DAGSTUHL REPORTS

Volume 15, Issue 4, April 2025

Dagstuhl Reports, Vol. 15, Issue 4 ISSN 2192-5283

ISSN 2192-5283

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at
https://www.dagstuhl.de/dagpub/2192-5283

Publication date
December, 2025

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at https://dnb.d-nb.de.

Digital Object Identifier: 10.4230/DagRep.15.4.i

Aims and Scope
The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.
In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:
- an executive summary of the seminar program and the fundamental results,
- an overview of the talks given during the seminar (summarized as talk abstracts), and
- summaries from working groups (if applicable).

This basic framework can be extended by suitable contributions that are related to the program of the seminar, e. g. summaries from panel discussions or open problem sessions.

Contact
Schloss Dagstuhl – Leibniz-Zentrum für Informatik
Dagstuhl Reports, Editorial Office
Oktavie-Allee, 66687 Wadern, Germany
reports@dagstuhl.de
https://www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 25151

Disruptive Memory Technologies

Haibo Chen*[1], Ada Gavrilovska*[2], Jana Giceva*[3], and Olaf Spinczyk*[4]

1 **Shanghai Jiao Tong University, CN.** haibochen@sjtu.edu.cn
2 **Georgia Institute of Technology – Atlanta, US.** ada@cc.gatech.edu
3 **TU München – Garching, DE.** jana.giceva@in.tum.de
4 **Universität Osnabrück, DE.** olaf@uos.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 25151 "Disruptive Memory Technologies".

Memory is a central component in every computer system. Hardware evolution has lead to greater capacities and higher speeds, but essential properties of its hardware/software interface have been unchanged for decades: Main memories used to be passive, largely homogeneous, and volatile. These properties are now so firmly anchored in the expectations of software developers that they manifest in their products.

However, a wave of innovations is currently shattering these assumptions. In this sense, several new memory technologies are disruptive for the entire software industry. For example, new servers combine "high-bandwidth memory" with classic memory modules and "CXL" enables even more hybrid architectures (non-homogeneous). The "in-/near-memory" computing approaches abandon the Von Neumann architecture and promise huge performance improvements by allowing CPU-independent processing of data objects in or close to the memory (non-passive). Finally, "persistent memory" is available for servers and embedded systems (non-volatile).

Overall, the expectations are high. Computers could have lower energy consumption, more performance, improved reliability, and reduced costs. However, from the (system) software perspective it is largely unclear how to use the novel memory technology efficiently. The seminar tackled this problem by discussing the state and potential of disruptive memory technologies, the challenges for system and application software, and important research directions.

Seminar April 6–11, 2025 – https://www.dagstuhl.de/25151

2012 ACM Subject Classification Computer systems organization → Serial architectures; Hardware → Communication hardware, interfaces and storage; Information systems → Data management systems; Information systems → Information storage systems; Software and its engineering → Software organization and properties

Keywords and phrases data-centric computing, disaggregated memory, persistent memory (pmem), processing in memory (pim), system software stack

Digital Object Identifier 10.4230/DagRep.15.4.1

* Editor / Organizer

Disruptive Memory Technologies, *Dagstuhl Reports*, Vol. 15, Issue 4, pp. 1–27
Editors: Haibo Chen, Ada Gavrilovska, Jana Giceva, and Olaf Spinczyk
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Haibo Chen (Shanghai Jiao Tong University, CN)
Ada Gavrilovska (Georgia Institute of Technology – Atlanta, US)
Jana Giceva (TU München – Garching, DE)
Olaf Spinczyk (Universität Osnabrück, DE)

The continued increase in demand for data and data-intensive applications is exposing scaling limitations in the capacity and performance of traditional DRAM-based memory system designs. In response, new memory system designs are emerging, based on disaggregation, new heterogeneous memory components, and near-/in-memory processing. However, to fully leverage the benefits of the hardware innovation requires redesign of the software stack, from the low-level operating system and virtualization primitives managing hardware access, to programming and compiler support and application runtimes. This seminar gathered researchers from industry and academia working across the entire stack, to discuss the pressing challenges in this new memory landscape, and to identify the most promising paths forward. The five-day seminar was structured to guide the discussion across the different layers:

Day 1 set the stage by first discussing the driving challenges from the database and datacenter communities, and on surveying the state-of-the-art of the hardware technologies for processing-near memory, processing-in memory and for disaggregation (with emphasis on CXL).

Day 2 included a deep-dive in diverse use cases, including bioinformatics and drug discovery, database processing, visual analytics, and AI/ML, followed by presentations on lessons-learned from practical adoptions of memory disruptors such as persistent memory.

Day 3 focused on identifying the limitations of state-of-the-art software technologies in leveraging new memory capabilities, complemented by a panel discussion addressing challenges of cross-stack co-design to optimize their potentials, considering both general-purpose and domain-specific needs.

Day 4 explored vision talks outlining future research directions for Processing-in-Memory (PIM) , Processing-near-Memory (PNM) , hyper-heterogeneous computing, and software-hardware co-design, etc.

Day 5 concluded with collaborative proposal discussions, synthesizing insights to define a strategic roadmap for advancing these technologies.

The breakout sessions, visionary talks, and panel discussions proved highly effective in identifying the challenges and opportunities posed by disruptive memory technologies. These discussions were particularly impactful due to the diverse mix of participants, including experts from hardware, computer architecture, operating systems, databases, and parallel software domains, representing both academia and industry. For instance, industry leaders and academic researchers approached emerging memory technologies from complementary perspectives, and the ensuing vigorous debates helped broaden understanding across both sectors. Insights shared by practitioners provided researchers with valuable context to refine and prioritize critical research questions. Our report offers a comprehensive summary of the seminar's key discussions and outcomes.

2 Table of Contents

3 Overview of Talks

3.1 Vertical integration for more efficient memory/storage

Gustavo Alonso (ETH Zürich, CH)

In this talk I will argue that the best way to use the advances in hardware around storage and memory is to build vertically integrated systems. By that, I mean building end-to-end pipelines of processing elements from computational storage through smart NICs, near memory accelerators, processors in memory, programmable switches, etc. This type of architecture is possible in the context of the increasing use of scale up architectures as alternative to the scale out, elastic, highly distributed systems common in the cloud. Scale up systems provide a more compact system with higher density of storage, memory and computational power, making them ideal to explore data processing pipelines involving active elements all along the data path from storage to the processor registers.

3.2 Memory and Storage Challenges in ML Pipelines

Oana Balmau (McGill University – Montréal, CA)

The rapid expansion of machine learning (ML) workloads has introduced significant challenges for memory and storage systems. In this talk, we highlight critical pain points along the ML pipeline – spanning training, checkpointing, and inference – with a focus on storage inefficiencies and emerging opportunities for innovation.

During training, storage bottlenecks are primarily caused by the exponential growth of dataset sizes, which increasingly exceed system memory and must be accessed from persistent storage. While existing dataloaders serve as the bridge between storage and compute, they are poorly equipped to handle tiered storage architectures and exhibit high variability in sample preprocessing times. These inefficiencies lead to head-of-line blocking and underutilized GPUs, with utilization often falling to 30% in real workloads. There are multiple opportunities for improvement, including hardware-software co-design, processing-in-memory (PIM) capabilities, and intelligent caching mechanisms.

A second major challenge is the write-intensive nature of model checkpointing. Modern checkpoints encompass model weights, optimizer states, and various metadata, with storage footprints ranging from 100 GB for 7B models to 17 TB for trillion-parameter models. Checkpointing requires all-to-all synchronization across nodes and is prone to stragglers and failures, making efficient recovery a pressing problem. Persistent memory (PM) and emerging interconnects like CXL offer promising avenues to mitigate these bottlenecks.

In the inference phase, large language models (LLMs) introduce new memory pressures due to key-value (KV) caches. To meet stringent service-level objectives (e.g., 200–450ms time to first token), these caches must reside in GPU memory and are maintained per user. For example, serving 1200 users with a LLaMA 2 70B model under realistic workloads requires approximately 3 TB of KV cache memory – far exceeding the combined 640 GB available on 8 H100 GPUs. The situation is further exacerbated in Retrieval-Augmented

Generation (RAG) pipelines, where context lengths may reach 200K tokens and additional cache entries are needed for knowledge base documents. Efficient management of these caches – through tiered storage, cache offloading, or optimized vector search – is becoming a crucial frontier.

To address these mounting challenges, MLCommons has launched the MLPerf Storage benchmark suite, which aims to systematically evaluate and improve the storage performance of ML pipelines. MLPerf Storage targets diverse scenarios, including training, checkpointing, and inference, by simulating real-world workloads and stressing different components of the storage hierarchy. This benchmark provides a common ground for researchers and practitioners to test new systems, co-design solutions, and drive forward innovations that ensure scalable and efficient ML pipelines.
More information on MLPerf Storage is available at `https://mlcommons.org/working-groups/benchmarks/storage/`.

3.3 CXL Pooling: What's Real and When?

Daniel Berger (Microsoft – Redmond, US)

This talk briefly reviews the broad opportunities promised by the CXL standard and then dives into pooling. Pooling and disaggregation at scale sound appealing but bring many implementation and deployment challenges. CXL switches are slow and expensive. Large multi-ported devices are cheaper and faster, but efforts to scale them have not been successful. We are essentially left with only two-ported pooling devices.

This talk proceeds with a proposal to build larger CXL pods out of these smaller devices. The key idea is simple: connect every CPU to multiple pooling devices, which in turn connect to different subsets of CPUs. Instead of connecting every device to every CPU ("fully connected pods"), our design connects every pair of CPUs to a shared pooling device ("partially connected pods"). This leads to cheap and potentially large pods but requires more complexity in the software stack. For example, CPUs need to carefully place data into the right pooling device that is accessible by those CPUs it seeks to communicate or collaborate with.

3.4 Programming and compiler abstractions for emerging computer architectures

Jerónimo Castrillón-Mazo (TU Dresden, DE)

Main reference Asif Ali Khan, Hamid Farzaneh, Karl Friedrich Alexander Friebel, Clément Fournier, Lorenzo Chelini, Jerónimo Castrillón: "CINM (Cinnamon): A Compilation Infrastructure for Heterogeneous Compute In-Memory and Compute Near-Memory Paradigms", in Proc. of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 4, ASPLOS 2024, Hilton La Jolla Torrey Pines, La Jolla, CA, USA, 27 April 2024 – 1 May 2024, pp. 31–46, ACM, 2024.
URL http://dx.doi.org/10.1145/3622781.3674189

Compute-in-Memory (CIM) is a promising non-von Neumann computing paradigm that promises unprecedented improvements in performance and energy efficiency. Moving past manual designs, automation will be key to unleash the potential of CIM for multiple application domains and to accelerate cross-layer design cycles. This talks reports on an ongoing effort to build a high-level compiler infrastructure for different CIM approaches, built with MLIR to abstract from individual technologies to foster re-use. This includes abstractions and optimizations flows for logic-in memory [2], content-addressable memories [3, 1], arithmetic operations in crossbars [4], and near-memory architectures.

References

1. Hamid Farzaneh, João Paulo Cardoso de Lima, Mengyuan Li, Asif Ali Khan, Xiaobo Sharon Hu, Jeronimo Castrillon. *C4CAM: A Compiler for CAM-based In-memory Accelerators*. Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'24), Volume 3, Association for Computing Machinery, pp. 164–177, New York, NY, USA, May 2024.
2. Hamid Farzaneh, João Paulo Cardoso De Lima, Ali Nezhadi Khelejani, Asif Ali Khan, Mahta Mayahinia, Mehdi Tahoori, Jeronimo Castrillon. *SHERLOCK: Scheduling Efficient and Reliable Bulk Bitwise Operations in NVMs*. Proceedings of the 61th ACM/IEEE Design Automation Conference (DAC'24), Association for Computing Machinery, New York, NY, USA, Jun 2024.
3. João Paulo C. de Lima, Asif Ali Khan, Luigi Carro, Jeronimo Castrillon. *Full-Stack Optimization for CAM-Only DNN Inference*. Proceedings of the 2024 Design, Automation and Test in Europe Conference (DATE), IEEE, pp. 1-6, Mar 2024.
4. Adam Siemieniuk, Lorenzo Chelini, Asif Ali Khan, Jeronimo Castrillon, Andi Drebes, Henk Corporaal, Tobias Grosser, Martin Kong. *OCC: An Automated End-to-End Machine Learning Optimizing Compiler for Computing-In-Memory*. In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), IEEE Press, vol. 41, no. 6, pp. 1674-1686, Aug 2021.

3.5 Systems Software Meets Persistent Memory: Lessons and Experiences

Haibo Chen (Shanghai Jiao Tong University, CN)

A central challenge in storage research has long been balancing durability, speed, and capacity. Persistent memory, with its promise of durability, high-speed byte addressability, and scalable capacity, emerged as a transformative solution to this enduring tension. Over the past 12 years, our research and practical exploration has focused on rethinking the systems software stack to harness the potential of emerging hardware like persistent memory.

In this talk, I first examine our early efforts to redesign systems – including storage engines, file systems, distributed logging, and RDMA protocols – around nascent persistent memory prototypes. While these innovations demonstrated compelling performance gains in databases, file systems, and key-value stores, widespread adoption was hindered by immature hardware, cost inefficiency, and the need for intrusive changes to existing software stacks.

We then turn to recent advancements leveraging cutting-edge persistent memory and RDMA technologies to optimize transactional workloads in distributed, many-core database clusters. Our latest work achieves significant performance improvements by minimizing transactional overhead, demonstrating the viability of persistent memory-enabled systems in modern architectures.

Finally, I will reflect on our key lessons learned: the necessity of an evolutionary path for practical adoption, prioritizing solutions to critical problems over broad but shallow innovations, and the interplay between hardware maturity and software design. This journey underscores how emerging hard like persistent memory continues to reshape the frontier of storage research, demanding both technical rigor and strategic pragmatism in systems innovation.

3.6 50 years and Beyond – a look in the rearview mirror before looking forward

David Cohen (Solidigm – Santa Fe, US)

Amin Vahdat's "5th Epoch" keynote at the SIGCOMM 2020 a frame for more than 50 years of distributed computing. At the beginning of this period there were scale-up clusters but these gave way to roughly 40 years of x86-based, scale-out computing. In the wake of the End of Denard Scaling and the rise of Machine Learning, x86/Scale-Out is being disrupted by Accelerator-Centric, Scale-Up clusters that Harkin back to the beginning of the period. This talk looks at this evolution and questions assumptions that are biased toward x86/scale-out orthodoxy.

References

1 Amin Vahdat. *Coming of Age in the Fifth Epoch of Distributed Computing: The Power of Sustained Exponential Growth*. SIGCOMM 2020, keynote address.
2 P. Ranganathan and U. Holzle, *Twenty Five Years of Warehouse-Scale Computing*. In IEEE Micro, vol. 44, no. 05, pp. 11-22, Sept.-Oct. 2024, doi: 10.1109/MM.2024.3409469.

3.7 A Programming Model for CXL

Michal Friedman (ETH Zürich, CH)

Joint work of Michal Friedman, Gal Assa, Ori Lahav, Lucas Burgi
Main reference Gal Assa, Michal Friedman, Ori Lahav: "A Programming Model for Disaggregated Memory over CXL", CoRR, Vol. abs/2407.16300, 2024.
URL http://dx.doi.org/10.48550/ARXIV.2407.16300

CXL (Compute Express Link) is an emerging open industry-standard interconnect between processing and memory devices that is expected to revolutionize the way systems are designed in the near future. It enables cache-coherent shared memory pools in a disaggregated fashion at unprecedented scales, allowing algorithms to interact with a variety of storage devices using simple loads and stores. Alongside unleashing unique opportunities for a wide range of applications, CXL introduces new challenges of data management and crash consistency. Alas, CXL lacks an adequate programming model, which makes reasoning about the correctness and expected behaviors of algorithms and systems on top of it nearly impossible. In this talk I'll present CXL0, the first programming model for concurrent programs running on top of CXL. I'll present a high-level abstraction for CXL memory accesses backed up by a formal definition of operational semantics on top of that abstraction.

3.8 Using Performance-Aware Behaviour Models for Hardware Characterization and Performance Prediction

Birte Kristina Friesel (Universität Osnabrück, DE)

Joint work of Birte Friesel, Olaf Spinczyk
Main reference Birte Friesel, Olaf Spinczyk: "Performance-Aware Behaviour Models for Feature-Dependent Runtime Attributes in Product Lines", in Proc. of the 19th International Working Conference on Variability Modelling of Software-Intensive Systems, VaMoS 2025, Rennes, France, February 4-6, 2025, pp. 131–135, ACM, 2025.
URL http://dx.doi.org/10.1145/3715340.3715435

Cost models are pervasive throughout the system stack, for instance to decide whether it is sensible to move data to faster memory prior to processing. Typically, they are embedded into scheduling or placement algorithms, and not designed to be understandable or usable outside of those. This vision talk proposes behaviour models as a common formalism for cost models in system software, and shows how they can help understand and predict the performance of database operations on UPMEM PIM memory modules. Behaviour models decompose runtime actions into individual steps within a state machine, and associate each step (i.e., transition) with a distinct performance model. Thus, they allow for efficient hardware characterization and reasoning about performance, while still being compatible with (and usable as) conventional cost models.

3.9 Input from the Data Management community – challenges and opportunities

Jana Giceva (TU München – Garching, DE)

For decades data management systems and their well-defined workloads have been a great source of inspiration for hardware acceleration and a testing ground for new technologies. In this talk, we give a short overview of the different types of data-intensive workloads and their specific requirements and characteristics. After a brief historical overview of prior customized approaches to accelerate their workloads over the years (e.g., the idea of the database machine, data appliances and custom data accelerators), we go into what changes today with resource disaggregation in the cloud. One proposal is to push compute along the data path so we can have processing close to where to data sits or as it moves. For that we argue for a novel abstraction layer based on declarative operator primitives that decouples semantics from imperative implementation and explicitly reasons about state, data placement and movement.

3.10 Don't forget the "else" workloads (they are the majority!)

Boris Grot (University of Edinburgh, GB)

Joint work of Amna Shahab, Mingcan Zhu, Artemiy Margaritov, Boris Grot
Main reference Amna Shahab, Mingcan Zhu, Artemiy Margaritov, Boris Grot: "Farewell My Shared LLC! A Case for Private Die-Stacked DRAM Caches for Servers", in Proc. of the 51st Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2018, Fukuoka, Japan, October 20-24, 2018, pp. 559–572, IEEE Computer Society, 2018.
URL http://dx.doi.org/10.1109/MICRO.2018.00052

The slowdown in technology scaling mandates rethinking of conventional CPU architectures in a quest for higher performance and new capabilities. As a step in that direction, this talk questions the value of on-chip shared last-level caches (LLCs) in server processors and argues for a better alternative leveraging 3D memory stacking technology and tight CPU/memory integration.

3.11 The Persistence of Big Memory: Unlocking Memory Hierarchy

Yu Hua (Huazhong University of Science & Technology – Wuhan, CN)

The artificial separation between volatile memory and persistent storage has long constrained system architectures, resulting in inefficient data movements across registers, cache, DRAM, and block storage. Big Memory flattens this hierarchy by making persistence an inherent property of the memory layer with the aid of networking technologies. The memory pooling in CXL creates a unified address space across disaggregated nodes, allowing memory resources to be dynamically composed and shared with local-DRAM latency. The zero-copy and kernel-bypass mechanisms in RDMA mitigate storage stack overhead by enabling direct memory-to-memory transfers between hosts, bypassing traditional CPU and OS involvements. Byte-addressable memory semantics further replace block I/O as the fundamental access primitive. This transformation doesn't merely blur the line between memory and storage, and systematically creates a continuum where all data exist natively in persistent memory.

This architectural breakthrough stems from a fundamental insight: persistence must be the foundational property of memory systems. By establishing persistence as the first principle, Big Memory reconfigures the traditional storage-memory dichotomy. Instead of treating persistence as a storage-tier feature, it becomes a native capability of the memory layer. This core principle directly enables the unlocked memory hierarchy. When the memory is inherently persistent, the entire architecture of registers/cache/DRAM/storage coalesces into a flat, scalable memory plane. The result is a self-consistent architecture where persistence enables hierarchy flattening that in turn maximizes the benefits of persistent memory, creating a virtuous cycle of efficiency and performance.

3.12 Near-Storage and Near-Memory – Two Peas in a Pod

Sudarsun Kannan (Rutgers University – Piscataway, US)

In this talk, I will motivate and explore how operating systems must evolve to effectively support near-storage and near-memory processing in the face of increasing hardware heterogeneity. I will begin by providing a historical perspective on how these technologies have evolved over the past three to four decades, from early intelligent disks to today's commercial computational storage and near-memory accelerators. Traditional systems were designed with a binary view of memory and storage, but emerging workloads and devices demand more fine-grained data placement and processing strategies. I will describe our efforts to reduce data movement and processing overheads through techniques such as CISC-style I/O operations, horizontal caching, and metadata-aware offloading. By viewing near-storage and near-memory as complementary, we enable collaborative computation across tiers and improve performance. I will also highlight major system software challenges and briefly discuss the opportunities these technologies present for edge-aware systems.

References

1. Y. Ren, C. Min, and S. Kannan. *CrossFS: a cross-layered direct-access file system*. In Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation (OSDI'20). USENIX Association, USA, Article 8, 137–154, 2020.
2. J. Zhang, Y. Ren, and S. Kannan. *FusionFS: Fusing I/O operations using CISCOps in firmware file systems*. In Proceedings of the 20th USENIX Conference on File and Storage Technologies (FAST 22), pages 297- 312, Santa Clara, CA, February 2022. USENIX Association.
3. J. Zhang, Y. Ren, M. Nguyen, C. Min, and S. Kannan. *OmniCache: collaborative caching for near-storage accelerators*. In Proceedings of the 22nd USENIX Conference on File and Storage Technologies (FAST '24). USENIX Association, USA, Article 3, 35–50, 2024.

3.13 Thanks for the Memories: Lessons from Disruptive Memory Technologies

Kimberly Keeton (Google LLC – South San Francisco, US)

In this talk, I explore the factors that contribute to the success (or failure) of a disruptive memory technology. I begin by surveying the evolution of memory technologies over the last 75+ years. Looking across multiple generations, several themes emerge as to why technologies have faded, including: 1) its performance/cost/density/reliability wasn't as good as another (often newer) alternative; 2) production, supply chain, and/or operational challenges; 3) a lack of widespread adoption / insufficient use cases; and 4) limited compatibility. After exploring several concrete examples, I propose several ingredients in a recipe for maximizing the chances for success, including: 1) compelling use cases and/or benefits; 2) support for understanding technology characteristics; 3) support for developing the software ecosystem; and 4) open standards. I conclude by posing several questions for discussion by the group.

3.14 Industry Talk – "Near-Memory Processing is Becoming a Reality"

Hoshik Kim (SK hynix – San Jose, US)

In this Industy talk session, Hoshik explained the background of growing demands for near-memory processing and the opportunities from industry perspective as well as the state-of-the-art of the technology and available products and prototypes for research communities.

The demands for near-memory processing are growing due to data-intensive characteristics of modern AI and data analytics workloads and gigantic energy consumption in data centers. The evolution of CXL and Custom HBM also opens a door to near-memory processing.

While near-memory processing has a great potential, Hoshik also highlighted that there are still many technical challenges ahead and innovations are required to commercialize the technology.

3.15 Memory-Centric Computing: Recent Advances in Processing-in-DRAM

Onur Mutlu (ETH Zürich, CH)

Main reference Onur Mutlu, Saugata Ghose, Juan Gómez-Luna, Rachata Ausavarungnirun: "A Modern Primer on Processing in Memory", CoRR, Vol. abs/2012.03112, 2020.
URL https://arxiv.org/abs/2012.03112

Computing is bottlenecked by data. Large amounts of application data overwhelm the storage capability, communication capability, and computation capability of the modern machines we design today. As a result, many key applications' performance, efficiency, and scalability are bottlenecked by data movement. In this talk, we describe three major shortcomings of modern architectures in terms of 1) dealing with data, 2) taking advantage of vast amounts of data, and 3) exploiting different semantic properties of application data. We argue that an intelligent architecture should be designed to handle data well. We posit that handling data well requires designing architectures based on three key principles: 1) data-centric, 2) data-driven, 3) data-aware. We give examples of how to exploit these principles to design a much more efficient and higher performance computing system. We especially discuss recent research that aims to fundamentally reduce memory latency and energy, and practically enable computation close to data, with at least two promising directions: 1) processing using memory, which exploits the fundamental operational properties of memory chips to perform massively-parallel computation in memory, with low-cost changes, 2) processing near memory, which integrates sophisticated additional processing capability in memory chips, the logic layer of 3D-stacked technologies, or memory controllers to enable near-memory computation with high memory bandwidth and low memory latency. We show both types of architectures can enable order(s) of magnitude improvements in performance and energy consumption of many important workloads, such as machine learning, graph analytics, database systems, video processing, climate modeling, genome analysis. We discuss how to enable adoption of such fundamentally more intelligent architectures, which we believe are key to efficiency, performance, and sustainability. We conclude with some research opportunities in and guiding principles for future computing architecture and system designs.

3.16 Some Experience with PMem, NVLink, CXL

Tilmann Rabl (Hasso-Plattner-Institut, Universität Potsdam, DE)

Main reference Felix Werner, Marcel Weisgut, Tilmann Rabl: "Towards Memory Disaggregation via NVLink C2C: Benchmarking CPU-Requested GPU Memory Access", in Proc. of the 4th Workshop on Heterogeneous Composable and Disaggregated Systems, HCDS 2025, Rotterdam, TheNetherlands, 30 March 2025, pp. 8–14, ACM, 2025.
URL http://dx.doi.org/10.1145/3723851.3723853

The memory hierarchy is becoming increasingly complex and heterogeneous. Modern servers include not only different levels of caches and main memory, but can also feature persistent memory, as well as, remote coherent memory through high speed interconnects. A recent addition is CXL, which is already supported in many new CPUs and a hot topic of research. Similarly, modern GPU servers often contain alternative high speed interconnects, such as NVLink and InfiniFabric, which serve a similar purpose.

In this presentation, we give an overview of our results evaluating persistent memory (PMem) in the form of Intel Optane and report on our experience in using it in a hybrid DRAM/PMem key-value store. We further evaluate the performance of CPU initiated GPU memory access through NVLink and show how fast interconnects enable fast GPU-based DB operations. We finally translate our findings to CXL and how first experiments on prototypical hardware.

3.17 Accelerating Bioinformatics Workloads

Tajana Simunic Rosing (University of California – San Diego, US)

Dramatic decreases in the cost of sequencing and other biological and chemical data acquisition has created a deluge of biomedical data. Personalized medicine is driven today by genomics, transcriptomics, proteomics, and metabolomics, which characterize the interactions between genes, RNA molecules, proteins, and metabolites respectively. These four "omics" disciplines are a key to understanding complex diseases including cancer, autoimmune disease, and response to infection, and are crucial for advances in medical diagnostics and therapeutics. Most omics data is generated by sequencing or mass spectrometry, using common tools and analysis pipelines that rely on CPU-based servers. Such systems spend a large majority of the time just moving data from storage and memory into the processors. For example, our recent profiling of algorithms used for analysis of mass spectrometry data shows that 80% of the run time is dominated by moving data from storage for preprocessing. We got similar results when profiling our COVID-19 genome sequence analysis pipeline.

