and use cases on Day 1 presentations. Working group 3 in particular has industrial members from Stardog and GraphPolaris, so in subsequent days we got a lot of chances reviewing concrete real-world use cases to see how graph database queries are applied, such as searching from enterprise knowledge graphs and biological networks.

During the discussion, members with less familiarity in the specific graph querying languages found it not easy to compose and even understand the traditional graph queries. The working group agreed upon the conclusion that learning the grammar of a graph querying language leads to a steep learning curve for end users, such as attendees with graph visualization background rather than graph database background. We expect that tools for formulating a graph query with drag-and-drop visual widgets would provide end users more intuition on what they are searching for, and it is also beneficial to support interactive and explorative query reformulation where users can learn from (at least partial) query results to incrementally revise their queries based on their search intents.

In fact, even our members from the industry admitted that their colleagues can formulate bad queries that accidentally create a large amount of unnecessary information. Figure 2 illustrates such an example to find the editors of all journals and conference proceedings from a backend graph database, where the SPARQL query on the left would unnecessarily lead to an expensive Cartesian product operation that can easily overwhelm computing and memory resources; the correct form of such a query is shown on the right which uses the UNION keyword to allow efficient execution. We expect that some visual widgets can better guide users to avoid formulating bad queries, such as giving a warning sign on excessive intermediate result size in the above SPARQL query example, or even to recommend an equivalent but more efficient query formulation. The implementation, however, would require techniques such as query cardinality estimation and seq2seq deep learning from curated (bad query, correct query) pairs captured in real enterprise operations.

The working group members identified several open challenges to address for effective VGQ. One challenge lies in how to define a set of visual querying paradigms that are effective in real applications. Several possible paradigms were discussed, including (i) pattern matching



Wrong queries can be very expensive

Give me editors of all journals and all proceedings volumes

```
SELECT ?journal_editor ?inproc_editor
WHERE {
  ?journal rdf:type :Journal ;
           :editor  ?journal_editor .
  ?inProc  rdf:type :InProceedings ;
           :editor  ?inproc_editor
}
```

disconnected BGP

this computes all pairs of journal and volume editors!

full Cartesian Product

(a) Expensive query

The right query: fast and correct

Give me editors of all journals and all proceedings volumes

```
SELECT ?journal_editor ?inproc_editor
WHERE {
  { ?journal rdf:type :Journal ;
             :editor  ?journal_editor . }
  UNION {
    ?inProc  rdf:type :InProceedings ;
             :editor  ?inproc_editor }
}
```

(b) Fast and correct query

Figure 2 An examples of a bad and a better queries.

as adopted by existing languages such as SPARQL, (ii) query-by-examples where end users list some desired results for the query engine to learn and recommend the possible queries and query semantics/intentions, (iii) query-by-sketch where end users sketch an incomplete graph query and rely on the learned data schema to auto-complete the actual query or to guide the formulation of the complete query. The group members agreed that effective data schema discovery techniques would be critical for implementing those VGQ paradigms. It is also an open problem to explore whether those query forms are sufficient for real user tasks, and how users can select among these VGQ paradigms. Novel query forms such as Cypher path matching could need to be invented to meet newly discovered querying demands as the field of VGQ progresses forward.

Graph Query Result Visualization. The output of a graph query can be of various forms such as many subgraph instances, or many path instances. Moreover, intermediate results before aggregation/reduction could be huge, requiring effective summative visualizations to make sense of the results that would be otherwise overwhelming to enumerate one by one.

A particularly interesting type of query is the path query as supported by Cypher, where end users specify a path pattern which is then matched against the backend graph database to find all matching path instances. The results are often numerous as indicated by our industrial members, and current systems usually enumerate individual path instances one after another leading to overwhelmingly many results to examine by end users. We envision that more optimized solutions can be easily developed, such as organizing the path instances (including partially matched ones) by tries so that common prefix paths can be shared to avoid redundancy. This method would not only speed up query evaluation, but also reduce the storage space requirement and the number of visual elements to display. Advanced visualization techniques can be integrated, such as making the nodes shared by more paths larger, and making the edges shared by more paths thicker. Of course, path queries are themselves relatively new, so the effectiveness of their result visualization approaches is yet to be verified in real applications.

Some other graph data are geospatial and/or topological in nature (e.g., nodes are associated with coordinates), which enable more effective visualization to bring intuition. Some effective visual representations already exist including radial layout, edge bundling and metro map metaphor, and they have been used in applications such as visualizing metro maps and metabolic pathways, but how to scale them to larger graphs effectively is still an open problem. Possible solutions include multi-scale result representation and hybrid visualization models such as NodeTrix (resp. ChordLink) which collapses dense fragments of a graph into matrices (resp. chord diagrams). See Figure 3 for an illustration.