Our team accelerated a number of key tools used for applications such as microbiome, COVID-19 analysis, protein and peptide mass spectrometry, using novel machine learning techniques, such as hyperdimensional computing, and in-memory and in-storage acceleration. Our accelerated microbiome and COVID-19 pipelines have been used by the UCSD medical center to support clinical investigations and the award-winning "Return to Learn" program that ensures student, faculty and staff safety on campus including sequencing tens of thousands of complete viral genomes from clinical cases, asymptomatic population screening, and wastewater. More recently, CDC has adopted some of our work as well. The in-memory sequence alignment reduced the time needed to go from the output of the sequencer to results by 1,900x, making it possible to get results in less than a second! Just the preprocessing step of mass spectrometry data analysis has been accelerated 180x by doing in-storage analysis. Hyperdimensional computing was used to accelerate mass spectrometry protein and peptide identification using in and near memory computing, and resulted in 5,000x speed up relative to the state of the art, at comparable accuracy. Such results go a long way toward ushering a new age of portable tools that can be used for personalized medicine in near real time to benefit individual patients today, not only in cohort studies that take years to analyze and benefit only future patients.

3.18 Suggestions for a PIM research agenda

Kevin Skadron (University of Virginia – Charlottesville, US)

After settling on a working definition of processing in memory (PIM) as placing logic within the memory die and processing near memory (PNM) as any logic outside the memory die (e.g., directly interfaced to the memory channel), discussions regarding future PIM research directions emphasized several topics.

One major issue was whether PIM will be more beneficial as an accelerator that is separate from the regular "host memory" ("accelerator-first") or as an additional feature within main memory ("memory first"). Some difficulties with the memory-first approach are that logic is costly in die area in a DRAM process, sacrificing memory capacity or severely limiting PIM functionality; that contention between PIM access and conventional memory read/write may negatively impact latency for the latter category; and that memory address interleaving is not friendly to PIM. However, requiring data to be copied from main memory to an accelerator-first PIM device sacrifices the opportunity to perform computing without moving data, and instead is limited to leveraging the internal parallelism of DRAM (which in itself is a powerful acceleration story). A hybrid approach is possible by mapping the accelerator-first PIM into the physical address space, or connecting it through CXL, but these likely sacrifice data access bandwidth and latency for conventional memory read/write.

Beyond these considerations, the compute placement (subarray vs bank-level), functionality (natively supported compute capabilities), and buffering (registers or scratchpad) are open areas of research. In addition, many applications will benefit from some mechanism for bank-to-bank and channel-to-channel communication without having to go through the host-side memory controller.

A further important consideration is how PIM should be programmed. The tight coupling of data placement and computation on that data are not well expressed in current programming languages.

Finally, participants discussed how PIM should be interfaced to the system, including virtual memory abstractions, and how PIM computation should be invoked.

3.19 Memory Management for the 21st Century

Michael Swift (University of Wisconsin-Madison, US)

The complexity of operating system memory management is increasing along with the complexity of compute, memory, acceleration, and I/O hardware. Current memory managers are typically monolithic code with brittle, inflexible policies.

In this talk I discuss our work on principled policies using a cost-benefit approach, showing that considering the cost of OS memory operations can dramatically reduce latency and improve performance. I also consider the challenge of modifying policies and show how to use the Linux virtual file interface (VFS) as an effective approach for introducing new memory management approaches. Finally, I address the challenges of tiered memory management in the OS, showing how lack of information can lead to poor policies that do not optimize tiering performance.

3.20 HYPNOS – Co-Design of Persistent, Energy-efficient and High-speed Embedded Processor Systems with Hybrid Volatility Memory Organization

Jürgen Teich (Universität Erlangen-Nürnberg, DE)

Joint work of Nils Wilbert, Stefan Wildermann, Jürgen Teich
Main reference Nils Wilbert, Stefan Wildermann, Jürgen Teich: "Hybrid Cache Design Under Varying Power Supply Stability – A Comparative Study", in Proc. of the International Symposium on Memory Systems, MEMSYS 2024, Washington, DC, USA, 30 September 2024 – 3 October 2024, pp. 257–269, ACM, 2024.
URL http://dx.doi.org/10.1145/3695794.3695819

This talk discusses the idea to introduce modern non-volatile memory (NVM) technology in the cache architecture of CPUs with the goal to save energy and and potentially save execution time of programs. Particularly for IoT devices that are subject to intermittend power outages, NVM caches sound promising. But in order to to not give up high clock rates (speed), a CPU cannot be built completely using NVM due to endurance limitations, hybrid solutions seem to be sound solution.

After an introduction to hybrid cache architectures, we present a design of a low-latency energy-efficient hybrid cache design for embedded systems and analyze different techniques on which data to keep in the volatile and which data to keep in the non-volatile caches lines and how to backup volatile data into non-volatile sections upon power outages.

From experimental results using a cycle-accurate simulation based on extensions of the GEM5 environment, we can see that different benchmarks do require different cache policies in order to save the highest amount of energy. It is also shown that current technologies of NVM can only be although not severely degrade the performance. For best exploitation of energy savings, we believe that hints coming from program analysis and compiler could provide an even better co-designed solution for such promising hybrid cache CPU architectures.

3.21 Memory Systems for Visual Computing: Challenges and Opportunities

Nandita Vijaykumar (University of Toronto, CA)

Visual computing has been revolutionized by emerging applications like multi-modal deep neural networks, generative AI, and differentiable rendering. These applications demand systems that can handle unprecedented challenges in efficiency, scalability, and security. In this talk, I will explore new challenges in efficiency by emerging visual computing applications and how innovations and technologies in memory systems can help address them.

3.22 Vision on Heterogeneous Hardware

Zeke Wang (Zhejiang University – Hangzhou, CN)

Joint work of Zeke Wang, Jie Zhang, Hongjing Huang, Yingtao Li, Xueying Zhu, Mo Sun, Zihan Yang, De Ma, Huajing Tang, Gang Pan, Fei Wu, Bingsheng He, Gustavo Alonso
Main reference Zeke Wang, Jie Zhang, Hongjing Huang, Yingtao Li, Xueying Zhu, Mo Sun, Zihan Yang, De Ma, Huajin Tang, Gang Pan, Fei Wu, Bingsheng He, Gustavo Alonso: "FpgaHub: Fpga-centric Hyper-heterogeneous Computing Platform for Big Data Analytics", CoRR, Vol. abs/2503.09318, 2025.
URL http://dx.doi.org/10.48550/ARXIV.2503.09318

Modern data analytics requires a huge amount of computing power to analyze on a massive amount of data. Therefore, heterogeneous data analytics platforms appear as an attempt to close the gap between compute power and data. For example, FPGA-based systems, GPU-based systems, SSD, and programmable switches.

Even though these heterogeneous systems achieve better performance than CPU-only homogeneous system due to the dying of Moore's law, we still identify that simple heterogeneous systems cannot harvest the potential of each heterogeneous device. For example, GPU-based approaches suffer from severe IO issues and programmable switch, e.g., P4 switch, has very limited compute and memory capacities.

To this end, we argue for hyper-heterogeneous computing for big data analytics. Therefore, we present FpgaHub, a hyper-heterogeneous computing platform for big data analytics. Our platform is centralized on FPGA-based SmartNIC, and explores its potential of co-optimization with any other heterogeneous devices, such as GPU, P4 switch, and SSD. The key idea of FpgaHub is to use FPGA to complement other heterogeneous devices via deep co-design, with a goal of "1+1 > 2".

3.23 CXL Memory – Where are we today?

Thomas Willhalm (Intel Deutschland GmbH – Feldkirchen, DE)

Today, the compute capabilities of servers are growing at a much faster pace than the memory capacity and bandwidth. This creates pressure on the number of pins and DIMM slots on a motherboard for realizing a balanced system. As a way to mitigate these limits, CXL memory (type 3) devices can expand the memory capacity and bandwidth using PCIe channels on current x86 processors. CXL memory devices come in various form factors, which provide different use cases: EDSFF in several sizes, number of PCIe lanes, and power envelopes as well as PCIe add-in cards, which also allow the (re-)usage of conventional DDR4 and DDR5 modules.

The latest Intel Xeon 6 processors support CXL 2.0 devices in two modes. By default, CXL memory is exposed as a separate NUMA node without cores assigned. OS and applications then manage the data placement based on the proximity information provided by the ACPI tables. In contrast to that, "Flat Memory Mode" provides a flat access to memory. The hardware is then managing the tiering between native DRAM and CXL memory on a cache line level transparently to the OS and application: Each cache line in the physical address space is either located in native DRAM or CXL memory. If the cache line is accessed, it will simply be returned in case it is located in native DRAM. In case the cache line is found in

CXL memory, the cache line will be moved to native DRAM, evicting some other cache line from native DRAM to CXL memory. Flat Memory Mode therefore requires a 1:1 ratio for native DRAM and CXL memory, but has the advantage that the total capacity of native DRAM and CXL memory can be accessed.

Beyond memory expansion, the CXL 3.0 standard will provide further capabilities for "memory pooling". This opens the door for innovative usage taking multiple hosts into account: dynamic memory allocation, data sharing, preserving data in the pool during restart, or producer-consumer access. These innovative use-case come with the burden that the whole stack from hardware, OS and hypervisor, device drivers and libraries, to the application level need to support the required capabilities. The best way to achieve broad adoption are solutions that work out-of-the-box and bring benefit for existing user scenarios but offer the required extensions for novel use cases. Potential targets are therefore distributed systems that could benefit from faster or simpler communication.

3.24 Memory Requirements and Challenges in Large Language Models (LLMs)

Chun Jason Xue (MBZUAI – Abu Dhabi, AE)

Main reference Hongchao Du, Shangyu Wu, Arina Kharlamova, Nan Guan, Chun Jason Xue: "FlexInfer: Breaking Memory Constraint via Flexible and Efficient Offloading for On-Device LLM Inference", in Proc. of the 5th Workshop on Machine Learning and Systems, EuroMLSys 2025, World Trade Center, Rotterdam, The Netherlands, 30 March 2025- 3 April 2025, pp. 56–65, ACM, 2025.
URL http://dx.doi.org/10.1145/3721146.3721961

The deployment of Large Language Models (LLMs) still faces significant challenges due to their extensive memory requirements, especially on mobile and edge devices. For instance, even a quantized 7B-parameter model demands over 7GB of memory, exceeding the capacity of most mobile platforms. This talk explores the memory requirements and challenges in Large Language Models (LLMs), highlighting the shift from computational efficiency in traditional DNNs to memory efficiency as a critical bottleneck in modern LLMs. This talk discusses key challenges during inference, such as the growing model size and the high memory footprint of KV cache, along with optimization techniques like quantization, pruning, and offloading. The talk also introduces memory constraints in LLM training, including pre-training and post-training phases, and emphasizes solutions like parallel training and mixed precision. Additionally, it covers system-level optimizations such as virtual memory management and prefetching to enhance efficiency. The talk concludes by underscoring the importance of balancing memory and computational efficiency to advance LLM capabilities.

4 Working groups

4.1 Report of the Working Group on Hardware Coherence

Antonio Barbalace (University of Edinburgh, GB)

Hardware coherence is not the only solution for enabling inter-machine CXL memory sharing – it largely depends on the specific application. While many potential use cases for CXL memory sharing have been identified, including VM and container migration, Spark, key-value stores, databases, distributed AI/ML workloads (such as Ray), distributed file systems, high-frequency trading, and legacy system support, not all of them require hardware coherence. In general, the working group believes that the need for full address-space hardware coherence appears to be quite limited.

That said, certain scenarios may still benefit from some degree of hardware coherence. These include metadata management in distributed file systems, in-memory databases and transactional systems, high-frequency trading applications where low latency is critical, and support for legacy software that assumes hardware coherence. In these cases, hardware-based coherence can improve performance or simplify development, but for most other applications, software-based solutions or hybrid approaches may be more appropriate.

4.2 Report of the Working Group on Operating Systems Issues

David Cohen (Solidigm – Santa Fe, US), Frank Bellosa (KIT – Karlsruher Institut für Technologie, DE), Daniel Berger (Microsoft – Redmond, US), Yu Hua (Huazhong University of Science & Technology, CN), Kimberly Keeton (Google LLC – South San Francisco, US), Michael Swift (University of Wisconsin-Madison, US), and Thomas Willhalm (Intel Deutschland GmbH – Feldkirchen, DE)

An ad hoc working group on the role of the operating system for systems with disruptive memories met. The discussion covered topics such as how much should be delegated to applications, whether operating systems must get more disaggregated, what the right high-level abstractions are, how to secure and protect data, and what are the goals for operating system memory management.

4.3 Report of the Working Group on CXL Sustainability and Memory Fabric

Ada Gavrilovska (Georgia Institute of Technology – Atlanta, US), Gustavo Alonso (ETH Zürich, CH), Daniel Berger (Microsoft – Redmond, US), Thaleia Dimitra Doudali (IMDEA Software Institute – Madrid, ES), Hoshik Kim (SK hynix – San Jose, US), Alberto Lerner (University of Fribourg, CH), Ilia Petrov (Hochschule Reutlingen, DE), Tilmann Rabl (Hasso-Plattner-Institut, Universität Potsdam, DE), and Zeke Wang (Zhejiang University – Hangzhou, CN)

The breakout session brought seminar participants interested in two questions related to CXL memory: impact on sustainability and requirements for memory fabric design.

Regarding sustainability, the group discussed recent data published by industry [1] Operational cost of storage and memory, as well as of CPUs, are significant. Embodied cost associated with storage and memory are significant, whereas of CPUs not as much. The embodied costs of PNM are likely to be increased. Efficiency is important, but just energy efficiency (operational cost) is not sufficient. Operational costs are expected to go down substantially, for instance the hyperscalers are investing in offsetting these costs with renewables. However, embodied costs will remain important. It will be important to find ways to extend the lifetime on components, including of memory and storage, and to reuse them. Disaggregation solutions, powered by CXL, provide one way that can help in this regard [2].

Regarding the memory fabric design, the group first focused on contrasting current CXL-based memory fabric capabilities compared to other technologies (PCI, IB RDMA, NVLink, GPUDirect). The group then considered several LLM inference scenarios and worked through high level designs of fabric requirements that would be able to satisfy the data requirements of a high-end GPU system. The participants agreed such use-case driven approach provides a good way to drive the fabric design requirements and several of them planned to follow up on the initial conversation to derive more detailed requirements.

References

1. Wang et al. *Designing Cloud Servers for Lower Carbon*. ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA), Buenos Aires, Argentina, 2024, pp. 452-470, doi: 10.1109/ISCA59077.2024.00041.
2. Yuhong Zhong, Daniel S. Berger, Pantea Zardoshti, Enrique Saurez, Jacob Nelson, Antonis Psistakis, Joshua Fried, and Asaf Cidon. *My CXL Pool Obviates Your PCIe Switch*. In Proceedings of the 2025 Workshop on Hot Topics in Operating Systems (HotOS '25). Association for Computing Machinery, New York, NY, USA, 58–66. https://doi.org/10.1145/3713082.3730393

4.4 Report of the Working Group on Systems Support for Processing-Near-Memory

Ada Gavrilovska (Georgia Institute of Technology – Atlanta, US), Gustavo Alonso (ETH Zürich, CH), Birte Kristina Friesel (Universität Osnabrück, DE), Sudarsun Kannan (Rutgers University – Piscataway, US), Hoshik Kim (SK hynix – San Jose, US), Ilia Petrov (Hochschule Reutlingen, DE), Kai-Uwe Sattler (TU Ilmenau, DE), Kevin Skadron (University of Virginia – Charlottesville, US), Zeke Wang (Zhejiang University – Hangzhou, CN), and Chun Jason Xue (MBZUAI – Abu Dhabi, AE)

A working group on processing-near-memory (PNM) technologies gathered in one of the Day 2 breakout session to discuss the use cases, programming interfaces, and systems support for integration and efficient use.

Some part of the discussion was spent on differentiating PNM for processing-in-memory (PIM) technologies. PIM modifies the DRAM design, and/or operates at limited (bank) granularity. It's generally more challenging to integrate in systems. Multiple examples (e.g., vector-matrix dot product) show performance promise, but efficiency is not a clear win. It is well suited for examples where capabilities for flexible data access and data gather can be leveraged at the supported granularity, without the need to transform data (e.g., read/select from data records). In contrast, PNM adds computational capabilities at the channel level, like with CXL Computational Memory (CCM). It can be readily deployed. Some open question from industry include which specific functionality to integrate, are specialized cores or accelerators only adequate or is there a need for general purpose cores, and if so what kind.

Answering these questions requires consideration of use cases, and systems support for programming and integration of PNM in the OS and memory management stack. Several use cases were discussed.

Memory properties: PNM enhances the memory properties with new capabilities, such as for (de-)/compression, quantization, encryption/decryption, etc;

Data movement: PNM adds support for functionality common in data movement operations, such as for memory copying, initialization, etc. One tradeoff is that offloading these operations bypasses the CPU cache.

Data analytics: Existing industry prototypes already demonstrate the benefits of operations such as scan, filter, similarity, nearest neighbor, etc., common in data analytics applications.

LLM inference: Existing industry prototypes also demonstrate benefits for offloading select operations common in LLM inference; this is particularly useful for mobile or edge devices due to potential improvements in energy efficiency.

A simple approach to exposing the PNM capabilities is through use of optimized libraries, similarly to what has been done for other hardware offload and acceleration engines. An interface that seems to generalize well to different use case is that of a function call with a pointer to data. In addition to programming support, there is need for compiler or runtime support to navigate tradeoffs in terms of function placement (e.g., to offload or not). In addition, even for fixed function computations, there may be need for data reshaping, to deal with specific requirements for data layout/data format support, for instance to offset to appropriate data element in record, or when computation requires aggregation across memory channels (e.g., pointer to other memory, traversal of graph data structures, etc.). Finally, the programmability raises decisions on the control-plane interface, i.e., how does one program PNM. The group discussed whether eBPF provides a sufficiently adequate extension model, and observed additional work may be required.

Another set of important design decisions for PNM rests with the integration with the operating system, in terms of addressing, access control, reliance on TLB and MMU hardware, etc. One important question that must be considered is whether computational capabilities be integrated with all memory or only some parts of it. In the latter case, new support is needed to present a united view of memory with different capabilities (including local DRAM + CCM).

The working group discussed some of the possible design points and associated tradeoffs. Some of the breakout sessions in the later part of the seminar focused on a deep dive into some of these aspects (e.g., programming APIs and virtual memory integration).

4.5 Report of the Working Group on Drivers for Disruptive Memory Technologies

Jana Giceva (TU München – Garching, DE), Oana Balmau (McGill University – Montréal, CA), Jerónimo Castrillón-Mazo (TU Dresden, DE), Thaleia Dimitra Doudali (IMDEA Software Institute – Madrid, ES), Tajana Simunic Rosing (University of California – San Diego, US), Nandita Vijaykumar (University of Toronto, CA), and Huanchen Zhang (Tsinghua University – Beijing, CN)

The working group evaluated the application-level drivers from various domains to determine how they may influence the design and benefit from the development of future disruptive memory technologies. We discussed various aspects from performance bottlenecks across a diverse set of domains to current painful points and limitations. The main pursuit was identifying a common set of challenges. This short summary structures our discussion in two parts: the first one characterizes the most common domain workloads and their memory-related challenges, and the second one then tries to map some of these challenges to specific memory technologies that could help address them.

4.5.1 Domain Workload Analysis

The group discussed several classes of applications and domains (machine and deep learning, computer vision, data management, and genomics.) trying to identifying common and domain-specific memory bottlenecks:

Deep Learning and Large Language Models (LLMs)

- Training is in general quite memory-intensive and demands high-bandwidth.
- Inference can sometimes involve extremely large key-value (KV) caches, particularly in long-context or multi-session workloads.

Databases

- Vector databases and retrieval-augmented generation (RAG) systems, may require frequent updates to vector indices
- HTAP (Hybrid Transactional/Analytical Processing) systems need efficient support for in-memory updates without compromising snapshot consistency for analytics.
- Housekeeping tasks like index building, data partitioning, garbage collection, generating consistent snapshots from the logs, etc. are quite memory intensive and should not have to burn user-facing CPU cycles.

Genomics, Proteomics and Metabolomics

- Omic workloads are characterized by high-dimensional, large-volume datasets across heterogeneous formats.
- Preprocessing stages are costly and can last a few hours or even days.
- Prior to database search, the data needs to be clustered, which is currently impossible on regular basis due to large performance overhead
- Database search is massively parallel, focused on exact and nearest neighbour pattern match. For genomics alignment, dynamic programming and/or graph based methods are needed as well. For proteomics and metabolomics, extremely fast nearest neighbor pattern match is key.

Visual Computing

- 3D vision and interactive applications are sensitive to memory latency, especially for real-time training and inference.
- Generative AI for visual data (e.g., scene generation) requires fast, low-latency access to long sequences of data.

Graph Applications

- Graph traversal, pattern matching, and dynamic analytics require memory systems capable of handling irregular accesses, and with frequent updates also a consistent snapshot support.

4.5.2 Technology Mapping and Opportunities

In the second part of the session, the group discussed how emerging memory technologies could address some of the identified challenges:

CXL (Compute Express Link)

Enables memory pooling and scale-up architectures. These could be of general use for any domain that is constrained on the memory capacity (e.g., large scale data analytics, graph processing, etc.). With memory-extensions provided by CXL one can also support larger KV caches for LLM inference (10s of TBs possible), potentially extending down into storage layers. May alleviate memory bottlenecks in long-context generative workloads, including video generation and long sequence modeling in AI applications. The group did not identify many use-cases for coherency, or rather a need for coherency beyond small areas that are needed for coordination (e.g., cross-partition transactions in distributed databases).

Processing-in-Memory (PIM), Near-Data Processing (NDP), and Near-Storage Computing

These technologies are considered a great match for workloads that aim to reduce data movement. Some specific tasks that we identified may benefit the most are the following:

- Pattern matching (e.g., database search, nearest neighbor queries),
- Clustering (e.g. prior to database search for omics disciplines, for unsupervised learning)
- Irregular compute (pointer chasing),
- Housekeeping tasks (e.g., garbage collection, compaction, ML preprocessing),
- Sensor data summarization,
- Memory-bound operations like hash tables and joins in databases

Useful for omics disciplines, graph analytics, and databases for restructuring and lightweight computation close to data.

One potential concern was raised by the data management side, that accelerating individual tasks with PIM/PNM may have diminishing returns for workloads that exhibit a frequent set of accesses to common data regions and may in the long run benefit from caching. So the interplay between caching and processing close to where the original data resides may need to be explored with care.

Checkpointing and Persistent Memory

Managed Retention Memory (MRM) was brought up as a novel approach to simplifying checkpointing by enabling memory regions with flexible durability semantics.

4.6 Report of the Working Group on Programming Models for Disruptive Memory Technologies

Tianzheng Wang (Simon Fraser University – Burnaby, CA), Jerónimo Castrillón-Mazo (TU Dresden, DE), Birte Kristina Friesel (Universität Osnabrück, DE), Jana Giceva (TU München – Garching, DE), Yu Hua (Huazhong University of Science & Technology, CN), Kimberly Keeton (Google LLC – South San Francisco, US), Wolfgang Lehner (TU Dresden, DE), Kai-Uwe Sattler (TU Ilmenau, DE), Kevin Skadron (University of Virginia – Charlottesville, US), Olaf Spinczyk (Universität Osnabrück, DE), and Huanchen Zhang (Tsinghua University – Beijing, CN)

Traditional memory programming models (e.g., malloc, memcpy, madvise) have provided a robust abstraction for decades of general purpose computing. However, emerging memory technologies – including processing-in/near-memory, persistent memory or memory disaggregation via novel coherent protocols like CXL – expose fundamentally different performance characteristics (latency, bandwidth, granularity of accesses) and capabilities (coherency, ordering guarantees, persistency or even compute). These break the long-standing assumptions about address-spaces, pointer validity, update semantics, fault handling and expose a world where data and memory need to be decoupled from the traditional process-centric view/abstraction of a machine.

This report summarizes a recent discussion aiming to define new abstraction layers that can better support the development (and maintenance) of complex data-intensive applications (e.g., database systems and large-scale data processing) on top of modern memory technologies.

4.6.1 Limitations of existing programming models

Current models provide only rudimentary support for the emerging memory landscape:

- Device-specific features: traditional interface APIs like malloc do not distinguish between memories with or without compute capabilities, nor do they express memory locations (local, remote, disaggregated), volatility or performance (latency, bandwidth).
- Device-specific constraints: some compute capabilities near/in memory may not share the global address space and existing memory semantics do not translate easily. For example, UPMEM's PIM requires manual partitioning and data movement.
- Disaggregated memory challenges: explicit data movement and non-cache coherence can be challenging to work with. It is also not clear how to capture the fault model and the guarantees (atomicity, ordering) that can be supported.
- Inflexible abstractions: lack the flexibility to express requirements and/or to allow applications to adapt and fully leverage the features of the emerging technologies.
- Lifetime of allocated objects: may not fit the traditional process-based model. For example the concept of a pointer in heterogeneous memory systems is complex analogous to persistent memory and the need for pointer swizzling, as virtual addresses might not be consistent after reboots or remapping.

4.6.2 Towards new-abstractions and programming models

We argue that now is a good time to embrace a declarative model, in which application developers specify what they want, and the runtime/compiler decides how and where it should run. This allows for a separation of concerns related to the execution runtime, and decouples the details of technology's capabilities and its relevant optimizations from the development of the application logic.

Similar ideas have been explored in the past for partitioned global address space (e.g., X10) or Intel's library/runtime for TBB which defers the placement/scheduling to the runtime.

We expect that this should enable:

1. better optimization across heterogeneous compute/memory sub-systems;
2. decoupling of the device-specific logic from the high-level application code;
3. simplify the programming for future, unknown architectures.

To achieve that we propose a two layer approach, where:

- A logical region-based abstraction layer is used by application developers to declaratively mark their requirements (e.g., persistent, encrypted, high bandwidth, coherent, etc.).
- A calibration layer captures the device characteristics (e.g., through reading through the specs and benchmarking) and stores the results in a catalog consumed by optimizers, compilers, and the runtime system's scheduler.

Upon a short analysis and discussion we concluded that a dataflow programming model can be a good fit for many data-intensive applications such as databases, streaming systems, big-data frameworks like Spark, and possibly also other ML-systems.

The key idea behind such a dataflow programming model is to explicitly model the (1) compute operators (kernels) and the (2) data transfers into the dataflow graph. One can mark the pipelines of operations that can be potentially merged (operator fusion) if the data-transfer allows for streaming the data. In case of pipeline-breakers, one can explicitly mark the state (e.g., the hash-table).

This also can naturally support annotations, where developers can annotate performance/security goals of their program while the system can infer other requirements (e.g., persistence, memory locality, coherence, etc.).

Once the annotations are in place, the optimizer/runtime can select operators and memory regions using the cost model from the catalog of the calibration layer. For example, if the operational intensity is low it may choose to run the operation within a PIM/PNM device. Once the state and data transfer is explicit we can also reason about required data transformations.

4.6.3 Taxonomy of Memory Properties

To facilitate reasoning about the calibration layer and the type of properties that application developers may need to annotate (or the compilation framework to infer), we also thought of a list of taxonomy of memory properties:

Property	Examples
Performance	(latency, bandwidth)
Volatility	(non-volatile, volatile)
Active/Passive	(active (PIM) vs normal DRAM)
Location	local, NUMA, remote (CXL or RDMA)
Granularity	Byte-addressable vs. page-based (and size of packets/pages)
Coherency	(coherent vs. non-coherent)
Security	(encrypted or not)
Compression	(type of compression)
Ordering	(strong/weak memory model, explicit fencing, causal consistency, etc.)
Fault model	(atomic execution, exactly once, transient or partial fault, etc.)

The specific technologies that were discussed were:

Processing-in-memory (PIM): advantages for data processing is that it will reduce data movement and may result in better performance for workloads with sequential access patterns. The limitations we see is that it may not be beneficial for workloads that have strong data reuse and/or non-sequential patterns. Examples: UPMEM requires explicit data movements; lacks global address space; in the cloud set-up pushing predicate evaluation down to storage (e.g., with AWS S3-select) had problems with caching, and on a workload-level over longer period of time there were diminishing returns for the one-off performance improvement.

Memory extensions/pooling (CXL): offers good latency, coherency, but may introduce pointer consistency issues (what if a link fails?). It is not clear how much of the techniques and insights from RDMA will play a role in the context of CXL and how it should be used. For example, many benefits for remote memory pooling (over RDMA) can already be seen as reported by production systems like Google's BigQuery shuffling layer. At the same time, the lack of ordering guarantees on RDMA writes showed that it is impractical for synchronization purposes. How much of these can be avoided (or be customized/controlled) with novel protocols like CXL (or photonics) is to be seen. Nevertheless, the broader systems community already explored some of these questions in the context of far-memory and one needs to check them to see how it may influence the design of the proposed programming model.

Participants

- Gustavo Alonso
ETH Zürich, CH
- Oana Balmau
McGill University – Montréal, CA
- Antonio Barbalace
University of Edinburgh, GB
- Frank Bellosa
KIT – Karlsruher Institut für Technologie, DE
- Daniel Berger
Microsoft – Redmond, US
- Jerónimo Castrillón-Mazo
TU Dresden, DE
- Haibo Chen
Shanghai Jiao Tong University, CN
- Jian-Jia Chen
TU Dortmund, DE
- David Cohen
Solidigm – Santa Fe, US
- Christian Dietrich
TU Braunschweig, DE
- Thaleia Dimitra Doudali
IMDEA Software Institute – Madrid, ES
- Michal Friedman
ETH Zürich, CH
- Birte Kristina Friesel
Universität Osnabrück, DE
- Ada Gavrilovska
Georgia Institute of Technology – Atlanta, US
- Jana Giceva
TU München – Garching, DE
- Boris Grot
University of Edinburgh, GB
- Timo Hönig
Ruhr-Universität Bochum, DE
- Yu Hua
Huazhong University of Science & Technology – Wuhan, CN
- Sudarsun Kannan
Rutgers University – Piscataway, US
- Sanidhya Kashyap
EPFL – Lausanne, CH
- Kimberly Keeton
Google LLC – South San Francisco, US
- Hoshik Kim
SK hynix – San Jose, US
- Wolfgang Lehner
TU Dresden, DE
- Alberto Lerner
University of Fribourg, CH
- Till Miemietz
Barkhausen Institut – Dresden, DE
- Onur Mutlu
ETH Zürich, CH
- Ilia Petrov
Hochschule Reutlingen, DE
- Tilmann Rabl
Hasso-Plattner-Institut, Universität Potsdam, DE
- Tajana Simunic Rosing
University of California – San Diego, US
- Kai-Uwe Sattler
TU Ilmenau, DE
- Wolfgang Schröder-Preikschat
Universität Erlangen-Nürnberg, DE
- Kevin Skadron
University of Virginia – Charlottesville, US
- Olaf Spinczyk
Universität Osnabrück, DE
- Michael Swift
University of Wisconsin-Madison, US
- Jürgen Teich
Universität Erlangen-Nürnberg, DE
- Nandita Vijaykumar
University of Toronto, CA
- Tianzheng Wang
Simon Fraser University – Burnaby, CA
- Zeke Wang
Zhejiang University – Hangzhou, CN
- Thomas Willhalm
Intel Deutschland GmbH – Feldkirchen, DE
- Youjip Won
KAIST – Daejeon, KR
- Chun Jason Xue
MBZUAI – Abu Dhabi, AE
- Huanchen Zhang
Tsinghua University – Beijing, CN

Report from Dagstuhl Seminar 25152

Multi-Faceted Visual Process Mining and Analytics

Claudio Di Ciccio*[1], Pnina Soffer*[2], Christian Tominski*[3], Katerina Vrotsou*[4], and Giovanni Meroni†[5]

1 **Utrecht University, NL.** c.diciccio@uu.nl
2 **University of Haifa, IL.** spnina@is.haifa.ac.il
3 **Universität Rostock, DE.** christian.tominski@uni-rostock.de
4 **Linköping University, SE.** katerina.vrotsou@liu.se
5 **Technical University of Denmark – Lyngby, DK.** giom@dtu.dk

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 25152 "Multi-Faceted Visual Process Mining and Analytics". The seminar brought together experts from the process mining (PM) community and the visual analytics (VA) community to strengthen the identified synergies of both fields and identify further novel and promising research directions. A particular focus of the seminar was on the challenges arising from the multi-faceted nature of processes and the multi-faceted data to be investigated. The relevant facets include time (when do processes happen), space (where do processes happen), topology (how are processes connected), object centricity (how are processes characterized), uncertainty (what are we unsure about), analytic provenance (how did we obtain our knowledge), and more. This report deals with challenges related to these different data facets, individually and in combination. As a general principle, VA methods are advocated to be an integral part of all phases of the PM process to facilitate a comprehensive multi-faceted data exploration, hypothesis generation, and presentation of results. More concretely, the discussions revolve around several aspects at the crossroads of the two disciplines workflows, including the data facets under analysis, the human factors at play, the catalog of aided tasks, novel combinations of visual, interactive, and computational methods, as well as integration, scalability, and general applicability of the devised solutions.

Seminar April 6–11, 2025 – https://www.dagstuhl.de/25152
2012 ACM Subject Classification Applied computing → Business process management; Human-centered computing → Visualization
Keywords and phrases human in the loop, process mining, visual analytics
Digital Object Identifier 10.4230/DagRep.15.4.28

1 Executive Summary

Claudio Di Ciccio (Utrecht University, NL)
Pnina Soffer (University of Haifa, IL)
Christian Tominski (Universität Rostock, DE)
Katerina Vrotsou (Linköping University, SE)

This Dagstuhl Seminar "Multi-Faceted Visual Process Mining and Analytics" (25152) brought together 27 experts from the Process Mining (PM) and Visual Analytics (VA) communities to Schloss Dagstuhl to work on the challenges arising from the multi-faceted nature of

* Editor / Organizer
† Editorial Assistant / Collector

processes and corresponding event log data. The seminar was held from April 6 to 11, 2025 as a follow-up seminar to Dagstuhl Seminar 23271, “Humans in the (Process) Mines” (https://www.dagstuhl.de/23271).

PM is a rapidly growing discipline blending machine learning and data mining concepts with ideas taken from the field of business process management. PM studies event log data to support business process execution for a variety of tasks, from the automated discovery of graphical process models to operational support. VA is a multidisciplinary approach that combines interactive, visual, and analytical methods to make complex data comprehensible, facilitate new insights, and enable knowledge discovery. VA research happens at the intersection of data mining and knowledge discovery, information visualization, human-computer interaction, and cognitive science.

The focus of this seminar was on discussing and investigating the challenges of multi-faceted visual process mining and analytics. The relevant facets include time (when do processes happen?), space (where do processes happen?), topology (how are processes connected?), object centricity (how are processes characterized?), uncertainty (what are we unsure about?), analytic provenance (how did we obtain our knowledge?), and more. The seminar discussed approaches to deal with and gain insight into these different data facets, individually and in combination, and outlined novel ideas and promising directions for future research to further strengthen the synergies of PM and VA.

The seminar started with a general introduction by the seminar organizers. In addition to presenting the general goals of the seminar, the organizers also reflected on the impressive outcomes of the previous seminar, which initiated the collaboration between PM and VA. The general introduction was followed by the introduction of the seminar participants, who briefly stated their background, expertise, and expectations in the seminar.

The first day of the seminar featured a series of expert presentations, mainly introducing the participants to key concepts and methods related to the different data facets. Gennady Andrienko gave an overview of VA for spatio-temporal data, highlighting the need for dedicated visual representations and the aspect of spatial and temporal scale. Hans-Jörg Schulz focused on the topology facet by introducing VA methods for visually analyzing graph structures (or networks). He introduced fundamental network visualization principles and also showcased examples of how the data facets of space and time can be combined with network visualization. The important issues of data quality and uncertainty were presented in the talk by Silvia Miksch. She emphasized different types of uncertainty and data quality problems for temporal, spatial, and network data. She also introduced basic strategies for visually representing uncertainty and discussed insights from user experiments. Claudio Di Ciccio turned the participants’ attention towards an object-centric perspective of processes, where processes are defined through multi-valued entities and relations among them. He introduced the object-centric event data (OCED) meta model as a means to describe processes in an object-centric manner. Finally, Francesca Zerbato gave a detailed introduction to the actual process of PM, which involves various iterative steps, each generating different results and artifacts. She also highlighted the importance of integrating provenance and corresponding analytical tools into the PM process for informed decision-making.

The theoretical aspects conveyed by the talks were supplemented with practical hands-on challenges based on two multi-faceted data sets. The organizers presented a data set from the VAST challenge series addressing a fictitious scenario related to involving people in urban planning and shaping social communities. A second data set was concerned with processes from truck shipment logistics. Both data sets illustrated the richness of multi-faceted processes and the event logs that they create, and indicated the challenges involved in exploring, analyzing, and understanding such processes.

Figure 1 Whiteboards with investigation topics and preferences following the brainstorm session.

Talks and hands-on challenges were followed by discussions and ideation toward working group formation. The seminar participants brainstormed potential ideas and collected them on the whiteboard (see Figure 1). From a list of about twenty ideas for working groups, five promising topics were merged and crystallized based on relevance, potential impact, and participant preferences. Eventually, the five groups worked on the following topics.

Group A: Towards Improving Processes Using Multi-Faceted Visual Analysis
Group B: Progressive Visual Analytics for Streaming Process Mining – VESPA
Group C: Interactivity: Visual Feedback and Feedforward for Process Exploration
Group D: Coordinated Projections: A New Approach to Multi-Faceted Process Exploration
Group E: Towards Visual Process Analytics for Process Ecosystems

Overall, the working groups had about six sessions to work on their topics. During intermediate group reports, all participants had the opportunity to provide feedback and contribute their expertise to all working groups. Moreover, lightning talks were given on specific aspects that arose during the seminar. Natalia Andrienko provided an overview of storyline visualizations for the analysis of event logs, highlighting their use to track the unfolding of processes based on the business objects evolution over time. Philipp Koytek held a demonstration of the object-centric process mining functionalities in the Celonis suite, with a special focus on visualization and user-guided exploration. Iris Beerepoot presented a novel dataset for the seminar attendees to explore pertaining to personal information management, with three years worth of data records tracking and categorizing knowledge workers' tasks at their workstation.

The final day of the seminar included the presentation of the results of the working groups and set the stage for the official closing of the seminar. The results of the working groups can be read on the following pages of this report. Although the seminar was held in a smaller format with fewer participants (compared to the previous seminars in the series), the reports from the working groups present an impressive amount of creative new ideas for combining PM and VA approaches. Given the success of the collaboration of PM and VA experts, also beyond Dagstuhl, the participants suggested and agreed to submit a proposal for continuing the series of Dagstuhl seminars on combining PM and VA. The planned follow-up seminar shall reflect the fruitful collaboration by merging PM and VA to a new unified research area of Visual Process Analytics (VPA).

2 Table of Contents

3 Overview of Talks

3.1 Visual Analytics of Spatio-Temporal Data

Gennady Andrienko (Fraunhofer IAIS – Sankt Augustin, DE)

In my presentation entitled "Visual analytics of spatio-temporal data", I've discussed the specifics of space and time and their implication on data analysis. Further on, I presented the main types of spatio-temporal data (events, time series, trajectories) and possible transformations between these representations. I presented major approaches to analysis of spatio-temporal data, including topic modelling, and proposed ideas for adapting these methods for process mining tasks.

3.2 A Primer on Network Visualization

Hans-Jörg Schulz (Aarhus University, DK)

Main reference Steffen Hadlak, Heidrun Schumann, Hans-Jörg Schulz: "A Survey of Multi-faceted Graph Visualization", in Proc. of the 17th Eurographics Conference on Visualization, EuroVis 2015 – State of the Art Reports, Cagliari, Sardinia, Italy, May 25-29, 2015, pp. 1–20, Eurographics Association, 2015.
URL https://doi.org/10.2312/EUROVISSTAR.20151109

Network visualizations form an important pillar of visual process analytics, as on one hand many processes can be directly captured in a graph representation (e.g., biological processes as pathways or software processes as UML diagrams), but also indirectly by their effects (e.g., migration flows indicating underlying socio-economic and geopolitical processes). This short primer on network visualization will give an overview of the various ways in which networks can be diagrammatically depicted – including networks with additional attributes and facets, such as geospatial networks or dynamic networks. A collection of the most important overview articles and surveys on the topic rounds off this short presentation.

3.3 Visual Analytics: Data Uncertainty & Quality

Silvia Miksch (TU Wien, AT)

Main reference Theresia Gschwandtner, Markus Bögl, Paolo Federico, Silvia Miksch: "Visual Encodings of Temporal Uncertainty: A Comparative User Study", IEEE Trans. Vis. Comput. Graph., Vol. 22(1), pp. 539–548, 2016.
URL https://doi.org/10.1109/TVCG.2015.2467752

Data uncertainty and quality are critical components of visual process analytics, directly influencing the validity, interpretability, trustworthiness, and reliability of analytical processes and outcomes. In this presentation, I will present a conceptual and methodological overview of data uncertainty and quality focusing on sources, taxonomies, visual encoding, temporal and spatial dimensions, models and model comparison, parameter space exploration as well as network visualization. A discussion will conclude with an examination of the challenges and opportunities associated with these approaches.

References

1 Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of Time-Oriented Data, Second Edition.* Springer, 2023.

2 Velitchko Filipov, Alessio Arleo, and Silvia Miksch. Are We There Yet? A Roadmap of Network Visualization from Surveys to Task Taxonomies. *Computer Graphics Forum*, 42(6), 2023.

3 Theresia Gschwandtnei, Markus Bögl, Paolo Federico, and Silvia Miksch. Visual Encodings of Temporal Uncertainty: A Comparative User Study. *IEEE Transactions On Visualization And Computer Graphics.* 22(1): 539-548, 2016.

3.4 Object-Centric Event Data

Claudio Di Ciccio (Utrecht University, NL)

Main reference Dirk Fahland, Marco Montali, Julian Lebherz, Wil M. P. van der Aalst, Maarten van Asseldonk, Peter Blank, Lien Bosmans, Marcus Brenscheidt, Claudio Di Ciccio, Andrea Delgado, Daniel Calegari, Jari Peeperkorn, Eric Verbeek, Lotte Vugs, Moe Thandar Wynn: "Towards a Simple and Extensible Standard for Object-Centric Event Data (OCED) – Core Model, Design Space, and Lessons Learned", CoRR, Vol. abs/2410.14495, 2024.
URL https://doi.org/10.48550/ARXIV.2410.14495

Recent trends in process mining research are evidencing a paradigm shift from the classical activity-centric approach, wherein the spotlight is on conducted tasks and events reporting their execution. Lately, the community has recognised the need to give prominence to objects that those activities create, observe, or alter. Hence the name of the new stream: Object-centric process mining. Accordingly, the IEEE Task Force on Process Mining has begun a procedure to establish a new structure and format to record event logs' information, namely Object Centric Event Data (OCED, https://www.tf-pm.org/resources/oced-standard) standard [1]. The talk revisits the steps that led to the current OCED meta-model, illustrates its rationale, and concludes with a call to action for visual analytics research to join the challenge of making sense of this inherently multi-faceted information source for process mining.

References

1 Dirk Fahland, Marco Montali, Julian Lebherz, Wil M. P. van der Aalst, Maarten van Asseldonk, Peter Blank, Lien Bosmans, Marcus Brenscheidt, Claudio Di Ciccio, Andrea Delgado, Daniel Calegari, Jari Peeperkorn, Eric Verbeek, Lotte Vugs, Moe Thandar Wynn. *Towards a Simple and Extensible Standard for Object-Centric Event Data (OCED) – Core Model, Design Space, and Lessons Learned.* CoRR abs/2410.14495, 2024

3.5 The Process of Process Mining and Provenance

Francesca Zerbato (TU Eindhoven, NL)

Joint work of Francesca Zerbato, Andrea Burattin, Hagen Völzer, Paul Nelson Becker, Elia Boscaini, Barbara Weber
Main reference Francesca Zerbato, Andrea Burattin, Hagen Völzer, Paul Nelson Becker, Elia Boscaini, Barbara Weber: "Supporting Provenance and Data Awareness in Exploratory Process Mining", in Proc. of the Advanced Information Systems Engineering – 35th International Conference, CAiSE 2023, Zaragoza, Spain, June 12-16, 2023, Proceedings, Lecture Notes in Computer Science, Vol. 13901, pp. 454–470, Springer, 2023.
URL https://doi.org/10.1007/978-3-031-34560-9_27

The process of process mining is emergent, insight-driven, and knowledge-intensive. Process analysts engage in iterative steps, generating diverse results and artifacts that must be validated and reproduced for purposes such as storytelling and auditing. However, current process mining methods and tools offer limited support for managing these evolving workflows. In this talk, we explore how integrated provenance and data views can support analysts by enabling reflection in action, informed decision-making, and traceability of results back to raw data. We conclude with a call to the process mining and visual analytics communities to advance this area by addressing key questions: What types of provenance are most useful for process analysts? How can we make provenance information accessible and actionable? And how should it be effectively visualized?

References

1 Francesca Zerbato, Andrea Burattin, Hagen Völzer, Paul Nelson Becker, Elia Boscaini, and Barbara Weber. Supporting provenance and data awareness in exploratory process mining. In *International Conference on Advanced Information Systems Engineering*, pages 454–470. Springer, 2023.

3.6 Storyline Visualizations for Object-Oriented Process Analysis

Natalia V. Andrienko (Fraunhofer IAIS – Sankt Augustin, DE)

In my lightning talk, I addressed the challenge of object-oriented process analysis, which requires not only tracing process flows but also understanding the roles and interactions of the objects involved. I proposed that storyline visualizations, which are commonly used to depict evolving relationships between entities over time, could possibly serve for this purpose. To illustrate this, I presented examples of storyline visualizations from recent research papers. Additionally, I mentioned the potential of the Marey chart, originally developed for visualizing train timetables, as another suitable approach for representing object lifelines and their involvement in process events. These visual approaches may support better comprehension of complex, multi-object process dynamics.

4 Working groups

4.1 Towards Improving Processes Using Multi-Faceted Visual Analysis

Zhicheng Liu (University of Maryland – College Park, US), Wolfgang Aigner (FH – St. Pölten, AT), Lena Cibulski (Universität Rostock, DE), Marie-Christin Häge (Universität Mannheim, DE), and Pnina Soffer (University of Haifa, IL)

Motivation

Improving processes has for long been one of the aims of process mining analysis [3]. In this report, we explore the ways by which multi-faceted visual analysis of process data can contribute to process improvement. To design meaningful visualizations that solve real-world problems such as process improvement, we need to characterize the goals and tasks that are to be accomplished and thus frame the visualization use [2].

Description of Method and Dataset

We approach this by breaking down the task of process improvement into six sub-tasks (see Figure 2). Following the task abstraction by Tominski and Schumann [1], we then characterize each sub-task by describing four key aspects: goals (i.e., the overarching intent), analytical questions (i.e., what is to be investigated), targets (i.e., which specific data we need to look at to complete that sub-task), and means (i.e., how a sub-task might be performed). We demonstrate exemplary instantiations of the six sub-tasks by providing a case study of the Logistic dataset 2014[1]. The data set includes information about trucks delivering shipments across Europe between 23.03.2014 and 30.03.2014. Besides basic event information (event type and timestamps) it also includes spatial (latitude, longitude, and mileage) and speed information, which can be utilized for a multi-faceted analysis. We extracted the process flow from the data using Disco[2] and depicted it using a process map (see Figure 3), but the insights that can be drawn from it are very limited.

In what follows, we describe a case study based on the Logistics data set to demonstrate each of our identified sub-tasks and its outcomes.

Case Study: Truck Shipment Log

1. Identify *KPIs*
   While different KPIs can be identified for this process, we note that these apply to three main facets:
   - *Time* (as an absolute measure, with respect to the distance passed, or considering an agreed upon delivery time)
   - *Distance* (as an approximation of fuel consumption)
   - *Compliance* (with respect to defined company policies or external regulations)
2. Identify undesirable behavior with respect to identified *KPIs*
   - *Compliance*. Shipments exceeding reference values as dictated by regulations, e.g., driving times longer than allowed (see Figure 4).

[1] https://drive.google.com/drive/folders/17F94erxk4KveMpKbnXCEOwBbhnUQkRzI?usp=sharing
[2] https://fluxicon.com/disco/

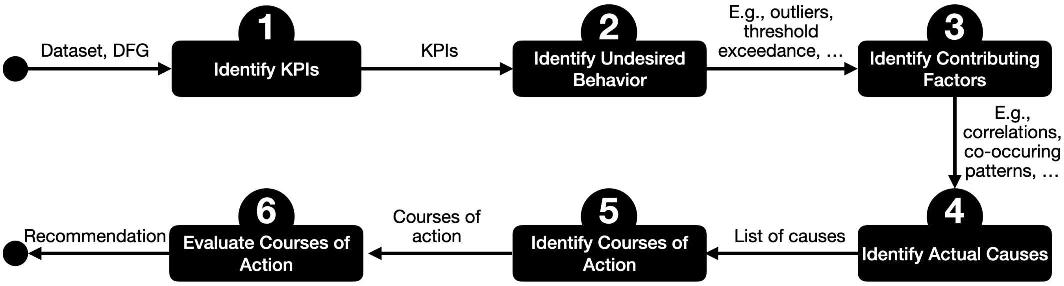


Figure 2 We break down the task of process improvement into six sub-tasks. The central outcome of each sub-task serves as input for the following sub-task. The end result is a recommendation of a course of action that represents the most promising process improvement opportunity.

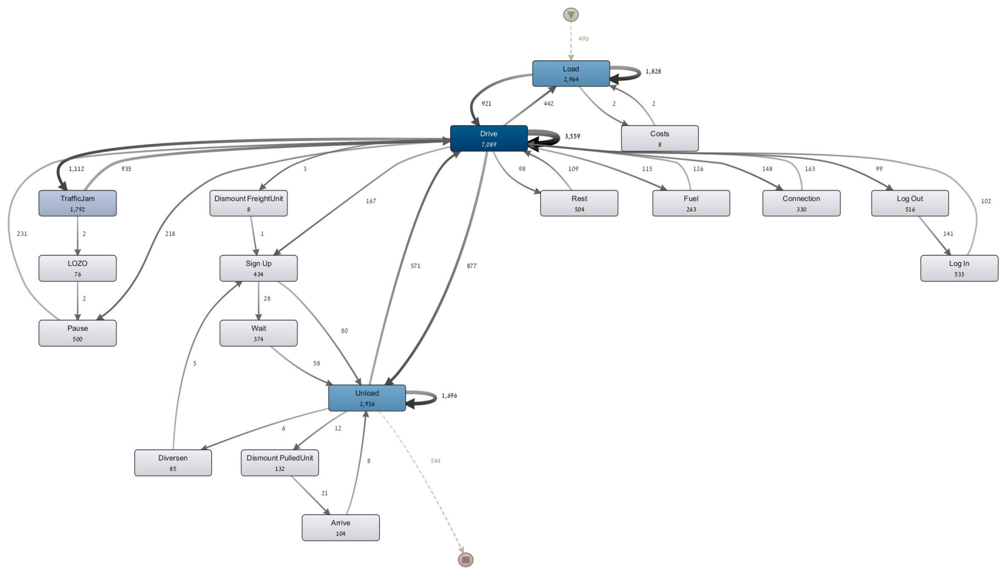


Figure 3 A process map of the logistic process underlying the analyzed truck shipment data, showing a directly-follows graph of the activities and their frequencies.

- *Time and Distance.* (Statistical) outliers with respect to distance and duration (see Figure 5).
- *Compliance.* Routes crossing regions in a certain time period, e.g., routes should go through Paris on Sundays (see Figure 6).
- *Time.* Shipments that arrive at the destination after the estimated time of arrival.

3. Identify contributing factors, i.e., potential explanations for identified undesired behavior
   a. What activities are associated with cases that exhibit long driving hours, i.e., occur in the same shipment? Answer: we find that cases with long driving hours contain time periods of low to zero speed → Possible explanation: lower speed due to traffic jam(s) needs to be compensated by longer driving hours to still arrive at the destination on time
   b. What activities are associated with cases that exhibit long driving hours, i.e., occur in the same shipment? Answer: we find that low speeds occur more frequently shortly before a break is due in cases with long driving hours than with other cases → Possible explanation: being stuck in a traffic jam does not allow drivers to take a break

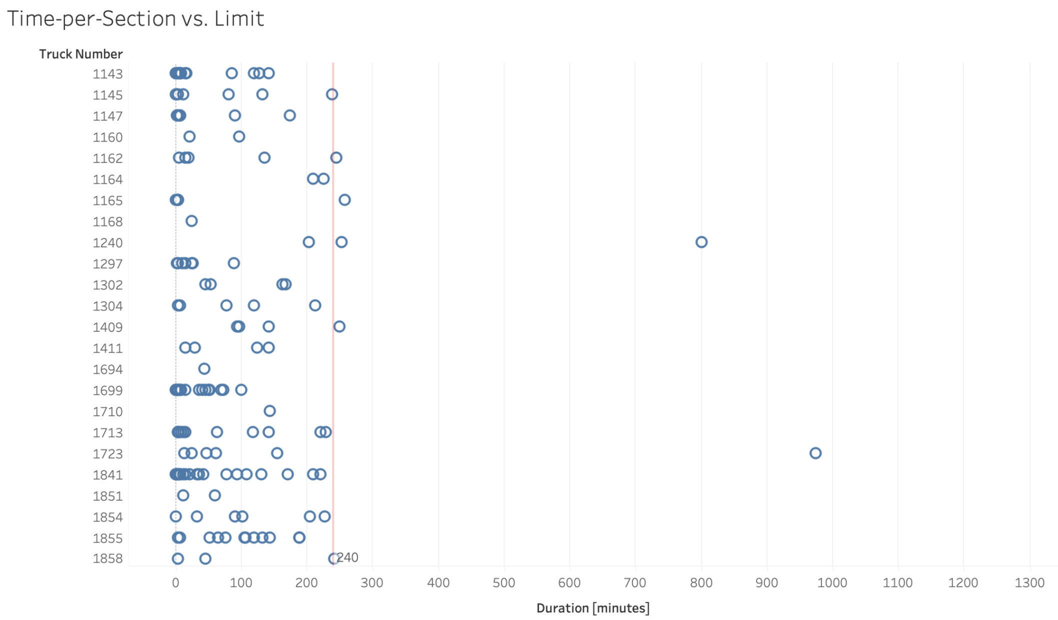


Figure 4 This plot depicts the durations of each driving section that a truck's shipment route is composed of. The red line reflects the regulation that drivers are not allowed to drive more than four hours (i.e., 240 min.) non-stop. Durations to the right of this line thus represent sections of a shipment route that violate the regulations.