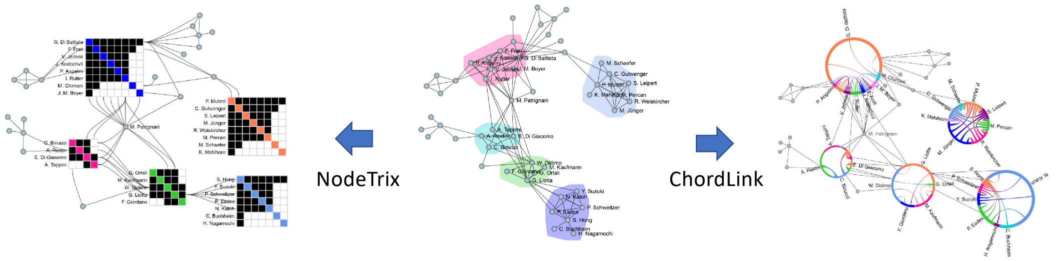
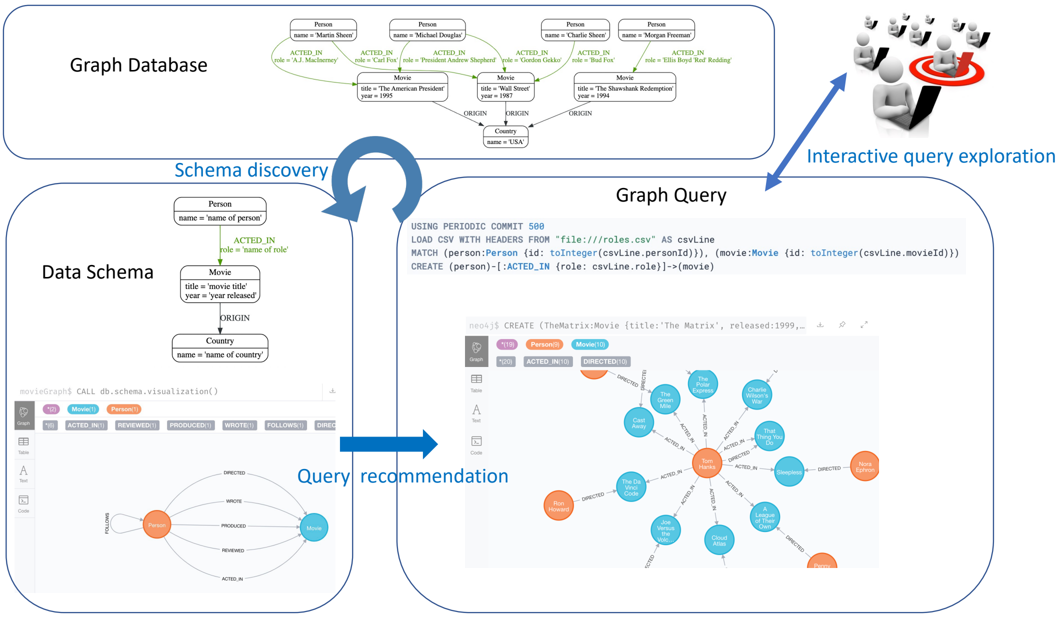


Figure 3 NodeTrix and ChordLink.



```
USING PERIODIC COMMIT 500
LOAD CSV WITH HEADERS FROM "file:///roles.csv" AS csvLine
MATCH (person:Person {id: toInteger(csvLine.personId)}), (movie:Movie {id: toInteger(csvLine.movieId)})
CREATE (person)-[:ACTED_IN {role: csvLine.role}]->(movie)
```

```
neo4j$ CREATE (TheMatrix:Movie {title:'The Matrix', released:1999,…
```

```
movieGraph$ CALL db.schema.visualization()
```

Figure 4 Integrating graph visualization and graph database.

Another interesting topic is to provide provenance explanations for graph query results, and some pioneering work has been conducted in the context of SPARQL, e.g., SPARQLprov published in PVLDB'21.

Summary: Integrating Graph Visualization and Graph Database. Figure 4 summarizes what we have discussed so far, where network visualization can be applied in the various stages of the graph querying pipeline, including data schema discovery, query result visualization and query reformulation recommendation. While a lot of those features have already been integrated into existing graph databases such as Neo4j, more diversified and advanced visualization techniques are yet to be implemented and integrated.

Participants

- Michael Behrisch
Utrecht University, NL & GraphPolaris.com Analytics Platform
- Walter Didimo
University of Perugia, IT
- Nadezhda T. Doncheva
University of Copenhagen, DK
- Henry Ehlers
TU Wien, AT
- George Fletcher
TU Eindhoven, NL
- Carsten Görg
University of Colorado – Aurora, US
- Katja Hose
Aalborg University, DK
- Karsten Klein
Universität Konstanz, DE
- Pavel Klinov
Stardog – Arlington, US
- Stephen G. Kobourov
University of Arizona – Tucson, US
- Oliver Kohlbacher
Universität Tübingen, DE
- Giuseppe Liotta
University of Perugia, IT
- Fabrizio Montecchiani
University of Perugia, IT
- Sebastian Müller
yWorks GmbH – Tübingen, DE
- Catia Pesquita
University of Lisbon, PT
- Falk Schreiber
Universität Konstanz, DE
- Juan F. Sequeda
data.world – Austin, US
- Hannes Voigt
Neo4j – Leipzig, DE
- Tatiana von Landesberger
Universität Köln, DE
- Markus Wallinger
TU Wien, AT
- Hsiang-Yun Wu
FH – St. Pölten, AT
- Da Yan
The University of Alabama – Birmingham, US