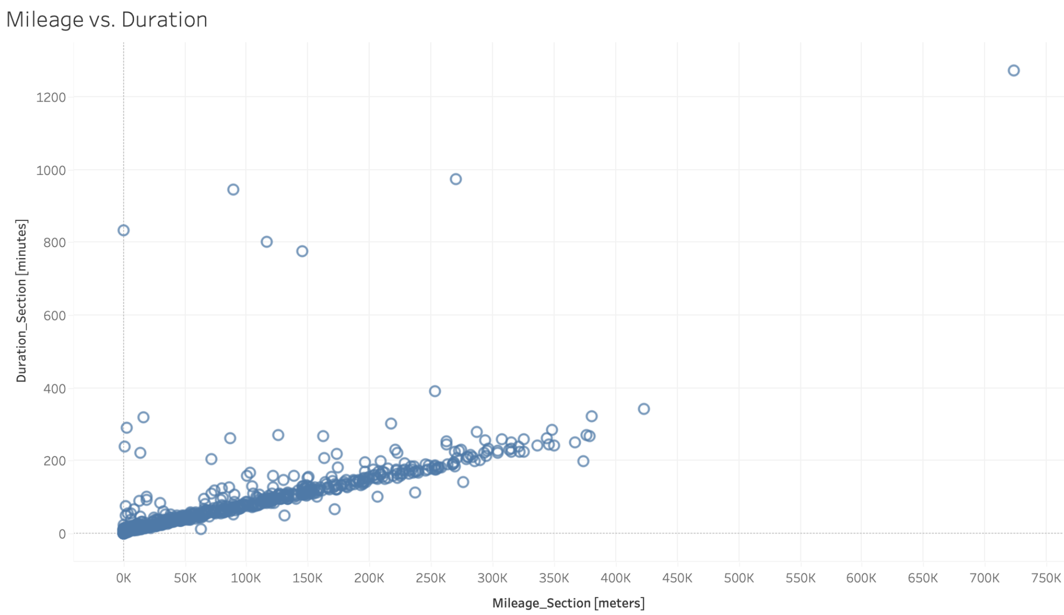


Figure 5 This scatter plot depicts driving sections (not entire shipment routes) over distance driven and duration. Most driving sections follow a linear relationship. However, five driving sections deviate from that in being rather short in terms of distance but still exhibiting significant driving time. Similarly, one driving section (in the upper right corner) deviates in showing exceptional distance as well as duration.

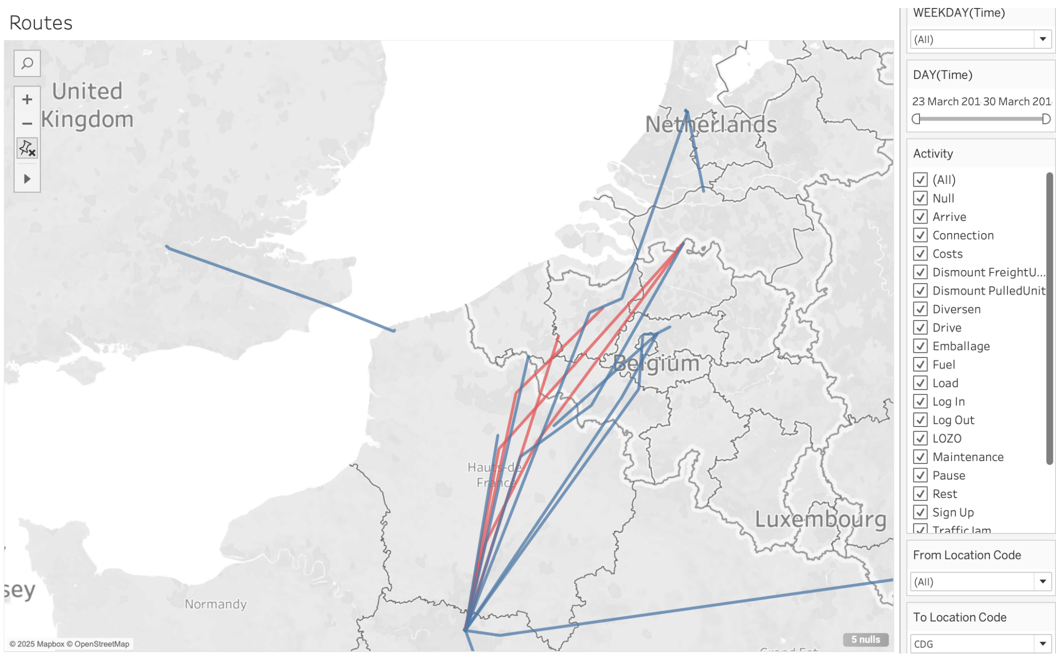


Figure 6 This map depicts all shipments ending in Paris. Routes that enter Paris during prohibited time periods according to heavy traffic bans (i.e., on Sundays) are highlighted in red.

c. What activities are associated with cases that exhibit delayed shipment? Answer: we find that delayed shipments are associated with time periods exhibiting zero to low speed → Possible explanations: 1) traffic jam or 2) serious breakdown of the truck
   - 1): Traffic jams can occur multiple times along the shipment route
   - 2): A serious breakdown is unlikely to occur multiple times along the same shipment route

d. What activities are associated with cases that exhibit delayed shipment? Answer: we find that delayed shipments are associated with exceptionally large distances → Possible explanations: driver took a detour to 1) avoid closed roads, 2) avoid steep roads during a snow storm, or 3) avoid forbidden roads (regulations)
shipment route numbers with large distances: 100001081854, 100001081380, 100001084371

4. Identify actual causes
   - 3a and 3b: Do the time periods exhibiting low speeds correspond to external reports of traffic jams?
   - 3c:
     - 1) Does the data show multiple time periods with low speed? If yes, this hints at traffic jam being the root cause, if not, this hints at a potential breakdown
     - 2) Is there a bill documentation available that reports a repair during the shipment?
   - 3d:
     - 1) Look up official information about road closings during the respective time periods
     - 2) Collect external data about the slope of roads as well as weather (temperature or snow falling). Was the shipment route passing steep road sections during snowfall (see Figure 7)?
     - 3) Collect information about which regions cannot be passed during certain time frames, e.g., Paris on Sundays between 10pm and midnight or Mondays between 6am and 10am.

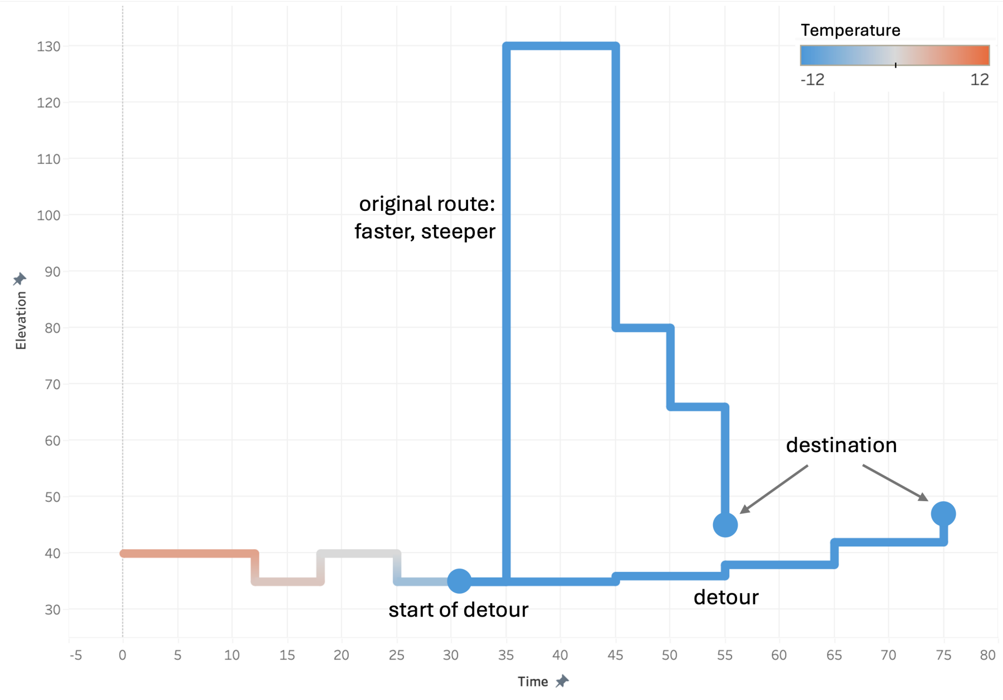


Figure 7 This line graph shows the original route as well as the detour considering time (x-axis) and elevation (y-axis). It highlights with the color how the temperature changes during the delivery.

5. Generate possible courses of actions that help eliminate the identified causes
   In order to mitigate unwanted behavior in the processes under consideration, several approaches can be considered. In our case study of truck shipment data, the following aspects are relevant:
   - **3c Traffic Jam:**
     - Reschedule departures or deliveries to off-peak hours, such as early morning or late night, when traffic is typically lighter (see Figure 8). This helps to avoid congestion and ensures faster transit times.
     - Pre-plan multiple route options for high-traffic corridors to provide flexibility in case of unexpected delays. Having alternative routes ready can significantly reduce delivery time during peak traffic hours.
   - **3c Breakdown:**
     - Implement a Preventive Maintenance Program and schedule regular inspections to ensure vehicles remain in optimal working condition. This reduces the likelihood of breakdowns during operations.
     - Equip trucks with diagnostic sensors that monitor real-time engine health, tire pressure, battery condition, and fluid levels. These sensors provide early warnings for potential issues, allowing for proactive maintenance.
     - Train drivers on basic maintenance techniques and how to detect early signs of mechanical problems. This training empowers drivers to address minor issues on the road and report major concerns promptly.
   - **3d Road Closing:**
     - Subscribe to local transportation authority alerts to stay informed about planned or emergency road closures. This ensures that dispatchers are aware of disruptions as they occur.

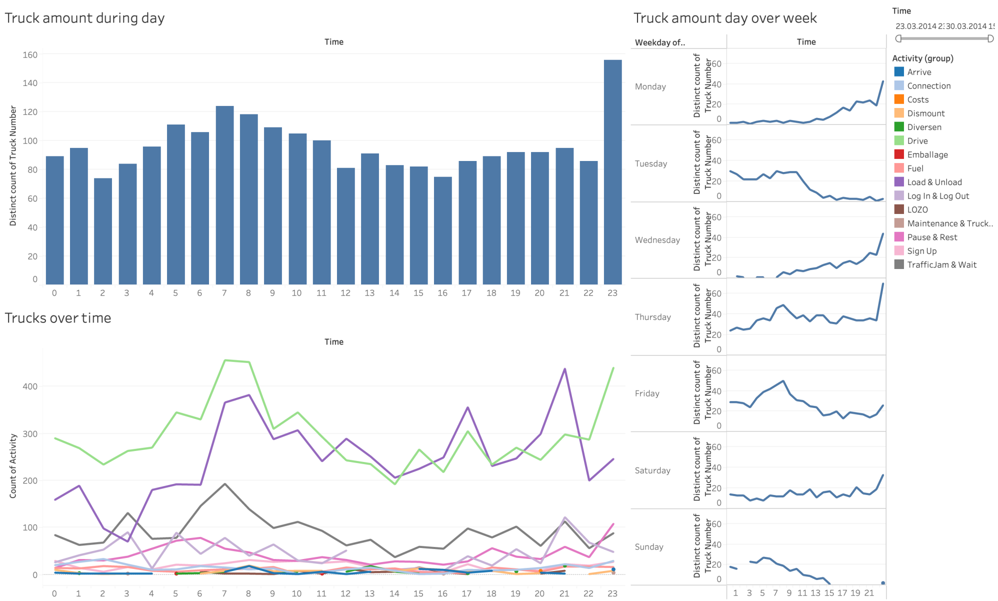


Figure 8 Top left: bar chart showing the number of trucks in operation across the time of day. The peak at midnight is clearly distinguishable. We also see a slight peak around 7am. Bottom left: The distribution of operating trucks across time of day broken down by activity type. The drive (green), load/unload (purple), and traffic jam activities (dark grey) show behavior similar to the bar chart. Right: distribution of recorded activities for each week day. Interestingly, the distribution on Tuesday does not follow the common pattern that shows a peak towards midnight.

  - Train dispatchers to respond quickly to road closures by adjusting delivery plans in real time. This includes rerouting vehicles and communicating changes effectively to drivers.
  - Provide drivers with tools, such as GPS systems or mobile apps, and the authority to reroute themselves when necessary. Drivers should base their decisions on verified guidance to avoid further delays or complications.
- **3d Inclement Weather (e.g., Snowfall):**
  - Equip vehicles for winter driving by installing snow tires, chains, or other necessary equipment. Perform seasonal maintenance checks before winter begins to ensure vehicles are prepared for adverse weather conditions.
  - Include buffer time in delivery schedules during winter months to account for potential delays caused by snowfall or icy roads. This ensures that delivery commitments can still be met despite challenging weather conditions.
- **3d Regulations on Forbidden Roads:**
  - Use regulation-aware routing software that accounts for various restrictions, including truck-specific limitations such as height, weight, and HAZMAT routes; time-based restrictions like no deliveries during school hours; and restricted access zones such as pedestrian areas or Low Emission Zones (LEZs). This ensures compliance with local regulations while optimizing delivery routes.
  - Apply for permits or exceptions where available to gain access to restricted areas when necessary. This is particularly useful for special deliveries that cannot avoid these zones.

  - Pre-plan delivery schedules and routes that avoid restricted areas or times whenever possible. Use geofencing technology to flag or block dispatch routes that intersect forbidden roads, ensuring drivers follow compliant paths at all times.

6. Evaluate courses of action As a final step, possible courses of action need to be critically evaluated and reflected. These considerations include aspects like:
   - **Cost-Benefit Ratios and Conflicting KPIs**: Taking a detour to not cross prohibited regions might avoid fines but involves increased cost for fuel and longer driving times. Similarly, a detour that avoids a traffic jam or closed road to still deliver the goods on time involves increased cost for fuel.
   - **Infeasible Actions**: Measures requiring additional resource allocation, such as increasing the number of trucks and drivers, might not be feasible due to budget constraints.
   - **Trade-off**: As conflicting KPIs prevent the existence of an obvious optimal course of action, decision-makers are required to find compromises. When facing a traffic jam, for example, a longer duration when staying on the road needs to be carefully balanced with the increased distance and risk of other issues when leaving the road to bypass the traffic jam.

Reflections

Through the systematic case study on the logistics dataset, we identify sub-tasks of process improvement that highlight the need for multi-faceted visual analysis support. Interactive visualizations can support the identification of process improvement opportunities from event logs as well as external factors including domain knowledge and experience. Identifying the relevant KPIs provides an initial indication of the relevant data facets to be explored. While our characterization revealed sub-tasks not relying on visualization, we give examples of visual representations that depict the relevant data facets and their combinations for the identification of undesired behavior, its potential explanations, and actual causes. We note that traditional process mining analysis typically focuses on the temporal dimension, activity ordering (i.e., the control flow), and resources. Our characterization reveals additional facets and their combinations that, coupled with an interactive visual analysis, enable a more comprehensive analysis for the purpose of process improvement.

References

1 Christian Tominski and Heidrun Schumann. *Interactive Visual Data Analysis.* AK Peters Visualization Series. CRC Press, 2020.
2 Michael Sedlmair, Miriah D. Meyer, and Tamara Munzner. Design Study Methodology: Reflections From the Trenches and the Stacks. In *IEEE Trans. Vis. Comput. Graph.* 18(12): 2431-2440, 2012.
3 Kateryna Kubrak, Fredrik Milani, and Alexander Nolte. A visual approach to support process analysts in working with process improvement opportunities. In *Bus. Process. Manag. J.* 29(8): 101-132, 2023.

4.2 Progressive Visual Analytics for Streaming Process Mining – VESPA

Andrea Burattin (Technical University of Denmark – Lyngby, DK), Silvia Miksch (TU Wien, AT), Shazia Sadiq (University of Queensland – Brisbane, AU), Hans-Jörg Schulz (Aarhus University, DK), and Katerina Vrotsou (Linköping University, SE)

4.2.1 Introduction and Motivation

Consider the real-time decision-making needed when managing a busy emergency response (ER) department. Patients are coming in and undergo a particular sequence of diagnosis and possibly also treatment steps that can be conceptually captured as a process model, which may change depending on the time of day (working hours vs. after hours) and case load (business as usual vs. state of emergency). The head of the department needs to monitor the current intake, throughput, and related KPIs, like the length of stay (LOS) or ward load (WL) to decide in real time whether to allocate additional resources (activate on-call doctors), to fast-track certain patients (increase their urgency levels), explicitly switch from the usual procedures to the streamlined emergency procedures or back, etc.

To support time-critical decisions, like these in real-time scenarios, we propose to combine streaming process mining with progressive Visual Analytics (VA).

Streaming Process Mining (SPM). To tackle the scenario above-mentioned, an "offline" process mining approach would not be suitable, as it is not capable of delivering real-time results. Streaming process mining [4] techniques, on the other hand, have emerged to handle these situations. In a streaming setting, events are processed, immediately after they are generated, by a streaming process mining pipeline and the corresponding (intermediate) results are made available.

Streaming process mining algorithms can be used to handle the control-flow discovery, where the control-flow is expected to represent the process *currently* being executed [6]. Another problem that can be tackled is streaming conformance checking [7], where the conformity of each event is verified against a corresponding reference model.

Progressive Visual Analytics (PVA). This concept is a flavor of VA that is tailored to sensemaking using partial intermediate computational results and visualizations [2, 9]. While these partial results are usually the outcome of some technical process (e.g., a running computation that refines its output over time or a complex data query that yields more and more matching data over time) – partial results can also be the result of a natural or organizational process.

In the given example, the progressive nature of the data stems from the fact that none of the currently treated patients in the ER have yet completed the process, being at some intermediate stage of it. If they complete the process and are either discharged or administered to a ward, we would have all their information in full, but that information is no longer relevant as they are not at the ER anymore. This means that the head of the ER department must make organizational decisions based on these incomplete patient trajectories, which yields a unique PVA scenario akin to Transient Visual Analytics – i.e., PVA with regression, which is a "forgetting" of data after a while [17].

Coupling Guidance and Progressiveness in VA Model. Guidance in VA is characterized as an active process addressing "knowledge gaps" of the users that hinder their analytical progress by identifying them and providing orienting, directing, and prescriptive guidance [8]. In Figure 9, we present a systematic view of how guidance and progressiveness can be coupled (for more detail, see [16]).

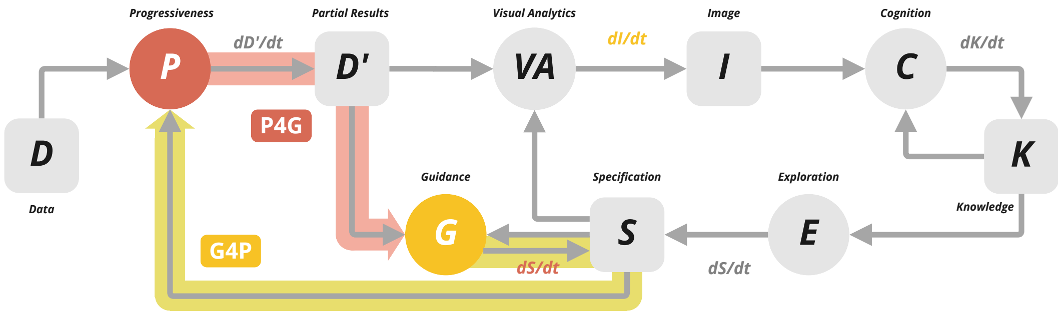


Figure 9 Coupling Guidance and Progressiveness in VA model [16] – Extension of van Wijk's model of visualization [19], including a guidance agent G (based on the extension already proposed by Ceneda et al. [8]) and progressiveness agent P. In **G4P** (yellow), G provides guidance for the steering of P, while P mediates between data D and the rest of the system, producing also the visualization progression dI/dt. In **P4G** (red), P only mediates between D and G. G behaves progressively in this case inducing the guidance progression dS/dt, while D outputs directly to VA.

4.2.2 Our Approach: VESPA

In developing our approach **VESPA** (**V**isual **E**vent-**S**tream **P**ro[gressive|cess] **A**nalytics), we considered the characteristics of our problem space, which includes SPM, PVA, and expectation of a multi-faceted problem. Our discussion revealed the dimensions of the problem space as provided below. We observed a natural connection between the streaming nature of the process and the value of progressiveness toward providing intermediate (partial) results. The dimensions further allowed us to articulate two research questions relating to timing and appropriateness of the visualization (and interaction).

Dimensions of the Problem Space

1. Context, i.e Business Process
2. Task, e.g conformance checking or process enhancement
3. Data space assumes at least an event log but it could be augmented with other facets relevant to the problem
4. Algorithm space, i.e. the specific algorithm relevant to the task
5. Guards/ Rules [11] that signal potential attention trigger for users
6. Users include full spectrum from monitoring scenario all the way to a fully explorative scenario inspired by categorization into Observers, Searchers, Explorers [13] (see Figure 10)
7. Visualization and Interaction space, e.g., visualizing event sequences [1] and dynamic networks [3, 10], details-on-demand [14].

Research Questions. We defined two research questions (RQs) relating to the timing and the effectiveness, efficiency, and appropriateness of progressive visual analytics. The questions are posed in the context of an SPM setting, and hence, there is an expectation of a continuous flow of events.

- *RQ1. What are the required time points for progressive visualization for streaming process mining?*
  Identification of time points is related to the needs of the analytical intention of the user. We identified three needs which may arise at different time points and refer to them as scheduled, triggered and on demand as explained below:

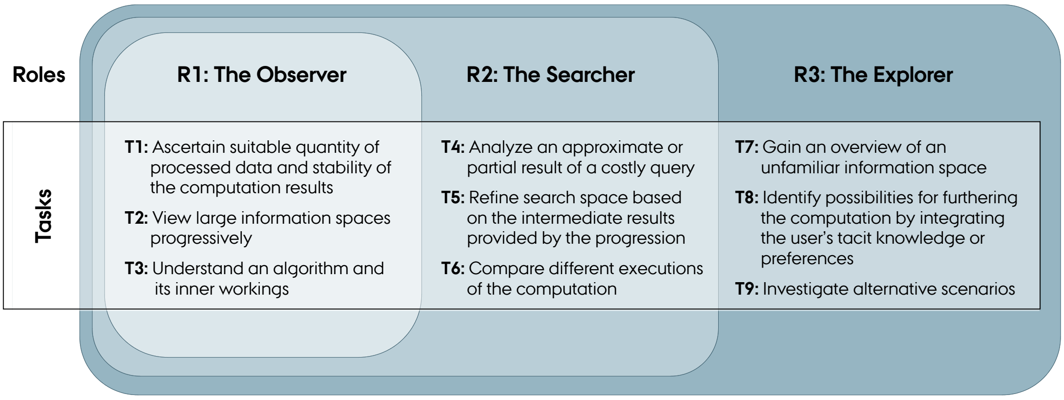


Figure 10 Common user roles and tasks in PVA. These range from the observer with only limited involvement and interaction possibilities to the explorer directly wrangling with one or possibly even more running processes in parallel. (Figure adapted from [9, ch.7]).

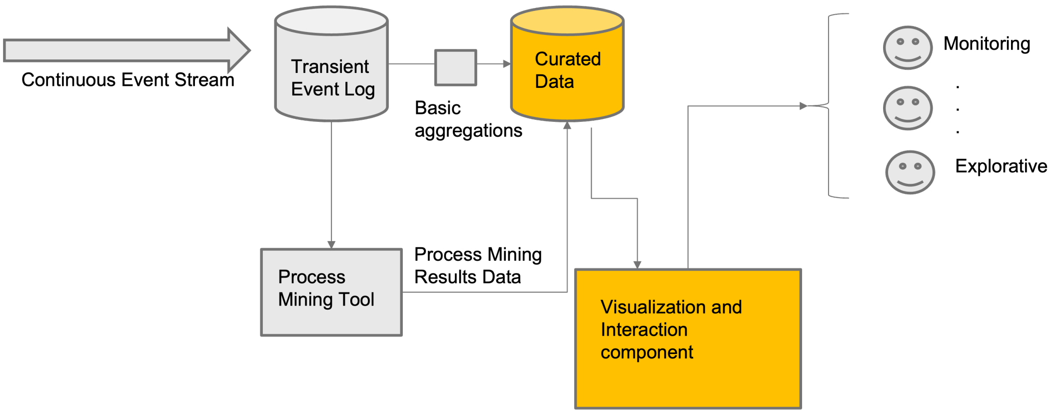


Figure 11 VESPA's Architecture. The proposed software architecture for PVA in SPM.

- Scheduled (e.g., results are ready)
- Triggered (e.g., a guard/rule fires when a conformance score is falling below threshold)
- On demand (e.g., an explorer wants to probe on a particular facet such as the trend in urgency levels)

The time points in turn will influence the suitability of the visualization and interaction which leads us to our second research question.

- *RQ2. What are the effective, efficient, and appropriate progressive visualizations and interactions for streaming process mining?*
  Expressiveness refers to the requirement of showing exactly the information contained in the data; nothing more and nothing less must be visualized [12]. *Effectiveness* primarily considers the degree to which visualization addresses the cognitive capabilities of the human visual system, but also the task at hand, the application background, and other context-related information, to obtain intuitively recognizable and interpretable visual representations [12]. Finally, *appropriateness* involves a cost-value ratio in order to assess the benefit of the visualization process with respect to achieving a given task [19].

VESPA's Architecture. The overall approach is proposed to be embedded in a software architecture as provided in Figure 11. The continuous event stream is a key feature of the problem space. Depending on the velocity of the event stream, there may or may not be a persistent storage and hence the system architecture presents it as a 'transient' event log. A process mining tool is selected based on the task e.g. conformance checking. In addition to the discovered process, the proposed system architecture also produces a process mining results dataset. This includes details such as conformance scores and multi-facted event data. When needed, the transient event log may also be used to produce some basic aggregations such as patient load over a period of time. Together the aggregations and the process mining results constitute a curated dataset that forms the input to the visualization component. The results from the visualization component are expected to empower users to perform a range of tasks from monitoring all the way to interactive exploration to support timely (or even real-time) decision making.

4.2.3 Preliminary Results

We outline a user story expressed in two levels of detail to frame and guide our VESPA approach.

- **Patient-centric:** As an ER administrator, I want to know if the LOS (length of stay) for one ER patient is too high so that I can prioritize them in the waiting queue.
- **Ward-centric:** As an ER administrator, I want to know if the overall LOS for a cohort of (or all) ER patients is too high or too low so I can adjust the allocation of resources.

This user story manifests in the problem dimensions as below:

1. Context: Healthcare
2. Task: Primarily we will focus on conformance checking, but this is intended to be augmented with relevant facets
3. Data space: A synthetic purpose-built event log has been generated using ChatGPT 4.0 (see Figure 12 and further explanation below)
4. Algorithm space: We use behavioral conformance checking (BCC) [7]
5. Guards/Rules: Three rules are considered: conformance falling below threshold; a load of an urgent category (critical/high) increasing over a threshold; and the LOS of a given patient increasing over a threshold
6. Users: Interchangeable roles of Observers, Searchers, Explorers
7. Visualization and Interaction space is aligned with the two user stories of patient-centric and ward-centric with more details provided below

To generate the synthetic dataset ChatGPT has been iteratively queried. The first datasets contains a collection of patient visits to the ER. Specifically, we asked to generate 300 patients according to the following schema: patient ID, condition, urgency level, gender, and age.[3] The distribution of the data resulting from the generation is represented in Figure 13.

[3] The exact prompt for the generation of the patients is: *can you create a dataset of 300 patients with the following attributes: Patient id, condition, urgency level, gender, age . conditions include things like flue, headache, fracture, infection, cold, allergy, chest pains, asthma etc. Aim for about 10-12 conditions. Urgency level includes critical, high, medium and low and should make sense for the condition of the patient. Sometimes the same condition can have a different urgency level for example a fracture can be low or critical. Try to generate a variety*

Patient ID	Condition	Urgency Level	Gender	Age
P0001	sprain	low	male	95
P0002	cold	low	male	18
P0003	migraine	medium	other	95
P0004	stomach ache	low	other	55
P0005	flu	medium	male	28
P0006	infection	critical	other	4
P0007	stomach ache	low	other	84
P0008	migraine	low	female	29
P0009	asthma	critical	female	1
P0010	fracture	low	female	36
P0011	fracture	medium	female	14
P0012	headache	medium	male	46
P0013	allergy	high	female	6
P0014	migraine	high	other	16
P0015	chest pains	critical	other	38
P0016	sprain	medium	other	25
P0017	migraine	medium	male	85
P0018	infection	high	male	30
P0019	headache	medium	female	59
P0020	sprain	medium	male	48
P0021	allergy	low	other	35
P0022	migraine	low	other	10
P0023	burn	critical	male	69
P0024	migraine	medium	male	60
P0025	chest pains	high	other	89
P0026	stomach ache	low	other	42
P0027	flu	medium	male	41
P0028	chest pains	high	male	28
P0029	burn	critical	female	28

Patient Profile

	Timestamp	Activity	Patient ID
0	2025-04-10 8:00:00	on route	P0039
1	2025-04-10 8:00:00	see a medical profes	P0124
2	2025-04-10 8:00:00	registration	P0162
3	2025-04-10 8:01:00	on route	P0118
4	2025-04-10 8:02:00	on route	P0139
5	2025-04-10 8:02:00	registration	P0205
6	2025-04-10 8:02:00	on route	P0161
7	2025-04-10 8:02:00	on route	P0120
8	2025-04-10 8:03:00	registration	P0088
9	2025-04-10 8:03:00	on route	P0083
10	2025-04-10 8:03:00	on route	P0200
11	2025-04-10 8:03:00	on route	P0273
12	2025-04-10 8:03:00	registration	P0153
13	2025-04-10 8:04:00	on route	P0210
14	2025-04-10 8:04:00	on route	P0174
15	2025-04-10 8:04:00	see a medical profes	P0282
16	2025-04-10 8:04:00	registration	P0228
17	2025-04-10 8:05:00	see a medical profes	P0215
18	2025-04-10 8:05:00	on route	P0233
19	2025-04-10 8:05:00	registration	P0234
20	2025-04-10 8:06:00	registration	P0154
21	2025-04-10 8:07:00	see a medical profes	P0096
22	2025-04-10 8:07:00	see a medical profes	P0118
23	2025-04-10 8:09:00	registration	P0093
24	2025-04-10 8:09:00	on route	P0064
25	2025-04-10 8:09:00	registration	P0056
26	2025-04-10 8:09:00	on route	P0016
27	2025-04-10 8:09:00	see a medical profes	P0091
28	2025-04-10 8:10:00	registration	P0126
29	2025-04-10 8:10:00	on route	P0200
30	2025-04-10 8:10:00	on route	P0280

Event Log

Figure 12 Data. Sample of the Generated Patient Profile and Event Log.

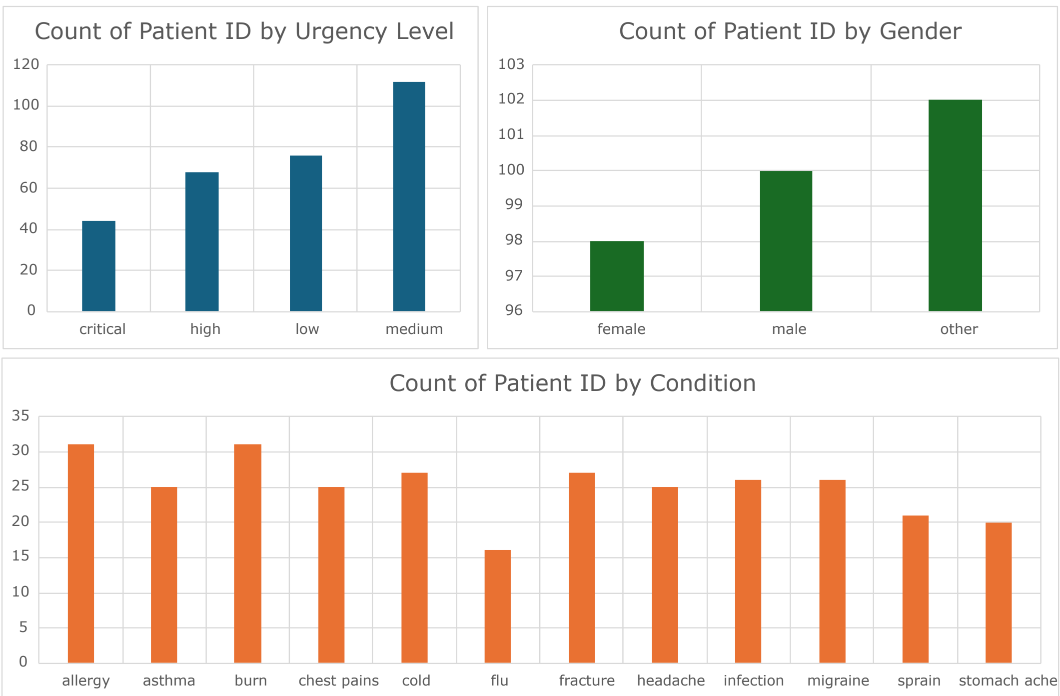


Figure 13 Background information of the data. Distribution of condition, urgency levels, and gender on the patients' dataset.

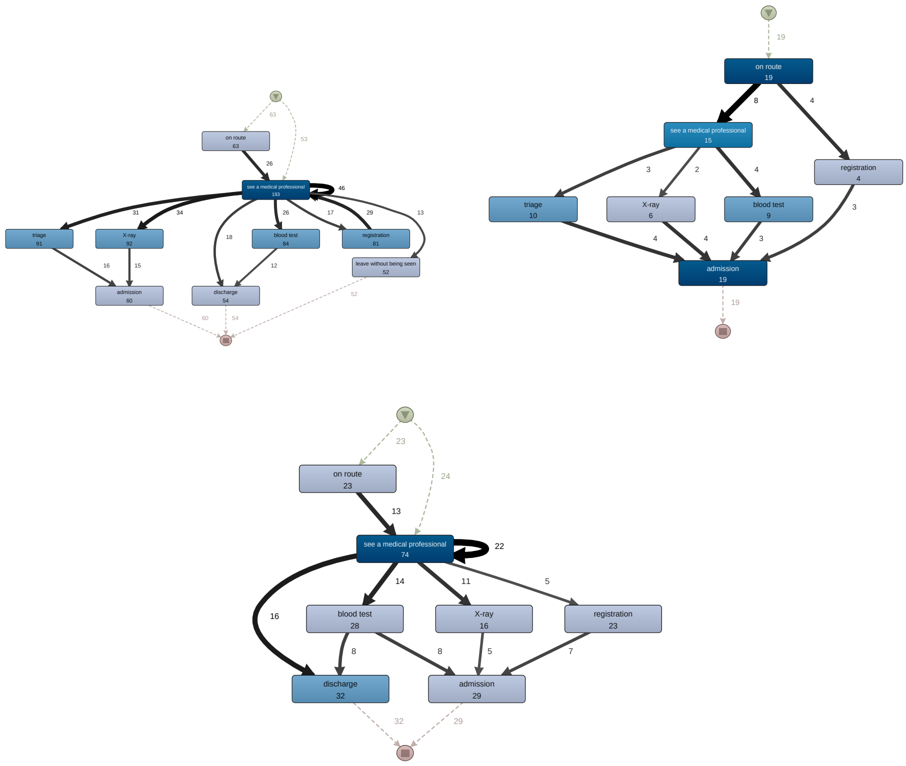


Figure 14 The three models using DFG notation. Models M1 (top left), M2 (top right), and M3 (bottom center) referring to the three variations of our process stream.

Starting from the set of patients, we asked the system to generate sequences of emergency room activities, hinting at the type of activities we are interested in.[4] The obtained dataset contained 1,512 events for the 300 patients referring to 268 case variants (so most of the patients followed a unique sequence of activities). On this dataset, some traces were manually removed to avoid particularly meaningless situations and the resulting event log had 770 instances and 142 variants. In the rest of this text, we call this log L1.

Starting from L1, we derived two additional logs, L2 and L3, by applying additional filtering to mimic an off-peak (L2) and an intense scenario (L3). L1, L2, and L3 have also been used to mine the corresponding 3 process models (i.e., M1, M2, and M3) using the approach described in [7]. A picture of the three models built using Fluxicon Disco[5] is in Figure 14.

Finally, L1, L2, and L3 have been transformed into three lists of events by sorting each event according to its execution time. With these lists, we constructed our synthetic event stream by concatenating the following lists: L1, L2, L1, and L3 with the intention of simulating a regular daytime period, followed by off-peak (i.e., night), then back to daytime and eventually an intense scenario.

[4] The exact prompt for the generation of the activities is: *for this patient set, can you generate a log of activities relating to an emergency department. The log contains a timestamp, an activity name and a patient id from the previous dataset. Activities in the beginning can include on route, registration and see a medical professional, activities in the middle can include triage, X-ray, blood test, see a medical professional, and activities at the end can include admission, discharge or leave without being seen.*

[5] See: `https://www.fluxicon.com/disco/`.

To analyze the data, a streaming process mining pipeline has been implemented using pyBeamline [5]. The pipeline processes each event and computes the following:

- The DFG model [5, 18] updated up to the given point in time;
- The behavioral conformance value [7] of the stream against model M1/M2/M3.

All these values represent the "Process Mining Results Data" in Figure 11, and are combined with the actual raw event, which can be used to compute basic statistics, to form the "Curated Data" (see Figure 11) that is provided as input to the "Visualization and Interaction component".

VESPA-VIS's Prototypical Mock-Up

Based on the use case, synthetic data and the outlined problem dimensions, we started designing a prototype mock-up. The prototype, named VESPA-VIS, accordingly comprises two main views, the *Ward view* (see Figure 15) and the *Patient view* (see Figure 16). Each one is designed to address the two outlined levels of detail of our use case: ward-centric vs. patient-centric level.

The *Ward view* is split into a *Patient Flow* representation (see Figure 15A) and a view showing temporal overviews of the relevant facets (see Figure 15B). The *Patient Flow* displays the event streams flowing into the ER ward. A node-link diagram representing the currently active reference model (e.g., M1) is drawn as a backdrop in the view. As the events stream into the ward model the current patient load is mapped on the size of the nodes and weight of the edges. If deviations to the model appear (i.e., "unexpected" events not included in the reference model), these are drawn dashed with node and edge size following the same conventions. The *Patient Flow* displays the flow of patients over a given expert-user defined time-interval preceding the current time point (e.g., 10 minutes). A time slider allows exploration of past intervals. Hovering over a node or edge pops up a tool-tip displaying the number of patients belonging to the corresponding event or transition over time.

On the right side of the view, a selection of graphs showing the temporal distribution of relevant facets is displayed (see Figure 15B). These facet graphs complement the main *Patient Flow* view and allow an expert to inspect surrounding factors and reason about the processing state of the ward. On the top, the conformance score over time is displayed. Conformance w.r.t. the currently explored model over time is displayed by default (e.g., M1), and on hover, the view is complemented with conformance w.r.t. to complementary model variations (e.g., M2, M3). This graph allows the expert to monitor the conformance of the process over time, detect fluctuations from the expected behavior, and assess whether the correct model is used as a reference or whether another model should be used. If the conformance score reduces below a certain threshold over a certain period of time, a guard is triggered, calling for the attention of the expert. The second graph displays the ward load over time, i.e. the total number of patients being processed, allowing the expert to monitor the overall stress on the ward over time. Third, the average Length of Stay (LOS) of patients being processed is displayed over time, providing an additional cue to the stress of the ward. The fourth graph gives a summary overview of the urgency of the patients being processed over time. The distribution of the urgency classes (e.g. low, medium, high) is displayed as a layered area graph, allowing an expert to reason about the characteristics of the patients currently putting load on the ward. Additional facets could be displayed in a similar manner in the view, if deemed appropriate for the task at hand.

The *Patient Flow* view and facets graphs are updated according to three timing strategies: (1) at regular pre-defined intervals (e.g. every 10 minutes or 100 events) by default (scheduled), (2) if a guard/trigger is activated (triggered), (3) upon request of the user (on-demand). The *Ward view* is displayed for all of the available reference models in different tabs. A user can switch between exploring the event-streams against these at any time.

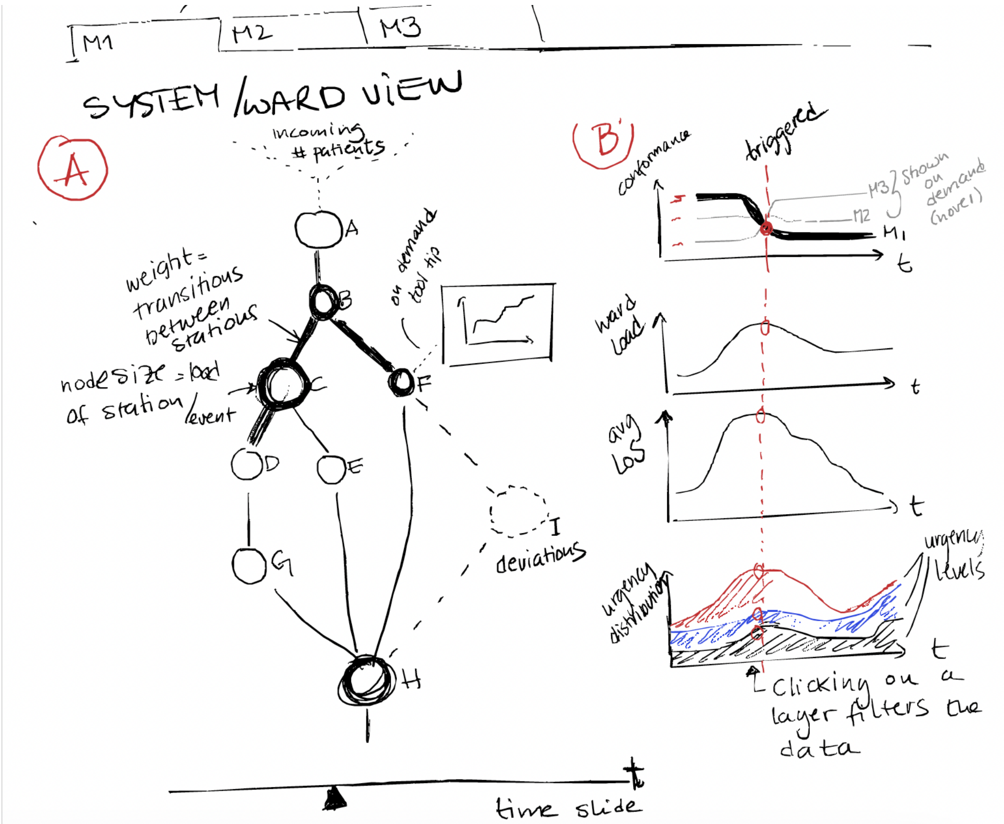


Figure 15 Mock up for Ward View.

The *Patient View* (see Figure 16) is designed to allow an expert to drill down into the individual patient event-streams when a need arises. This can, for example, be in anticipation of a forthcoming increase of load, or can occur after a guard has called attention to the need for intervention. In the *Patient View* the individual patient streams are displayed as sequences of events. Time is displayed on the horizontal axis and the user can toggle relative and absolute time. Patient sequences are sorted along the vertical axis by an *urgency score*. If the computed *urgency score* of a patient exceeds a pre-define threshold, an alarm is triggered to call attention to the need of prioritizing individual patients. The *urgency score* is computed as a distance from a benchmark sequence. Three alternative benchmark sequences are considered in VESPA-VIS:

1. "Ideal behavior". An expert pre-defined ideal path through the process both in terms of sequence of events and timing. Different ideal sequences can be defined for different times of day or days of the week.
2. "Current average". An average patient sequence reflecting the current ordering and average duration of events.
3. "User adjusted". A user-adjustable patient sequence where an expert (e.g. ER manager) can make on-line decisions regarding the target duration of events.

The ability to choose between benchmarks to compare against allows the expert user of VESPA-VIS (e.g. ER manager) to flexibly adjust the notion of urgency and control prioritization of patients according to the current situation, their domain knowledge and previous experience.

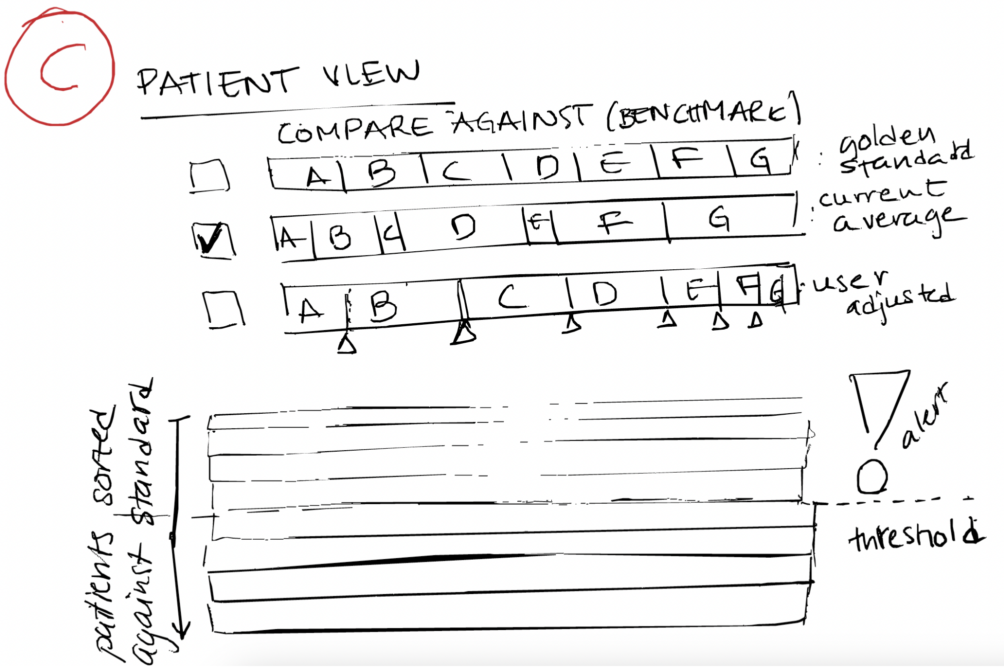


Figure 16 Mock up for Patient View.

Together the *Ward* and *Patient views* allow a user to move smoothly between user roles from observer to explorer, monitor the current situation, react on evolving changes, reason about possible explanations of these, and potentially anticipate outcomes.

4.2.4 Next Steps

The presented VESPA-VIS's mockups provide an initial illustration of our research questions. That is, when (monitoring or scheduled to more active exploration) and how (views appropriate to patient-centric and ward-centric requirements), progressive visual analytics can best support real-time decision-making in a streaming process mining context. However, there remains a number of further considerations for the proposed approach to be fully realized.

The illustration of the approach through the usecase indicates fertile ground for further developing the approach and robust evaluation to assess the effectiveness of progressive visual analytics for the (real-time) decision support. We anticipate that such an evaluation would require carefully planned user studies with representative participant groups.

Given the continuous nature of the event stream, it is natural to expect a need to "forget" previous event streams when they are no longer relevant for the current decision making. So far, we have considered a rather straightforward way of simply 'forgetting' patients who have exited the ER through discharge or transferal to another ward. Yet this prevents the head of the ER department from comparing the currently observed situation with previously observed situations – for example, the processes occurring on a current New Years holiday day to the processes on the same day in previous years – to identify best practices or simply "what has worked in the past". Identification of 'forgetfulness' thresholds is in itself a complex and multi-faceted problem that requires further work, although prior literature gives hints (see for example [15]).

Although the focus of the approach is to support real-time decision making, the insights gained from the proposed approach present an opportunity to inform process enhancement. Exploring this opportunity requires further consideration.

References

1 Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of Time-Oriented Data, Second Edition*. Springer, 2023.

2 Marco Angelini, Giuseppe Santucci, Heidrun Schumann, and Hans-Jörg Schulz. A review and characterization of progressive visual analytics. *Informatics*, 5(3):31:1–31:27, 2018.

3 Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. A Taxonomy and Survey of Dynamic Graph Visualization. *Computer Graphics Forum*, 36(1):133–159, 2017.

4 Andrea Burattin. Streaming process mining. In Wil M. P. van der Aalst and Josep Carmona, editors, *Process Mining Handbook*, volume 448 of *Lecture Notes in Business Information Processing*, pages 349–372. Springer, 2022.

5 Andrea Burattin. Beamline: A comprehensive toolkit for research and development of streaming process mining. *Softw. Impacts*, 17:100551, 2023.

6 Andrea Burattin, Alessandro Sperduti, and Wil M. P. van der Aalst. Control-flow discovery from event streams. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*, pages 2420–2427. IEEE, 2014.

7 Andrea Burattin, Sebastiaan J. van Zelst, Abel Armas-Cervantes, Boudewijn F. van Dongen, and Josep Carmona. Online conformance checking using behavioural patterns. In Mathias Weske, Marco Montali, Ingo Weber, and Jan vom Brocke, editors, *Business Process Management – 16th International Conference, BPM 2018, Sydney, NSW, Australia, September 9-14, 2018, Proceedings*, volume 11080 of *Lecture Notes in Computer Science*, pages 250–267. Springer, 2018.

8 Davide Ceneda, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jörg Schulz, Marc Streit, and Christian Tominski. Characterizing Guidance in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):111–120, 2017.

9 Jean-Daniel Fekete, Danyel Fisher, and Michael Sedlmair, editors. *Progressive Data Analysis – Roadmap and Research Agenda*. Eurographics Press, 2024.

10 Steffen Hadlak, Heidrun Schumann, and Hans-Jörg Schulz. A Survey of Multi-faceted Graph Visualization. In *Eurographics Conference on Visualization*. The Eurographics Association, 2015.

11 Jaemin Jo, Sehi L'Yi, Bongshin Lee, and Jinwook Seo. ProReveal: Progressive visual analytics with safeguards. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3109–3122, 2021.

12 Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.

13 Luana Micallef, Hans-Jörg Schulz, Marco Angelini, Michaël Aupetit, Remco Chang, Jörn Kohlhammer, Adam Perer, and Giuseppe Santucci. The human user in progressive visual analytics. In Jimmy Johannson, Filip Sadlo, and G. Elisabeta Marai, editors, *Short Paper Proceedings of EuroVis'19*, pages 19–23. Eurographics Association, 2019.

14 Tamara Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2015.

15 A. Pavan, Sourav Chakraborty, N. V. Vinodchandran, and Kuldeep S Meel. On the feasibility of forgetting in data streams. *Proceedings of the ACM on Management of Data*, 2(2), 2024.

16 Ignacio Pérez-Messina, Marco Angelini, Davide Ceneda, Christian Tominski, and Silvia Miksch. Coupling guidance and progressiveness in visual analytics. *Computer Graphics Forum*, (forthcoming), 2025.

17 Hans-Jörg Schulz and Chris Weaver. Transient visual analytics. In Mennatallah El-Assady and Hans-Jörg Schulz, editors, *Proceedings of the 15th International EuroVis Workshop on Visual Analytics (EuroVA'24)*. Eurographics Association, 2024.
18 Wil M. P. van der Aalst. *Process Mining – Data Science in Action, Second Edition*. Springer, 2016.
19 Jarke J Van Wijk. Views on visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):421–432, 2006.

4.3 Interactivity: Visual Feedback and Feedforward for Process Exploration

Francesca Zerbato (TU Eindhoven, NL), Irit Hadar (University of Haifa, IL), Philipp Koytek (Celonis Labs GmbH – München, DE), Christian Tominski (Universität Rostock, DE), and Maria-Cruz Villa-Uriol (University of Sheffield, GB)

This working group focused on the role that interactivity plays in supporting the exploration of processes in process mining (PM). The group first analyzed the current PM practice and its limitations. Then existing works related to interactive visual data exploration were collected from the Visual Analytics (VA) literature. Based on that, preliminary formalizations were sythesized and initial design ideas sketched. In particular, the focus was on enhancing PM with informative visual feedback and feedforward techniques from the VA realm.

4.3.1 The Problem of (Un)Informed Process Exploration

Like interactive visual data analysis in general [19], process mining in particular is exploratory and human-driven. Process analysts typically have to engage in iterative exploration cycles of process visualizations (see Figure 17) to build an understanding of the process, examine different scenarios, and generate hypotheses [26]. Hypotheses are then tested, refined, or discarded based on intermediate insights, which, in turn, guide analysts in choosing what to explore next [18] and lead to the crystallization of knowledge [21].

Many process mining tools support this *interactive exploration* through components such as filter masks or sliders that allow analysts to create views and isolate data subsets of interest. However, these interactions often lack transparency and context, making it difficult for users to anticipate the effects of their actions before executing them (*What should I usefully do?*) and understand the effect once an action has been executed (*Have I achieved the desired outcome?*).

As an example, consider a process mining analyst, let's name him Bob, who is in charge of analyzing a Road Fine Management process [7] visualized as a directly-follows graphs (DFG). His goal is to investigate cases where offenders do not pay their fines. Bob's analysis steps are sketched in Figure 18.

As a first step, Bob loads the raw data provided by the police (L_0) into a process mining tool. His goal is to get an initial understanding of the structure of the process. To achieve this goal, he intents to focus on the most common behavior using a *variant filter* to remove infrequent cases. However, to find a suitable abstraction level for his analysis, Bob has to go through the costly procedure of applying the filter three times (o_1, o_2, o_3). He selects different filtering thresholds of 75%, 85%, and 90% of the cases in the log, and each time he

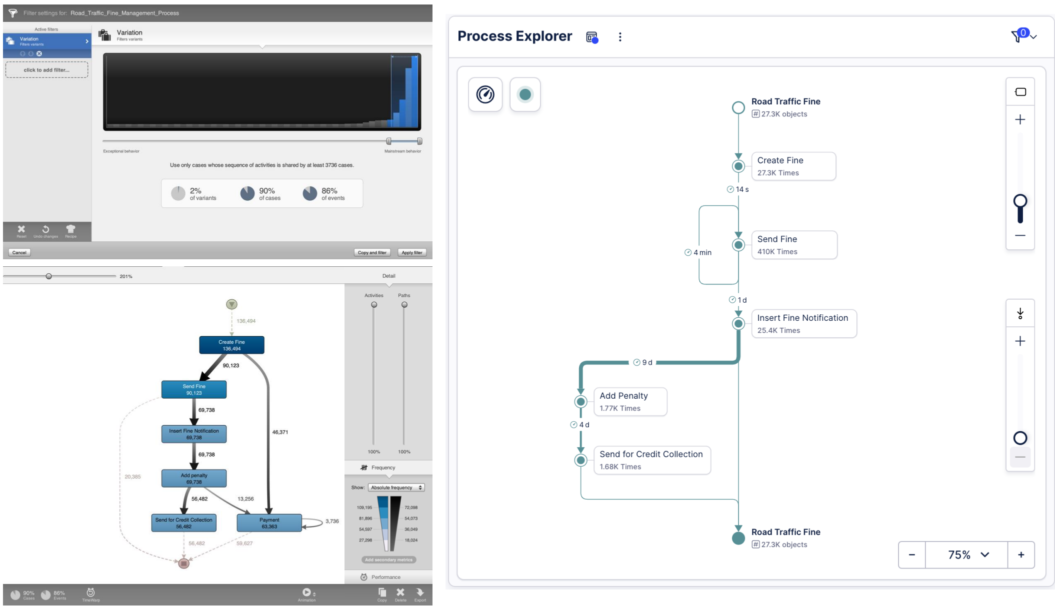


Figure 17 Most process mining tools provide visualizations of DFGs that can be filtered using sliders. However these sliders typically do not provide immediately visual feedback, which can make it difficult to understand their effects on the data and the visualizations.

	Id	Operation	I/O		Timestamp	User Intent
DFG Exploration	o_1	variantFilter(cases, keep, 75%)	L_0	L_1	07/10/22 10:01:18	Focus on most frequent behavior in DFG
	v_1	showDFG()	L_1	DFG_1	07/10/22 10:01:50	
	o_2	variantFilter(cases, keep, 85%)	L_0	L_2	07/10/22 10:02:03	
	v_2	showDFG()	L_2	DFG_2	07/10/22 10:02:32	
	o_3	variantFilter(cases, keep, 90%)	L_0	L_3	07/10/22 10:03:11	
	v_3	showDFG()	L_3	DFG_3	07/10/22 10:03:29	

Figure 18 Bob's first analysis steps adapted from [25] represented as **Operations**, the input and output **I/O**, the **Timestamp** at which each operation occurred and the higher level **User Intent**.

inspects the resulting DFG (v_1, v_2, v_3) to assess visually the effects of the filters. While the first two filter configurations remove too many cases, he settles for o_3.

This interactive selection of most frequent cases is a common first step of many process mining analyses. However, during this process, Bob runs into several limitations:

- There is no preview mechanism to support the decision for a suitable filter threshold: Bob must fully apply each filter to see what the result looks like;
- Comparisons across multiple visualizations resulting from the filtering are lost: Since each filter operation completely replaces the DFG with a new one, Bob is forced to take screenshots or compare from short-term memory;
- The DFG shows the impact of his filter on the control-flow only: Bob would need to create different views of the data to see how his filtering impacts other facets of the process.

The difficulties Bob faces demonstrate some of the challenges that stem from limitations of interaction functionalities currently being used in PM practice. These limitations introduce constraints to the process of process mining (PPM) [18] and potentially hinder performance when making sense of event log data. The objective of our working group is to investigate these challenges and outline the opportunity of enhancing the process exploration based on established concepts, methods, and techniques from the realm of VA.

With our working group, we aim to improve the overall process of making sense during the PPM. To this end, we compiled relevant previous work from the PM and VA communities, both to understand some of the cognitive challenges during process exploration and to bring together models and approaches that can inform the design of advanced process exploration techniques to overcome these challenges.

4.3.2 Relevant Works Related to Interactivity in Process Exploration

During the group discussion, we considered several works from PM and VA. These works are concerned with the cognitive, processing, and interactive mechanisms that are relevant during visual process exploration.

Related work on Process Mining

The process of process mining (PPM) is a recent stream of research studying the sequence of activities – both from behavioral and cognitive points of view [18, 26, 27]. Such studies are important for informing efforts toward providing support to the cognitive processes underlying the PPM. The PPM starts based on a general goal (e.g., identifying obstacles in the process), building on available event datasets, and continues to additional operations, such as filtering the data to explore it from different angles, interpreting the data, and trying to make sense of them in order to find insights relevant to the goal at hand.

Looking at the PPM from a cognitive perspective, during process mining, the data serve as input signals coming from the 'external world'. The sense-making process entails an iterative cycle, where attention is focused according to a set goal, leading to the generation of hypotheses about the process, which are then tested and reconsidered against the data for minimizing the prediction error [18]. This process not only aligns with general knowledge generation processes in VA [21, 15], but also with the post-cognitivism principle of prediction error minimization (PEM) [6, 11] in particular.

PEM conceives the brain as a probabilistic inference system, which attempts to predict the input it receives by constructing models of the possible causes of this input. While aiming to minimize the prediction error (i.e., the gap between the predicted and the actual input), it either introduces small refinements to the model or substantial revisions (or even a complete replacement of the model), depending on the size of the error. This process is iteratively performed until the prediction error is satisfactorily small.

Figure 19 illustrates the adapted model proposed by Sorokina et al. [18] named PEM4PPM. The model captures the sequence of PM steps and their corresponding cognitive operations. It begins with high-level business goals that can be decomposed or refined into more specific ones as needed. The refinement process iterates until the goals are concrete enough to be achieved through available PM operations. To focus attention on studied aspects of the input data, a relevant subset of the data is filtered and organized, enabling subsequent exploration of the data to identify behavioral patterns that are of interest. Data exploration is conducted to uncover behavior patterns that may be relevant to the set goals. Based on the exploration results, concrete predictions are generated, in the form of hypotheses or artifacts (e.g., process models) and then tested. The results obtained from these steps are assessed against the original goal or hypothesis to evaluate prediction errors and take actions for their minimization. This assessment serves as a basis for determining whether the goal has been achieved, thus leading to a conclusion, or if further refinement is needed, in which case the process continues in another iteration [18].

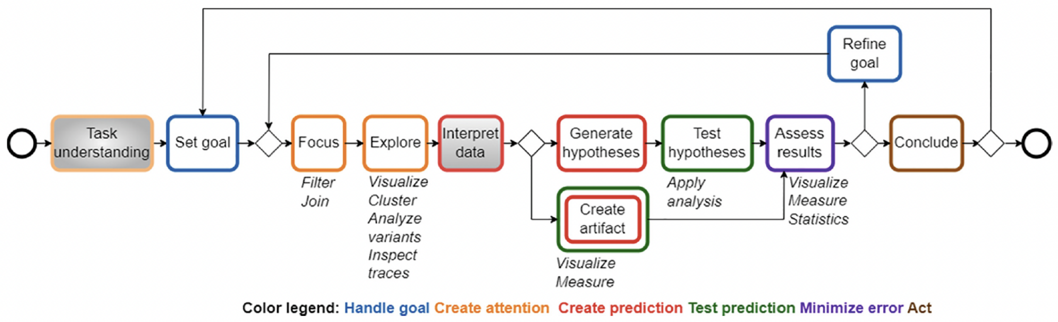


Figure 19 The PEM4PPM model illustrating the cognitive steps of the Process of Process Mining. Image from [18].

Related Works from Visual Analytics

The VA community has worked extensively on interactive visual data exploration. There are several important related works that are relevant in this regard, underpinning effective user interaction and data exploration:

- On a more abstract level than the PEM4PPM, **Norman's action cycle** [14] provides a crucial framework for understanding the stages users go through when interacting with a system, emphasizing the gulfs of execution and evaluation that interactive visualization interfaces should aim to bridge in order to minimize interaction costs [13].
- **Shneiderman's visual information seeking mantra** [17] characterizes the general process of interactive data exploration as: *Overview first, zoom and filter, then details on demand*. This mantra has been expanded to the **Visual Analytics Mantra** [12]: *Analyze first, show the important, zoom and filter, and analyze further, details on demand.*
- **Brushing & linking** [2] and **dynamic queries** [16] provide techniques that enable simultaneous highlighting and filtering of related data in different views. They offer users immediate and continuous visual feedback as they manipulate query parameters, fostering an iterative and exploratory analysis process across multiple perspectives.
- **Fluid interaction** [8] has been conceived to create seamless and responsive visualization interfaces that minimize the cognitive load of interaction, allowing users to stay in the analysis flow and focus on data insights.
- **Guidance mechanisms** [4, 3] aim to actively support users in their analytical process, ranging from subtle visual cues to more explicit recommendations, helping them navigate complex datasets and analysis tasks effectively.
- **Visual Feedback and Feedforward** [22] are essential principles for designing intuitive interactive systems. Commonly, visual feedback informs users about the results of their actions. However, only rarely is visual feedforward applied to provide users with cues suggesting available options and potential interaction outcomes.

These interconnected concepts collectively contribute to the design of powerful and user-friendly VA tools. This working group particularly focused on visual feedback and feedforward as promising, yet so far under-explored mechanisms to enhance process exploration in PM. By dynamically adding information to existing visual process representations, they can support the understanding of interaction effects and the decisions of the user about their next activity.

4.3.3 Conceptualization of Feedback and Feedforward

In an attempt to pinpoint the fundamental conceptual aspects of the desired process exploration support, we came up with the following (incomplete) list of notations inspired by the section on interactive selection and accentuation in [19]:

- D, the data to be visualized, explored, and understood
- $D_+ \subseteq D$, the currently relevant focus data, subject to change frequently during process exploration
- $D_- = D \setminus D_+$, the data currently not being of relevance for the process exploration
- S, a state capturing the data underlying the visualization
- S_{cur}, the "current" state
- S_{old}, the "old" state
- $\{S_{a_1}, S_{a_2}, \ldots, S_{a_n}\}$, a set of possible (useful) "alternative" states that can be entered through alternative interactions
- $\delta(S_i, S_j)$, the explicit difference(s) between two states
- ...

Based on these notations, we defined exploration as the repeated refinement and change of D^+ (and D_- respectively), which usually involves numerous state changes (e.g., the three different filtering states in Bob's exploration example). Moreover, it seems that understanding state changes is crucial for effective exploration. Possible options for supporting the understanding of state changes can be based on Gleicher et al.'s [10] strategies for visual comparison:

- **Juxtaposition:** Visualize S_i and S_j side by side
- **Superposition:** Superimpose the visualization of S_i over the visualization of S_j
- **Explicit encodingg:** Visualize $\delta(S_i, S_j)$ directly

So far, these strategies are not sufficiently integrated into existing process exploration practice!

To understand the users' needs better, we further conceptualized a cycle of interaction for a seamless analysis that is based on feedback and feedforward. Generally, in interactive exploration/analysis, the analyst interacts with the visualizations for dicing, slicing and relating different parts of the data. So, the exploration cycle starts with the analyst expressing an **intent to change** the visual representation. Yi et al. [24] identified several different categories of interaction intents called "Show me...", of which we focus on the intents related to changing the focused subset of the data exploration:

- **Show me something else:** A different subset of the data (e.g., navigate in time) will be visualized; $D_+ \to D'_+$.
- **Show me more/less:** A subset of different size or level of aggregation (e.g., reduce number of nodes in DFG) will be visualized; $|D_+| \neq |D'_+|$.
- **Show me something conditionally:** A subset that fulfills certain (filter) condition(s) (e.g., filter for frequent variants) will be visualized; $C(D_+) = true$.
- **Show me related things:** A subset that is (in some way) related to the currently shown subset (e.g., brushing and linking across multiple faceted views) will be visualized; $R(D_+, D'_+)$, typically $D_+ \sim D'_+$.

These intents lead to interactions to which the system responds by providing visual feedback, and what we would like to emphasize, also visual feedforward. We envisioned the following scenario in which a user executes an interaction and receives the corresponding visual feedback and feedforward.

1. The user's intention is typically communicated to the system through different ways of interaction (e.g., hovering a visual mark in the visualization or clicking a button or slider in the user interface).
2. The system interprets the user's action and then provides *relevant* context and suggests possible next steps with previews of their impact. Here we can explore a large design space of different useful visual feedback and feedforward, which generally are dependent on the semantics of the interaction and will incorporate different facets of the data. The additional context and possible next steps help the analyst to decide if the intended interaction outcome has been obtained, and if not, how to execute their alternative more fruitful interactions.
3. Based on the visual feedback and feedforward, the analyst can now better understand interaction effects and can more easily decide what to do next. The cycle starts again as the analyst continues the exploration and expresses their new intents through new interactions.

With this general scenario now being clear, the question that remains is how to design the visual feedforward and feedback concretely. However, given the huge design space, this is quite a challenging task.

4.3.4 Preliminary Design Examples

For our design sketches, we drew inspiration from previous work on enhancing interaction with visual feedback and feedforward. In particular, we considered:

- Scented widgets [23, 5] embed small miniature visualizations (e.g., histograms) directly into graphical control elements such as sliders.
- Small multiples and large singles [20] is a concept to preview thumbnails of alternative parameterizations of visual representations.
- Guidance visual cues [9] can be embedded into visualization views to indicate potentially interesting next navigation targets.
- Octopocus [1] is an interaction technique that provides feedforward as an interactive gesture is performed to indicate possible interaction outcomes.

The sketches used these inspirational techniques to outline possible solutions for informative visual feedback and feedforward for process exploration. Figure 20 shows a selection of our sketches, including conceptualization of interaction intents, general interface ideas, scented slider widgets, and preview thumbnails. From these and further similar sketches, we abstracted the following design dimensions that can play a role when implementing enhanced process exploration mechanisms:

Interaction control: What type of control is used to carry out the interaction? Slider, button, hover area, spoken command, etc.

Interaction integration: Where is the interaction control located? Integrated in the visualization vs. external to the visualization in a separate user interface.

Visual feedback/feedforward integration: Where is the visual feedback/feedforward shown? Visualization enhancement integrated into the visualization vs. Interface enhancement integrated into the interaction control.

Summary

In summary, the working group made some preliminary first steps toward overcoming the current PM limitations by integrating VA approaches. It became clear that completely solving the problem remains a formidable challenge for future work. Not only would it be

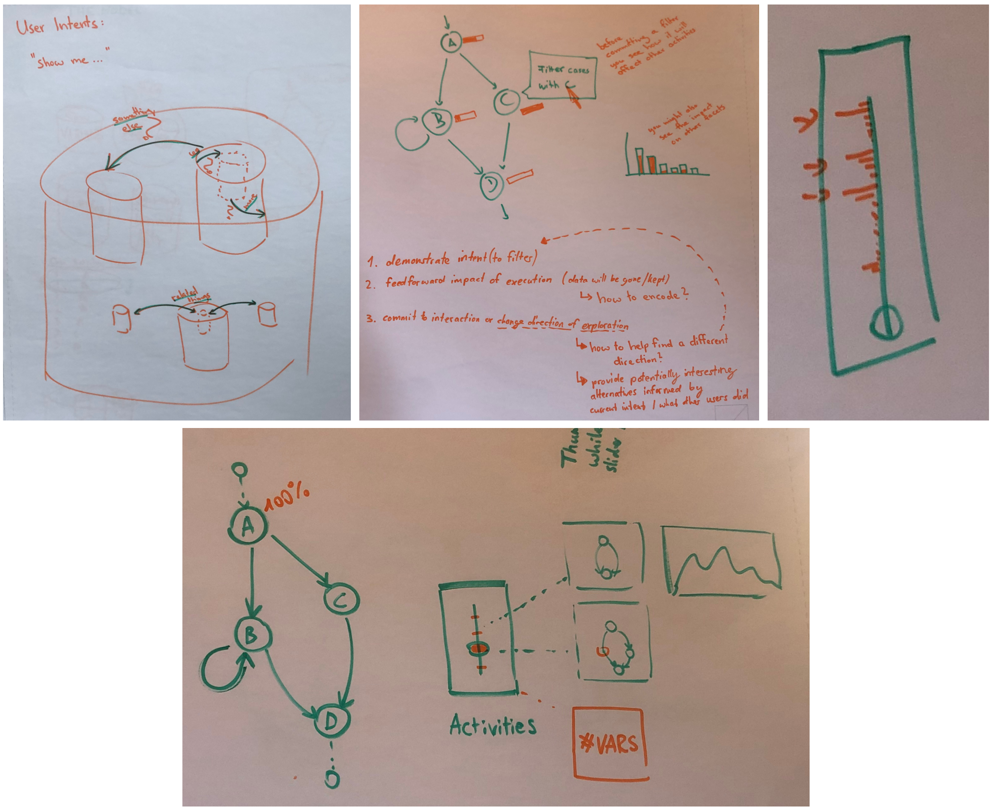


Figure 20 Selected sketches showing different "Show me..." interactions (top left), general action loop for DFG filtering (top center), scented slider widget (top right), and preview thumbnails attached to a slider (bottom).

necessary to more comprehensively map the design space of visual feedback and feedforward, but one would also need to implement the new designs into PM tools, which may just not be ready for handling the multiple states, visual feedbacks and feedforwards in their underlying architecture. Therefore, we suggest putting more research and development efforts into interactivity to support process exploration.

References

1 Olivier Bau and Wendy E. Mackay. Octopocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, page 37–46, New York, NY, USA, 2008. Association for Computing Machinery.

2 Richard A. Becker and William S. Cleveland. Brushing Scatterplots. *Technometrics*, 29(2):127–142, 1987.

3 Davide Ceneda, Natalia Andrienko, Gennady Andrienko, Theresia Gschwandtner, Silvia Miksch, Nikolaus Piccolotto, Tobias Schreck, Marc Streit, Josef Suschnigg, and Christian Tominski. Guide me in analysis: A framework for guidance designers. *Computer Graphics Forum*, 39(6):269–288, 2020.

4 Davide Ceneda, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jörg Schulz, Marc Streit, and Christian Tominski. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):111–120, 2017.

5 Vasile Ciorna, Nicolas Médoc, Guy Melançon, Frank Petry, and Mohammad Ghoniem. A tour through the zoo of scented widgets. In Christina Gillmann, Michael Krone, and Simone Lenti, editors, *25th Eurographics Conference on Visualization, EuroVis 2023 – Posters, Leipzig, Germany, June 12-16, 2023*, pages 9–11. Eurographics Association, 2023.
6 Andy Clark. Busting out: Predictive brains, embodied minds, and the puzzle of the evidentiary veil. *Noûs*, 51(4):727–753, 2017.
7 Massimiliano De Leoni and Felix Mannhardt. Road traffic fine management process. *Eindhoven University of Technology, Dataset*, 2015.
8 Niklas Elmqvist, Andrew Vande Moere, Hans-Christian Jetter, Daniel Cernea, Harald Reiterer, and TJ Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10(4):327–340, 2011.
9 Stefan Gladisch, Heidrun Schumann, and Christian Tominski. Navigation Recommendations for Exploring Hierarchical Graphs. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Baoxin Li, Fatih Porikli, Victor Zordan, James Klosowski, Sabine Coquillart, Xun Luo, Min Chen, and David Gotz, editors, *Advances in Visual Computing: Proceedings of the International Symposium on Visual Computing (ISVC)*, volume 8034 of *Lecture Notes in Computer Science*, pages 36–47. Springer, 2013.
10 Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D. Hansen, and Jonathan C. Roberts. Visual Comparison for Information Visualization. *Information Visualization*, 10(4):289–309, 2011.
11 Jakob Hohwy. The self-evidencing brain. *Noûs*, 50(2):259–285, 2016.
12 Daniel A. Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. Challenges in Visual Data Analysis. In *International Conference Information Visualisation (IV)*, pages 9–16. IEEE Computer Society, 2006.
13 Heidi Lam. A Framework of Interaction Costs in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1149–1156, 2008.
14 Donald A. Norman. *The Design of Everyday Things, Revised and Expanded Edition.* Basic Books, 2013.
15 Dominik Sacha, Andreas Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A. Keim. Knowledge Generation Model for Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1604–1613, 2014.
16 Ben Shneiderman. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6):70–77, 1994.

17 Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *IEEE Symposium on Visual Languages (VL)*, pages 336–343. IEEE Computer Society, 1996.
18 Elizaveta Sorokina, Pnina Soffer, Irit Hadar, Uri Leron, Francesca Zerbato, and Barbara Weber. PEM4PPM: A cognitive perspective on the process of process mining. In *International Conference on Business Process Management*, pages 465–481. Springer, 2023.
19 Christian Tominski and Heidrun Schumann. *Interactive Visual Data Analysis*. AK Peters Visualization Series. CRC Press, 2020.
20 Stef van den Elzen and Jarke J. van Wijk. Small Multiples, Large Singles: A New Approach for Visual Data Exploration. *Computer Graphics Forum*, 32(3):191–200, 2013.
21 Jarke J. van Wijk. Views on Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):421–433, 2006.
22 Jo Vermeulen, Kris Luyten, Elise van den Hoven, and Karin Coninx. Crossing the bridge over norman's gulf of execution: revealing feedforward's true identity. In *Conference on Human Factors in Computing Systems*, pages 1931–1940. ACM, 2013.
23 Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1129–1136, 2007.
24 Ji Soo Yi, Youn ah Kang, John T. Stasko, and Julie A. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.
25 Francesca Zerbato, Andrea Burattin, Hagen Völzer, Paul Nelson Becker, Elia Boscaini, and Barbara Weber. Supporting provenance and data awareness in exploratory process mining. In *International Conference on Advanced Information Systems Engineering*, pages 454–470. Springer, 2023.
26 Francesca Zerbato, Pnina Soffer, and Barbara Weber. Initial insights into exploratory process mining practices. In *Business Process Management Forum: BPM Forum 2021, Rome, Italy, September 06–10, 2021, Proceedings 19*, pages 145–161. Springer, 2021.
27 Francesca Zerbato, Pnina Soffer, and Barbara Weber. Process mining practices: Evidence from interviews. In *Int. Conf. on Business Process Management (BPM)*, LNCS, pages 268–285. Springer, 2022.

4.4 Coordinated Projections: A New Approach to Multi-Faceted Process Exploration

Barbara Weber (Universität St. Gallen, CH), Gennady Andrienko (Fraunhofer IAIS – Sankt Augustin, DE), Andreas Kerren (Linköping University, SE), Manuel Resinas (University of Sevilla, ES), Stef Van den Elzen (TU Eindhoven, NL), and Peilin Yu (Linköping University, SE)

Working group D (see Figure 21) focused on new visual representations for multi-faceted process exploration.

Figure 21 Group D (from left to right): Stef, Manuel, Gennady, Peilin, Andreas, Barbara.

4.4.1 Motivation

Process exploration constitutes a fundamental task in process mining, primarily aimed at facilitating the exploration and generation of hypotheses about the underlying process behavior captured in event data. In particular, during the initial phases of process mining projects, analysts engage in various exploratory activities – they dedicate time to familiarize themselves with the data, develop a preliminary understanding of the process, formulate or refine analytical questions, and uncover unexpected patterns or insights [6]. This iterative exploration plays a crucial role in shaping the direction of subsequent hypothesis testing [5].

The way process exploration is currently performed typically involves the use of discovered *Process Maps*, such as the Directly-Follows Graphs (DFGs) [27, 9, 17], where nodes represent activities and edges indicate direct successions based on event log timestamps. Using the DFG as a visual representation to enable exploration generally implies that the primary facet (i.e., the first-class citizen of our analytical workflow) is the temporal order of activities [16].

More detailed insights are often introduced through additional visual channels, such as projecting performance metrics (e.g., activity duration, waiting time) or frequency information (e.g., how often transitions occur) on top of the DFG structure. These additional visual cues enrich the visualization but do not alter the primary facet. This allows the analyst to gain insight on different aspects of the process, such as identifying activities (e.g., activities with high duration, infrequent activities, endpoint activities), fragments (e.g., the most frequent fragment), transitions (e.g., transitions with high durations), or bottlenecks [10].

Less commonly, the DFG can be restructured entirely by using resources as the primary facet [11]. In resource-centric DFGs, nodes represent individuals, roles, or organizational units, and the edges reflect handovers or collaborative interactions. This shift enables analysis of organizational dynamics, revealing patterns such as teamwork structures, silos, or handover inefficiencies.

Despite the value of these perspectives, a major limitation is that the primary facet is typically fixed, constraining the flexibility of exploration. Real-world analytical tasks often require users to continuously shift between facets [17] – from case-centric to activity-centric, from control flow to resource flow, or from process variants to attribute analysis in order to discover inter- and intra-dependencies. Fixing the facet can obscure relevant patterns, reduce user agency, and lead to a cognitive mismatch when visualizations do not align with the user's current task or mental model [13].

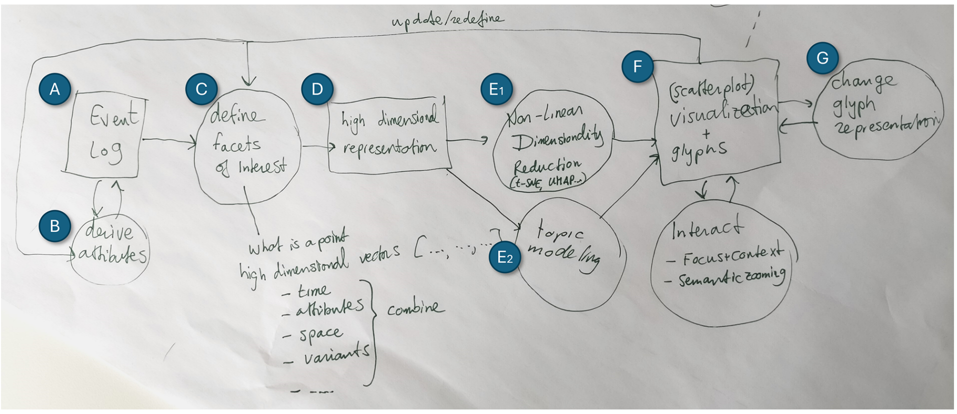


Figure 22 Our new approach to Multi-Faceted Process Exploration going from A) event log data with B) derived attributes, to user defined facets of interest C). These facets are then represented in a high-dimensional space through different encodings D). From the high-dimensional representations we can apply dimensionality reduction E1) or topic modeling E2) for visualization purposes F). Interaction techniques and the use of glyphs G) enable exploration and analysis to close the sense-making loop.

To truly support hypothesis generation and sense-making, process exploration tools must support flexible transitions between facets, allowing analysts to dynamically reframe the data perspective depending on their line of inquiry [17]. Typically, users are interested in exploring *similarities* and *differences* between (combinations of) facets, i.e., the visual analysis approach should support more complex comparison tasks. To support this, dimensionality reduction might help here; it is flexible in what facets are considered and helps to reveal similarities (local neighborhoods) in the high-dimensional space.

4.4.2 Approaches

We investigated methods for analyzing event logs from multiple, complementary facets [2], with the goal of decomposing them into smaller, more interpretable subsets. These event log subsets can then be explored in greater detail using compact and more readable representations such as DFGs. To this end, we experimented with both topic modeling [12] and embedding-based techniques [1][6], applying various strategies to represent event logs either as textual documents (for topic modeling) or as structured feature vectors (for embedding). In particular, we started exploring representations that capture both the composition of logs as collections of activities and as sequences of transitions between activities. These representations have potential to account not only for the temporal order in which activities occur, but also incorporate quantitative time aspects such as activity durations and additional contextual attributes. Finally, we explored the possibilities that multiple coordinated views can provide as a way to enhance the analysis process.

The result of these investigations is summarized in the workflow presented in Figure 22. In the following, we describe each of the steps involved in the workflow.

[6] https://va-embeddings-browser.ivis.itn.liu.se

A. Original Data Set. We used the publicly available Road Traffic Fine event log as example [4]. The Road Traffic Fines event log documents the handling of traffic fines by a local police force in Italy. It contains approximately 561,470 events across 150,370 cases, recorded between January 2000 and June 2013. The process involves 11 activities and 12 data attributes.

Each case starts with a *Create Fine* event, which includes the fine amount next to other attributes. The offender can pay the fine at any time via a *Payment* event. The amount paid is recorded in the attribute *paymentAmount*. If not paid, a *Send Fine* action sends a letter, possibly incurring additional charges (*expenses*). This is followed by *Insert Fine Notification* and, if necessary, *Add Penalty*, which increases the amount due. If the fine remains unpaid, an event *Send for Credit Collection* indicates an escalation to a collection agency. Offenders may also appeal to the prefecture or a judge, triggering events such as *Insert Appeal* and *Notify Result Appeal to Offender*. If an appeal is successful, the fine is marked as dismissed via the *dismissal* attribute.

We selected this dataset because it supports exploration across multiple meaningful facets. It includes a rich combination of control-flow, temporal, and data perspectives. Events are timestamped, enabling the discovery of relevant temporal constraints (e.g., the fine notification must be sent within 90 days of the fine creation; otherwise, there is no obligation to pay). In addition, the event log captures a variety of attributes, such as *amount*, *expense*, *totalPaymentAmount*, and appeal results (attribute *dismissal* containing a flag whether and by whom the fine is dismissed). These attributes enable the identification of patterns within specific subpopulations of cases and help correlate behavioral differences with underlying data characteristics. This multidimensionality makes the dataset particularly well-suited for studying the need for flexible faceting and for supporting dynamic transitions between different analytical perspectives during process exploration.

B. From Event Log to Enriched Case Log. We then transformed the event log into a case log and enriched it using case predicates as suggested in [7]. More specifically, we enriched the case log with process outcomes. For this, each case was assigned a specific outcome of the process (that is, fully paid, dismissed, credit collected, and unresolved) according to [7]. Fully paid cases are cases where the last outstanding balance is $<=$ to 0. Dismissed cases are cases with the dismissal code $\in \{\#, G\}$. Cases are assigned the label credit collected if activity *Send for Credit Collection* is present in the trace. The remaining cases were then classified as unresolved. Moreover, we added to the enriched case log the value of the dismissal attribute of the last activity of the case, as well as a derived attribute *outstandingBalance*, which is calculated as the sum of *amounts* plus the sum of expenses minus *totalPaymentAmount*.

C. Define Facets of Interest. To explore the facets of interest within the process data, we employ dimensionality reduction or topic modeling techniques to project high-dimensional attribute representations into a two-dimensional space suitable for visualization. This allows us to generate scatterplots in which each point corresponds to a specific process element, such as a case, activity, variant, or resource, depending on the analytical perspective adopted (e.g., similar to earlier work on dynamic network exploration [26]).

The goal of this step is therefore to determine which process elements will be represented as individual points and which attributes will define their position in the projection space. These attributes are selected based on the analytical goal and may capture various process dimensions. These include control-flow characteristics (e.g., activity sequences, frequency patterns), outcome-related indicators (e.g., outstanding balance, dismissal codes), contextual attributes (e.g., vehicle class, notification type), temporal aspects (e.g., activity duration, case throughput time), or process variants (e.g., distinct execution paths).

In this work, we focus exclusively on representing cases as points in the scatterplot, using attributes related to control flow and outcome indicators to construct the projections.

D. Multi-faceted Trace Encoding. Based on the selected facets of interest, we used the the original and the enriched case log to implement several different alternative encodings in a high-dimensional space. These encodings will then be used as input for the topic modeling and dimensionality reduction step.

For topic modeling, we considered representing traces as sets of activities and as sets of direct transitions between the activities. For example, having trace of consecutive activities A, B, and C, we represent either as a string *A B C* or as a string *A_B B_C*, reflecting activities and their transitions, accordingly. The third variant combines both representations, uniting two alternative viewpoints *A B C A_B B_C*.

For DR, we constructed two different encodings that covered different facets of the dataset. The first encoding focused on attributes deemed relevant to the outcome of each case [7]. Specifically, we included two attributes to represent whether the *outstandingBalance* and the *totalPaymentAmount* are greater than zero. We also included the *dismissal* code from the enriched case log and the last activity recorded in each case, since it helps to determine the outcome of the case if it goes to credit collection. Since both the dismissal code and the last activity are categorical attributes, we applied one-hot encoding to allow the application of a DR algorithm in the next step.

The second encoding was designed to capture the sequence of activities performed within each case. For this purpose, we extracted the sequence of activities for every case, preserving their order of execution. Since cases can vary in the number of activities, we padded shorter sequences with a special padding token to ensure all sequences had the same length, matching the case with the highest number of activities. This preprocessing step resulted in an $n \times m$ matrix, where n is the total number of cases, and m represents the length of the longest case. Finally, we applied one-hot encoding to each column of the matrix, expanding the $n \times m$ matrix into a higher-dimensional format where categorical activity labels are represented in a numerical form.

E. Non-Linear Dimensionality Reduction and Topic Modeling. We then applied dimensionality reduction techniques as well as topic modeling using the multi-faceted trace encodings described above. Here we opted for a non-linear dimensionality reduction techniques to support the exploration of *similarities* such as UMAP or t-SNE [14].

F. Visualization and Interaction. To support the exploration of the DR results we use a coordinated multiple view approach [3] in which two or more visualizations are connected through linking and brushing (see Figure 23), i.e., if items in one visualization are highlighted or selected, the corresponding items in the other visualization are also highlighted or selected.

This enables users to explore the different data-facets in context of each other, see Figure 24 (e.g., explore the correlation between temporal order and data attributes). For the visualization of the DR result, the most used choice is a scatterplot-like representation. This has the advantage that the main facet of interest is encoded with the highest ranked visual channel of position [24]. To encode the additional facets, we can then use additional visual channels, such as color, shape, and size. However, as these channels are limited, we propose to use glyphs (cf. next paragraph) to encode the additional facets. The scatterplot also enables a scalable solution with respect to the number of items as here interaction techniques such as *focus+context* and *semantic zooming* alleviate overplotting issues.

We have also applied topic modeling (see Figure 25) based on the combined representation of traces as sets of events and transitions between events. To validate the results, we made three UMAP projections (see Table 1) of all traces based on weights of topics, orders, and attributes of the logs. These projections have been colored according to outcomes of the traces (3rd column), main topic (fourth column), and also propagated continuous 2D color schemes (shown in the 2nd column) across the three projections (columns 5-7).

G. Changing Glyphs. Prior research demonstrates that glyph-based representations can be employed across diverse application domains and serve a variety of analytical and communicative purposes. Various design alternatives of the most commonly used glyph types have been examined and discussed in numerous prior studies [18, 19, 20].

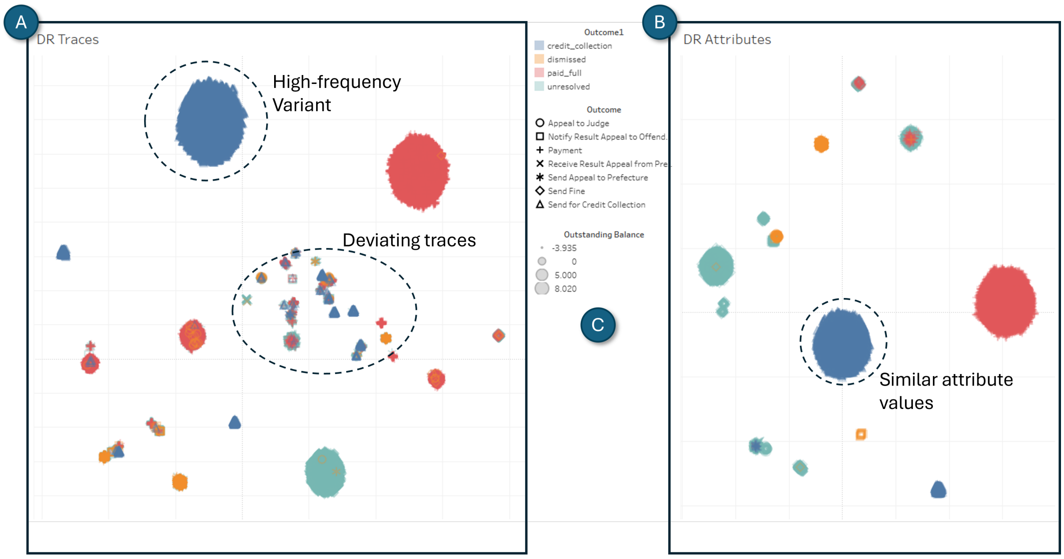


Figure 23 Multiple coordinated view of two scatterplots showing the dimensionality results of A) traces and B) attributes. C) An interactive legend enables filtering.

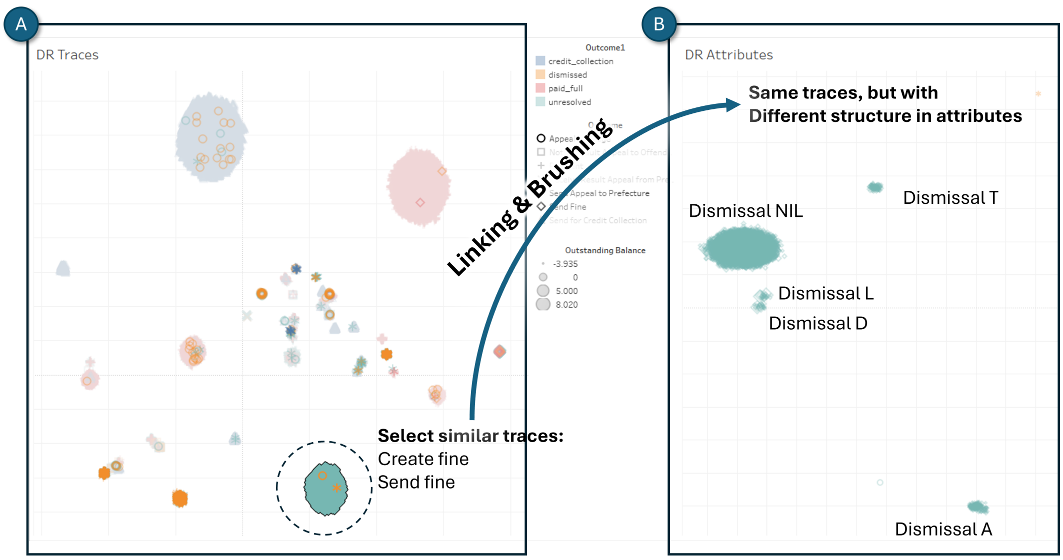


Figure 24 Linking and brushing enables multi-faceted exploration to explore both in context of one another.

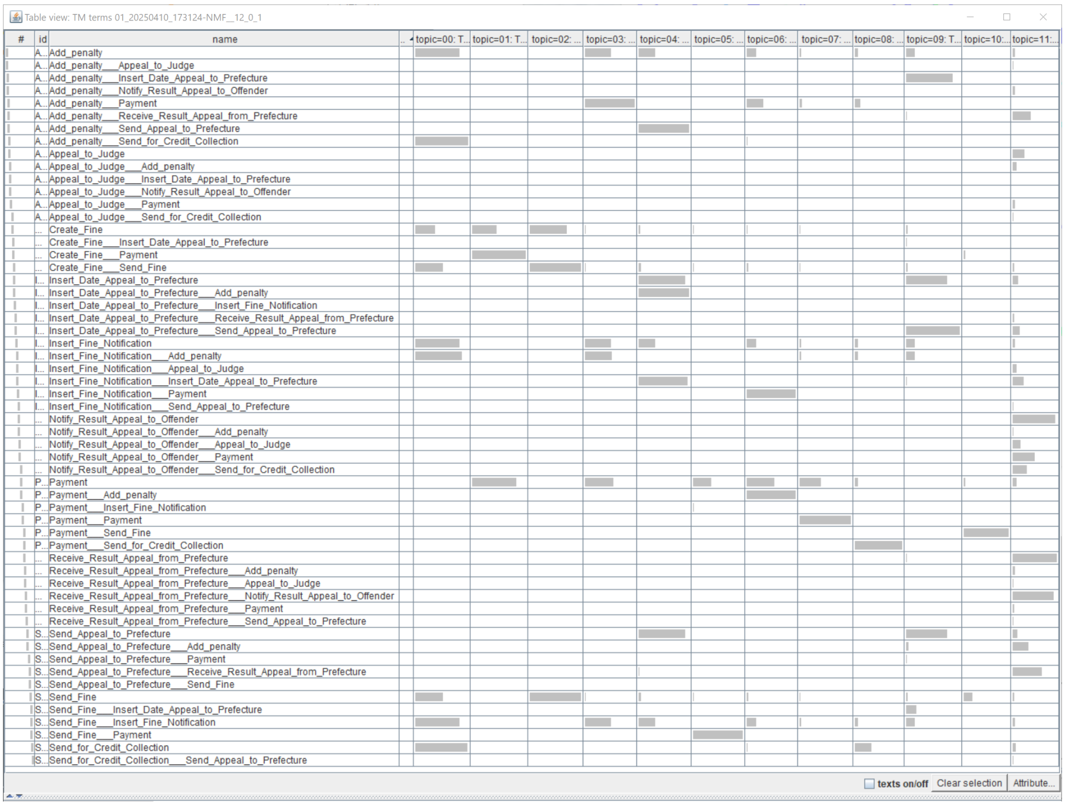
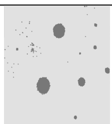
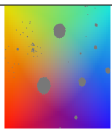
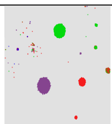
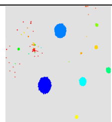
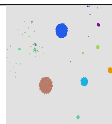
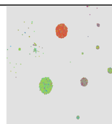
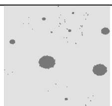
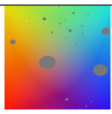
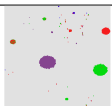
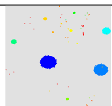
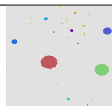
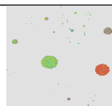
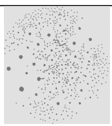
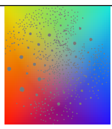
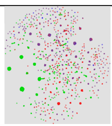
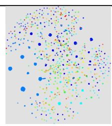
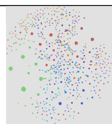
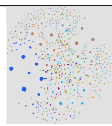


Figure 25 Table with bar charts demonstrates the composition of topics over terms, with bar charts representing term weights in the topics.

Multiple data attributes can be encoded within a single glyph, either to represent multiple properties of a single entity or to aggregate information across multiple entities. Additionally, certain glyph types (e.g., face-based and icon-based representations) typically exhibit a one-to-one mapping between glyph and data entity. Different glyph types can be combined to encode more complex data facets. However, excessive data encoding within a single

Table 1 Results of UMAP embedding based on topics (top row), order (middle row), and attributes (bottom row).

Base File	-	-outcome	-mainTopic	-topics	-order	-attrs
						
						
						

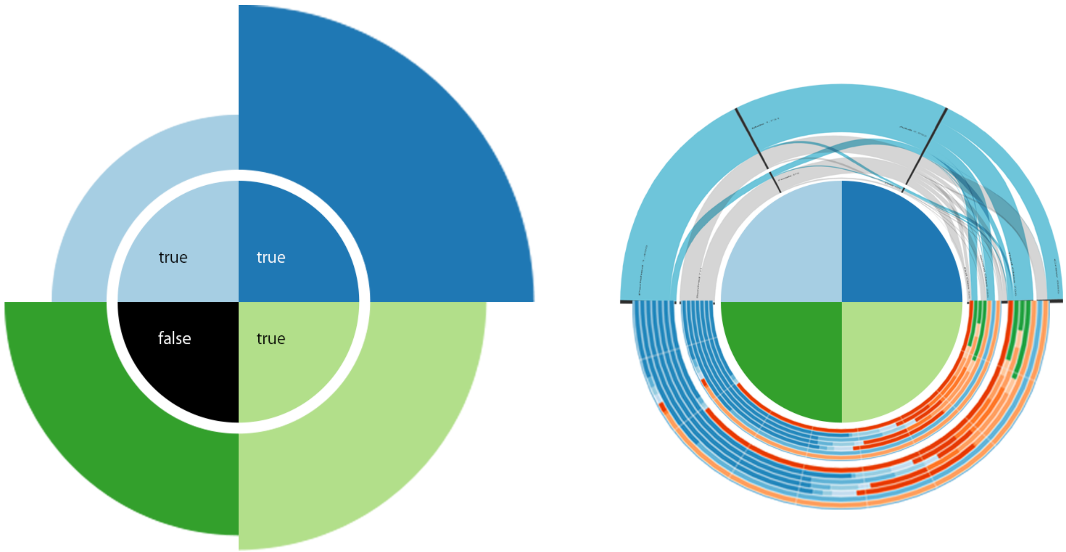


Figure 26 The proposed glyph prototype is designed to encode and represent both Boolean and quantitative attributes (left), and the proposed glyph prototype that incorporates encoding of Boolean attributes, directly-follows relations, and temporal information (right).

glyph may result in visual clutter or scalability issues, potentially impairing perceptual efficiency [18]. Layout strategies for glyph drawing commonly include linear and circular or radial configurations. While radial layouts can yield visually engaging designs, they should not be regarded as universally optimal solutions for all information visualization challenges [21]. Recent studies in information visualization have proposed methods for the automatic generation of glyph designs to facilitate creating effective and context-appropriate representations [22, 23]. In addition to the primary selected facet, alternative layout designs can be employed based on supplementary facets of interest and the specific analytical tasks at hand.

Two prototype designs for encoding and representing Boolean, quantitative attributes, directly-follows relations, and temporal information are shown in Figure 26. In both designs, the Boolean attribute is encoded using the inner circle, which is rendered in black by default when the Boolean value is false. The left design utilizes a surrounding radial bar chart to convey quantitative values across a defined range, whereas the right design integrates additional visual encodings to represent directly-follows relations and temporal progression.

The scatter plot, generated via dimensionality reduction, serves as an overview of high-dimensional data. Upon user selection of individual or multiple scatter points, a corresponding glyph-based representation is activated to reveal detailed attribute-level information. Users can interactively customize the glyphs to control which attributes are displayed, apply filters, or adjust visual encoding parameters, thereby supporting flexible, multi-faceted, and task-driven exploration of the underlying data.

4.4.3 Outlook

As part of future work, we plan to implement a comprehensive visual analytics system that builds on our current approach. In the current system design, each dot in the visualization represents a case positioned using DR techniques. Moving forward, we aim to explore alternative semantic representations for dots, such as activities or process variants. Moreover, we would like to systematically investigate how to encode activities, variants, and traces in a multi-faceted manner to best support various exploratory goals. To further enrich the visual expressiveness, we plan to integrate glyph-based representations that convey additional attributes or contextual cues (e.g., for selected subpopulations). Moreover, we intend to integrate our approach with existing process discovery algorithms to support the creation of DFGs for selected subpopulations. Additionally, we intend to expand the visual space to support additional facets, including relationships, control-flow, and resources. Another interesting research direction is to study ensemble methods for embeddings where various embedding approaches might be combined (either conceptually different state-of-the-art embedding technologies or the same embedding algorithm with various hyperparameter settings) to provide better performance [15]. A key objective of our future work will be to evaluate the effectiveness of the proposed system in supporting exploratory process mining, demonstrating its ability to accommodate a wide range of analytical tasks and user needs.

References

1. Zeyang Huang, Daniel Witschard, Kostiantyn Kucher, Andreas Kerren. VA + Embeddings STAR: A State-of-the-Art Report on the Use of Embeddings in Visual Analytics. In *Comput. Graph. Forum 42*, vol. 42, no. 3, pp. 539-571. 2023.
2. Johannes Kehrer, and Helwig Hauser. Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 495–513, March 2013.
3. Jonathan C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Fifth international conference on coordinated and multiple views in exploratory visualization* (pp. 61-71). IEEE. 2007.
4. Massimiliano de Leoni and Felix Mannhardt. Road traffic fine management process. *Eindhoven University of Technology, Dataset*, 284, 2015.
5. Elizaveta Sorokina, Pnina Soffer, Irit Hadar, Uri Leron, Francesca Zerbato, and Barbara Weber. PEM4PPM: A Cognitive Perspective on the Process of Process Mining. In Chiara Di Francescomarino, Andrea Burattin, Christian Janiesch, and Shazia Sadiq (Eds.), *Business Process Management – 21st International Conference, BPM 2023, Utrecht, The Netherlands, September 11-15, 2023, Proceedings*, volume 14159 of Lecture Notes in Computer Science, pages 465–481. Springer, 2023.
6. Francesca Zerbato, Pnina Soffer, and Barbara Weber. Initial Insights into Exploratory Process Mining Practices. In Artem Polyvyanyy, Moe Thandar Wynn, Amy Van Looy, and Manfred Reichert (Eds.), *Business Process Management Forum – BPM Forum 2021, Rome, Italy, September 06-10, 2021, Proceedings*, volume 427 of Lecture Notes in Business Information Processing, pages 145–161. Springer, 2021.
7. Hagen Völzer, Francesca Zerbato, Timothy Sulzer, and Barbara Weber. A Fresh Approach to Analyze Process Outcomes. In *5th International Conference on Process Mining, ICPM 2023, Rome, Italy, October 23-27, 2023*, pages 97–104. IEEE, 2023.
8. Christopher Klinkmüller, Richard Müller, and Ingo Weber. Mining Process Mining Practices: An Exploratory Characterization of Information Needs in Process Analytics. In Thomas Hildebrandt, Boudewijn F. van Dongen, Maximilian Röglinger, and Jan Mendling (Eds.), *Business Process Management*, volume 11675 of Lecture Notes in Computer Science, pages 322–337. Springer International Publishing, 2019.
9. Carlos Capitán-Agudo, María Salas-Urbano, Cristina Cabanillas, and Manuel Resinas. Analyzing How Process Mining Reports Answer Time Performance Questions. In *Business Process Management*, volume 13420 of Lecture Notes in Computer Science, pages 234–250. Springer International Publishing, 2022.

10 María Salas-Urbano, Carlos Capitán-Agudo, Cristina Cabanillas, and Manuel Resinas. Designing a User Interface to Explore Collections of Directly-Follows Graphs for Process Mining Analysis. In *Enterprise, Business-Process and Information Systems Modeling – 25th International Conference, BPMDS 2024, and 29th International Conference, EMMSAD 2024, Limassol, Cyprus, June 3-4, 2024, Proceedings*, volume 511 of Lecture Notes in Business Information Processing, pages 35–47. Springer, 2024.

11 Minseok Song and Wil M. P. van der Aalst. Towards comprehensive support for organizational mining. *Decision Support Systems*, 46(1):300–317, December 2008.

12 David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

13 Johannes Kehrer and Helwig Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Trans. Vis. Comput. Graph.*, 19(3):495–513, 2013.

14 Mateus Espadoto, Rafael M. Martins, Andreas Kerren, Nina ST Hirata, and Alexandru C. Telea. Toward a Quantitative Survey of Dimension Reduction Techniques. *IEEE Trans. Vis. Comput. Graph.*, 27(3):2153–2173, 2021.

15 Daniel Witschard, Ilir Jusufi, Rafael Messias Martins, and Andreas Kerren. A Statement Report on the Use of Multiple Embeddings for Visual Analytics of Multivariate Networks. *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) – IVAPP*, pages 219–223, 2021.

16 Wil M. P. van der Aalst. *Process Mining – Data Science in Action, Second Edition*. Springer, 2016.

17 Silvia Miksch, Claudio Di Ciccio, Pnina Soffer, and Barbara Weber. Visual Analytics Meets Process Mining: Challenges and Opportunities. *IEEE Computer Graphics and Applications*, 44(6):132–141, 2024.

18 Chun-houh Chen, Wolfgang Härdle, Antony Unwin, and Matthew O. Ward. Multivariate data glyphs: Principles and practice. Handbook of data visualization 179-198, 2008.

19 Johannes Fuchs, Petra Isenberg, Anastasia Bezerianos, and Daniel Keim. A systematic review of experimental studies on data glyphs. IEEE transactions on visualization and computer graphics, 2016, 23(7), 1863–1879.

20 Mandy Keck, Dietrich Kammer, Thomas Gründer, Thomas Thom, Martin Kleinsteuber, Alexander Maasch, and Rainer Groh. Towards glyph-based visualizations for big data clustering. Proceedings of the 10th international symposium on visual information communication and interaction, 2017, 129–136

21 Geoffrey M. Draper, Yarden Livnat, and Richard F. Riesenfeld. A survey of radial methods for information visualization. IEEE transactions on visualization and computer graphics 2009, 15(5), 759–776.

22 Lu Ying, Tan Tang, Yuzhe Luo, Lvkeshen Shen, Xiao Xie, Lingyun Yu, and Yingcai Wu. GlyphCreator: Towards example-based automatic generation of circular glyphs. IEEE Transactions on Visualization and Computer Graphics, 2021, 28(1), 400–410.

23 Lu Ying, Xinhuan Shu, Dazhen Deng, Yuchen Yang, Tan Tang, Lingyun Yu, and Yingcai Wu. MetaGlyph: Automatic generation of metaphoric glyph-based visualization. IEEE Transactions on Visualization and Computer Graphics, 2022, 29(1), 331–341.

24 Tamara Munzner. Visualization analysis and design. CRC press, 2014.

25 Kammer, Dietrich, Mandy Keck, Thomas Gründer, Alexander Maasch, Thomas Thom, Martin Kleinsteuber, and Rainer Groh. Glyphboard: Visual Exploration of High-Dimensional Data Combining Glyphs with Dimensionality Reduction. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 4, pp. 1661–1671, 2020.

26 Stef van den Elzen, Danny Holten, Jorik Blaas, and Jarke J. van Wijk. Reducing Snapshots to Points: A Visual Analytics Approach to Dynamic Network Exploration. In *IEEE Transactions on Visualization & Computer Graphics*, vol. 22, no. 1, pp. 1–10, Jan. 2016.

27 Lisa Zimmermann, Francesca Zerbato, Kristina Vrotsou, and Barbara Weber. A process-oriented approach to analyze analysts' use of visualizations: Revealing insights into the what, when, and how. *Computer Graphics Forum*, vol. 44, no. 3, 2025.

4.5 Towards Visual Process Analytics for Process Ecosystems

Giovanni Meroni (Technical University of Denmark – Lyngby, DK), Natalia V. Andrienko (Fraunhofer IAIS – Sankt Augustin, DE), Iris Beerepoot (Utrecht University, NL), Claudio Di Ciccio (Utrecht University, NL), Cagatay Turkay (University of Warwick – Coventry, GB), and Tatiana von Landesberger (Universität Köln, DE)

4.5.1 Introduction

The surge in the digitalization of processes has catered for an abundance of recorded data. Process mining successfully leverages this renownedly precious information source for knowledge acquisition and enhancement, thus garnering significant attention in research and industry alike. Digitalization, on the other side, has also spurred the transcend of business landscapes beyond the silos of single processes, departments, or organisations [1]. While favouring the hyperconnection of expertise and knowledge, it has also exposed the limits of classical process mining, whose techniques are typically designed to focus on those silos. Events like the outbreak of pandemics in healthcare, the domino effect of stock market agitations in finance, and the blockades caused by issues along logistic routes in supply chain management, disruptively illustrated the expansion in space and time of events well beyond the boundaries of their locality.

We are experiencing a more and more predominant passage from an architecture of processes [2] to what we call a *process ecosystem*. Recent advances in the scientific literature call for multi-perspective, inter-instance, cross-process approaches [3]. Beyond the technical objectives of efficacy and scalability for this new wave of approaches, we claim in this paper that an expressive, appropriate, effective analysis of data from hyperconnected settings is key to pursuing the ultimate aim of knowledge extraction. To this end, we advocate the integration of process analytics to propel the potential of process mining and significantly contribute to its capability to handle process ecosystems.

4.5.2 Motivating Example

To motivate our work, we consider the organization of a scientific conference. To achieve this, the process of reviewing papers has to be enacted. The process is organized as follows: authors create new submissions by sending their abstract and full paper. The program chair then assigns the papers to review to program committee (PC) members. PC members then review the papers, potentially delegating the task to subreviewers. Once the review deadline expires, PC members begin discussing their reviews with senior PC members. Once the discussion is over, program chairs take the final decision on the paper and, in case of acceptance, the authors submit a camera-ready version of their paper.

This process encompasses several interrelated activities, such as submitting papers, assigning papers, writing reviews, discussing reviews, and taking decisions. Such activities involve different resources, such as papers, reviews, and comments. They are also performed by actors under different roles, such as authors, (senior) PC members, and program chairs. It is also worth noting that the same resource may participate in different activities carried out by different actors, and the same actor may perform different activities, or the same activity multiple times, depending on the involved resource.

To support the review process, the conference organizers rely on a Conference Management System (CMS). Instead, other supporting processes, such as participating in PC meetings and discussions, happen outside the CMS through other channels, such as email or videoconferencing applications. These processes are influenced by and influence the review process, as they all share a subset of actors and resources. Figure 27 shows an Enterprise Architecture [4] model that captures the activities, the objects, the supporting applications, and the roles that the actors have in this example.

Key Goals and Tasks

The steering committee would like to improve the knowledge transfer from the organizers of the previous edition of the conference, to the ones of the current edition. To this aim, the following tasks have been identified:

- Provide suggestions on how to compose the PC, by identifying uncovered research areas and removing actors who did a poor job in reviewing papers.
- Identify bottlenecks that could cause delays in the review process, as well as possible workarounds.
- Identify under which conditions constraints in the process (e.g., a paper is reviewed by at least 4 reviewers, 2 of which are experts in the topic covered by the paper) are violated.
- Ensure that the conference aligns with the goals set by the publisher (e.g., that the acceptance rate is not higher than 30%).

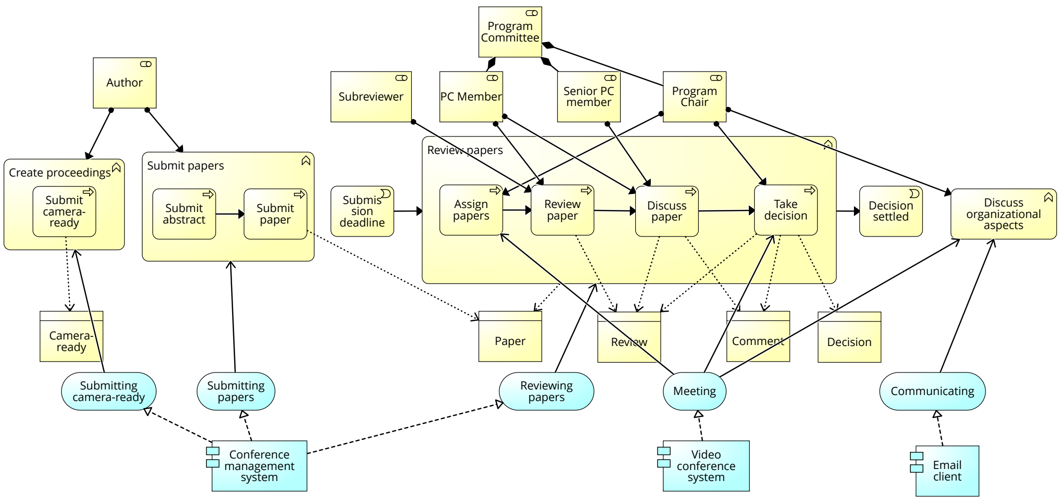


Figure 27 Enterprise Architecture model describing the running example.

4.5.3 Technical Challenges in Process Ecosystem Analysis

- **Multiplicity of perspectives:** Unlike analysis involving a single stakeholder or a particular part of a process, process ecosystems involve processes representing multiple perspectives within a complex system. Analyzing these multiple perspectives require creating multiple representations that enable switching from one perspective to the other seamlessly and with methods allowing cross-perspective analysis.
- **Inter-process interactions:** An inherent challenge with multiple processes in process ecosystems is the diverse range of interactions that can take place within the processes. For instance, processes can be interdependent on one another, could be competing for shared resources or processes can be sub-parts of larger processes. These range of relations require an analyst or a designer to devise bespoke computations or visualizations that can handle a large range of complex relations between processes.
- **Temporal constraints:** Temporal constraints and rules do already require special attention within analysis involving a single process. With multiple processes interacting in diverse ways, the effort to oversee temporal constraints is exacerbated. Analysts require multi-faceted but temporally-aligned representations to be able to concurrently analyze multiple interacting processes.
- **Data heterogeneity and quality issues:** With several processes that needs to be interlinked and analyzed concurrently, the data that captures these process will be diverse and be available in incompatible formats, requiring extensive effort in data processing. With data being captured in multiple levels, the probability of data errors and gaps occurring is also higher.
- **Dynamic evolution:** One inherent character of complex inter-related systems is their continuous evolution. This could mean that dependencies across entities could change, new dependencies could emerge and resource needs and pathways of processes could alter. This requires dynamic representations that enables the monitoring and analysis of evolving systems.

Limitations of Current Process Mining Solutions

Despite the growing maturity of process mining (PM) tools and techniques, significant limitations remain when it comes to analyzing systems composed of multiple interacting processes. In particular, existing PM solutions struggle to provide a comprehensive and time-aware understanding of how such systems evolve.

A key challenge lies in capturing and representing the temporal development of multiple processes and their interactions. Real-world process ecosystems, such as those seen in collaborative workflows, healthcare, logistics, or scientific conference management, involve numerous interdependent processes that run in parallel and influence one another over time. Understanding when and how these interactions occur is essential for diagnosing inefficiencies, anticipating problems, and identifying opportunities for intervention. However, most process mining visualizations are optimized for analyzing single processes in isolation and often focus on control-flow abstractions such as process models or variants.

When time is represented at all, current PM tools typically offer limited support for temporal overview or for tracing the evolution of interactions (other than, e.g., examining the differences between process models such as presented in Figure 28). Temporal data may be aggregated, reduced to average durations, or visualized in Gantt-like charts with limited interactivity and low scalability. This makes it difficult to analyze the unfolding of interactions, to understand the synchrony or asynchrony of processes, or to spot coordination problems that arise due to delays, resource contention, or structural dependencies.

This is where visual analytics (VA) can serve as a powerful complement to traditional PM. Visual analytics provides a rich toolkit for representing and analyzing time-oriented data, enabling interactive exploration of temporal patterns, event sequences, and process trajectories. Techniques such as storyline visualizations, Marey charts, and temporal networks can offer more nuanced perspectives on how multiple processes evolve and influence one another over time. Moreover, VA supports multi-scale and multi-faceted analysis, helping users to shift between overviews and detail, filter and compare temporal patterns, and incorporate contextual or domain-specific knowledge into the analysis.

By integrating temporal visual analytics into process mining, we can advance toward a more holistic understanding of process ecosystems – one that not only captures what processes occur, but also when, how, and in relation to what others.

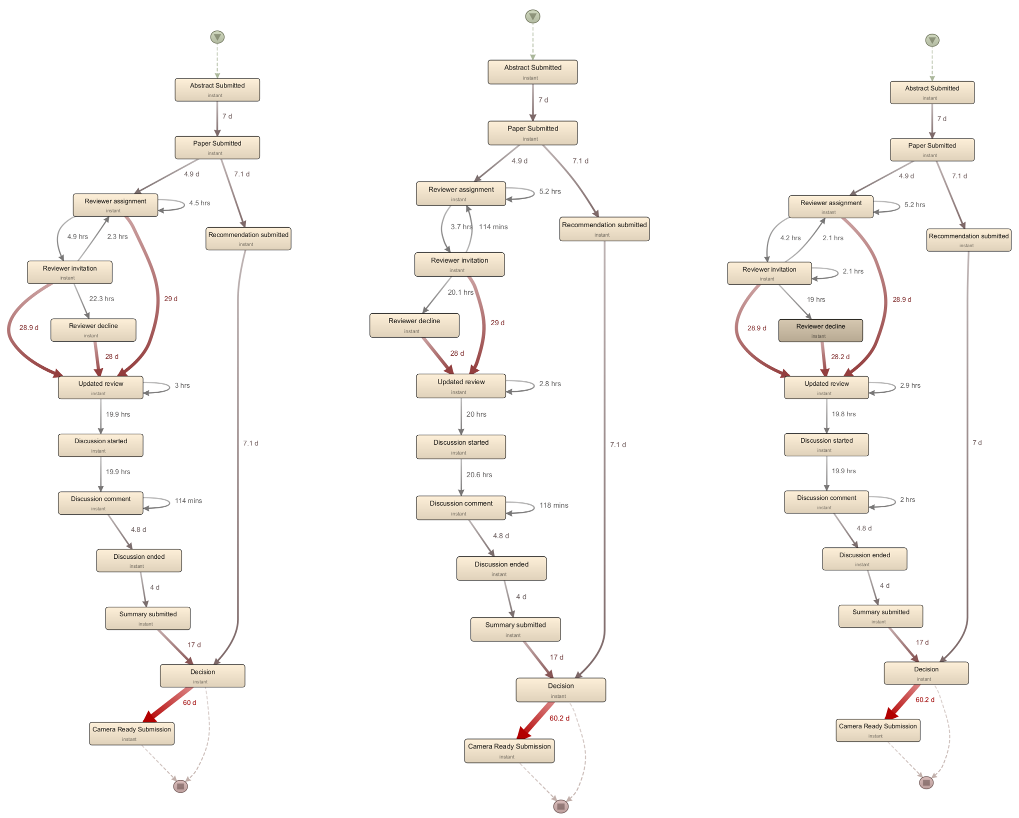


Figure 28 Median duration between activities related to conferences in 2023, 2024, and 2025.

4.5.4 Vision

Effective monitoring and analysis of process ecosystems requires a combination of techniques that can extract structured knowledge from event data and support interactive, human-centered exploration. Process Mining (PM) and Visual Analytics (VA) offer complementary strengths that, when combined, enable a powerful approach to understanding such complex systems.

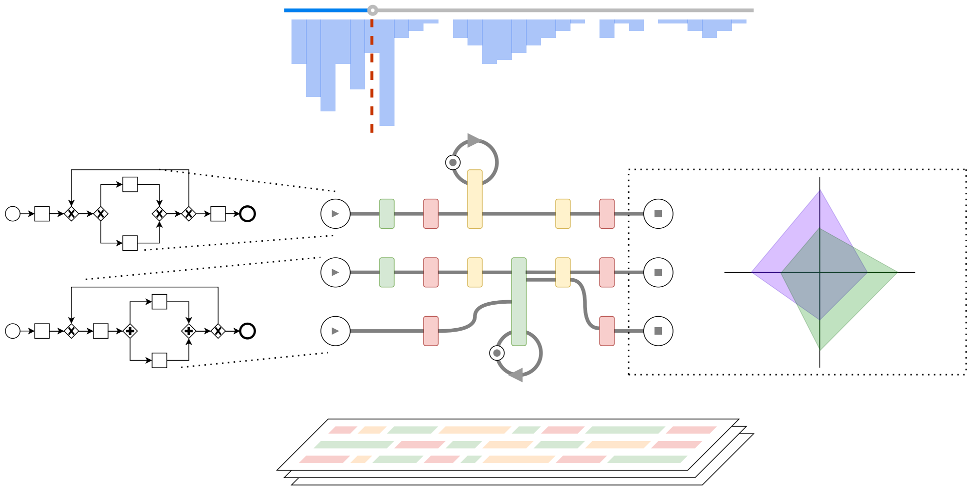


Figure 29 A depiction of the visual approach to process ecosystem analytics.

- **Linking Process Discovery with Interactive Exploration:** PM excels at deriving structured process models from event data, revealing control-flow patterns and performance metrics. VA complements this by enabling users to interactively explore these models and the underlying data, focusing on aspects most relevant to their questions.
- **Integrating Temporal Representations and Analysis:** PM tools provide time-based metrics and can segment process stages temporally. VA enriches this with expressive visual metaphors, such as timelines, storylines, Marey charts, or dynamic graphs, that allow users to perceive the progression, synchronicity, and timing of multiple processes in context.
- **Revealing Inter-Process Interactions:** While PM can identify shared activities or resources, VA supports the visualization of dependencies, influences, and synchronizations across process instances. This helps users reason about how processes interact, support, compete with, or block each other.
- **Supporting Multi-Actor and Multi-Perspective Analysis:** In process ecosystems involving many stakeholders (e.g., coordinators, reviewers, chairs), PM can identify roles and responsibilities. VA allows for flexible filtering and perspective switching, helping stakeholders understand the system from their own or others' viewpoints.
- **Facilitating Monitoring and Timely Intervention:** The combination enables the construction of rich monitoring dashboards where PM provides the structured event flow and derived KPIs, while VA supports intuitive visual layouts and alerts that guide attention to issues such as delays, coordination problems, or resource bottlenecks (see Figure 29).
- **Managing Complexity and Enhancing Interpretability:** PM's algorithmic capabilities scale to large datasets, and VA provides ways to abstract, aggregate, and interactively refine views, allowing users to navigate complexity without losing important details.
- **Enabling Retrospective Learning and Knowledge Externalization:** PM can identify patterns of behavior over historical data, and VA supports sensemaking and storytelling helping analysts communicate findings, compare scenarios, and build shared understanding of systemic dynamics.

4.5.5 Conclusion and Future Work

In this work, we highlighted the need to move beyond traditional, isolated views of processes and embrace the concept of *process ecosystems* – systems of multiple interrelated and interacting processes evolving in parallel. We argued that understanding such ecosystems requires not only the discovery of individual process models but also tools for monitoring, visualizing, and analyzing their temporal development, mutual dependencies, and interactions.

We discussed how the synergy between Process Mining (PM) and Visual Analytics (VA) can address this need. PM offers robust methods for discovering and quantifying processes based on event data, while VA brings in powerful techniques for time-oriented representation, interactive exploration, and multiscale analysis. Time-based visualizations, coordinated multiple views, and interaction techniques such as brushing, filtering, and dynamic level-of-detail provide essential support for making complex process interrelations observable and interpretable.

Using the example of scientific conference organization, we illustrated a process ecosystem and discussed some key goals an visual process analysis approach could support. We also pointed to scalability challenges and the importance of grouping and abstracting processes using clustering based on task-specific similarity metrics.

Future research directions include:

- Developing scalable visual representations that combine aggregate views with detailed process trajectories.
- Designing similarity measures and clustering techniques tailored to different types of process interactions (e.g., synchronization, resource sharing, interference).
- Supporting dynamic grouping and focus+context views to enable exploration of evolving sub-ecosystems.
- Integrating causal inference techniques to understand not only correlations but also influence among processes.
- Creating domain-specific dashboards that bring together PM and VA components to support monitoring and decision making in real-time.
- Conducting empirical studies with domain experts to evaluate usability and effectiveness of the proposed approaches.

By advancing in these directions, we can build more intelligent and adaptive tools for managing complex systems of interacting processes across domains such as healthcare, logistics, manufacturing, and scientific workflows.

References

1 Daniel Beverungen, Joos C. A. M. Buijs, Jörg Becker, Claudio Di Ciccio, Wil M.P. van der Aalst, Christian Bartelheimer, Jan vom Brocke, Marco Comuzzi, Karsten Kraume, Henrik Leopold, Martin Matzner, Jan Mendling, Nadine Ogonek, Till Post, Manuel Resinas, Kate Revoredo, Adela del Río-Ortega, Marcello La Rosa, Flávia Maria Santoro, Andreas Solti, Minseok Song, Armin Stein, Matthias Stierle, and Verena Wolf. Seven paradoxes of business process management in a hyper-connected world. *Business & Information Systems Engineering*, 63(2):145–156, 2021.

2 Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. *Fundamentals of Business Process Management, Second Edition.* Springer, 2018.

3 Dirk Fahland. Process mining over multiple behavioral dimensions with event knowledge graphs. In *Process Mining Handbook*, pages 274–319. Springer, 2022.

4 Marc M. Lankhorst, editor. *Enterprise Architecture at Work – Modelling, Communication and Analysis, Fourth Edition.* Springer, 2017.

5 Wil M. P. van der Aalst. Object-centric process mining: Dealing with divergence and convergence in event data. In Peter Csaba Ölveczky and Gwen Salaün, editors, *Software Engineering and Formal Methods – 17th International Conference, SEFM 2019, Oslo, Norway, September 18-20, 2019, Proceedings*, volume 11724 of *Lecture Notes in Computer Science*, pages 3–25. Springer, 2019.

Participants

- Wolfgang Aigner
FH – St. Pölten, AT
- Gennady Andrienko
Fraunhofer IAIS – Sankt Augustin, DE
- Natalia V. Andrienko
Fraunhofer IAIS – Sankt Augustin, DE
- Iris Beerepoot
Utrecht University, NL
- Andrea Burattin
Technical University of Denmark – Lyngby, DK
- Lena Cibulski
Universität Rostock, DE
- Claudio Di Ciccio
Utrecht University, NL
- Irit Hadar
University of Haifa, IL
- Marie-Christin Häge
Universität Mannheim, DE
- Andreas Kerren
Linköping University, SE
- Philipp Koytek
Celonis Labs GmbH – München, DE
- Zhicheng Liu
University of Maryland – College Park, US
- Giovanni Meroni
Technical University of Denmark – Lyngby, DK
- Silvia Miksch
TU Wien, AT
- Manuel Resinas
University of Sevilla, ES
- Shazia Sadiq
University of Queensland – Brisbane, AU
- Hans-Jörg Schulz
Aarhus University, DK
- Pnina Soffer
University of Haifa, IL
- Christian Tominski
Universität Rostock, DE
- Cagatay Turkay
University of Warwick – Coventry, GB
- Stef Van den Elzen
TU Eindhoven, NL
- Maria-Cruz Villa-Uriol
University of Sheffield, GB
- Tatiana von Landesberger
Universität Köln, DE
- Katerina Vrotsou
Linköping University, SE
- Barbara Weber
Universität St. Gallen, CH
- Peilin Yu
Linköping University, SE
- Francesca Zerbato
TU Eindhoven, NL

Report from Dagstuhl Seminar 25171

Holistic Graph-Processing Systems: Enabling Real-World Scale and Societal Impact

Alexandru Iosup*[1], Ana Lucia Varbanescu*[2], Hannes Voigt*[3], and Jože Rožanec†[4]

1 **VU Amsterdam, NL.** alexandru.iosup@gmail.com
2 **University of Twente – Enschede, NL.** a.l.varbanescu@utwente.nl
3 **Neo4j – Leipzig, DE.** hannes.voigt@neo4j.com
4 **Jozef Stefan Institute – Ljubljana, SI.** jmrozanec@gmail.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 25171, "Holistic Graph-Processing Systems: Enabling Real-World Scale and Societal Impact".

Motivated by the need to tackle the challenges that massive and complex data production and consumption bring to our interconnected, digital world, this seminar focused on large-scale graph processing as a systematic approach to transform these challenges into opportunities. Graphs provide a universal mathematical abstraction for such data, and they already influence various sectors – such as logistics, drug discovery, or fraud detection. However, we have only begun to realize their potential. Nevertheless, the benefits of graph processing could be canceled out by the rapid increase in data scale and diversity, as well as the increasing complexity in developing, executing, and sharing graph-based algorithms and workflows. The emerging field of graph processing systems promises to tackle these challenges. To make such systems effective and efficient, and facilitate their adoption, we need holistic approaches to cope with data transformation and ingestion, workload and system dynamics, high-tier graph programming and co-design with the platform, the emerging computing continuum, and domain-specific needs, among others.

Our seminar explored the symbiosis of graph systems, machine learning, and network science by bringing together researchers, developers, and practitioners actively working on these topics with a focus on graphs. The seminar featured a mix of invited talks, expert panels, and focused discussion groups. The report documents these different elements, summarizes the main findings, and identifies the open problems and challenges that we will tackle next as a joint community.

Seminar April 21–25, 2025 – https://www.dagstuhl.de/25171

2012 ACM Subject Classification Computing methodologies → Distributed computing methodologies; Computing methodologies → Machine learning; Hardware → Emerging technologies; Information systems → Data management systems; Mathematics of computing → Graph theory

Keywords and phrases digital continuum choreography, graph processing optimization, machine learning on graphs, massive graphs, sustainable distributed graph processing

Digital Object Identifier 10.4230/DagRep.15.4.79

---

* Editor / Organizer
† Editorial Assistant / Collector

Holistic Graph-Processing Systems: Enabling Real-World Scale and Societal Impact, *Dagstuhl Reports*, Vol. 15, Issue 4, pp. 79–91
Editors: Alexandru Iosup, Ana Lucia Varbanescu, Hannes Voigt, and Jože Rožanec
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Ana Lucia Varbanescu (University of Twente – Enschede, NL)
Alexandru Iosup (VU Amsterdam, NL)
Jože Rožanec (Jozef Stefan Institute – Ljubljana, SI)
Hannes Voigt (Neo4j – Leipzig, DE)

In today's digital landscape, complexity grows with increasing data volume and degree of interconnection. A suitable data abstraction is crucial for comprehending and navigating this dense network of connections. Starting from Euler's pioneering work on The Bridges of Konigsberg in 1735, graphs have steadily evolved as a robust and adaptable conceptual framework. Graphs are universal representations of concepts, where nodes are markers for distinct entities and edges delineate their interrelations, further enriched with detailed annotations when necessary. Graphs are successful in various domains, like bioinformatics, e-commerce, logistics and transportation networks, urban planning, and even pandemic analysis or vaccine development (e.g., during COVID-19).

Although graphs enable complex analysis and decision-making, processing graphs to understand real-world phenomena and to solve real-world problems raises many challenges that threaten to keep graph processing intractable for the current generation of applications. For example, creating graphs from massive data sources or with generative approaches poses multiple challenges, including volume, velocity, and variety. Furthermore, the variability and irregularity of graphs and their processing algorithms challenge the use of established heterogeneous hardware, general-purpose big data solutions, or computing continuum mechanisms. Continuous operation on (streaming) graphs requires new techniques for adaptivity and optimization – e.g., provisioning, allocation, elastic scaling, migration, offloading, partitioning, consolidation, and caching – to be combined across large-scale information and communication technology infrastructure.

Addressing these challenges for graph processing workflows at real-world scale and with societal impact requires a holistic approach that leverages the expertise and synergies of multiple communities related to graph processing. Our seminar did provide a unique opportunity for encounters between these distinct communities, each addressing graph processing at scale. We brought together three essential communities: distributed, parallel, and cluster computing, machine learning for, on, and with graphs, and social and information networks. We therefore facilitated a better understanding of each community's challenges regarding graph processing, and promoted a synergic relationship to shape holistic, actionable knowledge of graph processing.

In search of the holistic view of massive-scale graph processing, the seminar featured five topics of discussion: (1) massive graph creation with generative and analytical approaches, (2) graph processing algorithms and workflows, (3) graph operations across the digital continuum, (4) adaptivity and optimization to ensure performance, scalability, and sustainability, and (5) applications at real-world scale with near-term societal impact. For each of these topics, the seminar dedicated half a day, featuring expert talks, an expert panel, and break-out session to facilitate discussions. All these activities shaped a comprehensive vision and sketched a roadmap to guide research in graph processing for the upcoming years.

Our main goals were: (i) to establish a uniform vocabulary across communities for issues related to graph processing, (ii) to identify key open graph-processing challenges and opportunities across communities along with ideas for long-term research, (iii) to co-design

a holistic approach (blueprint, reference architecture, experimental methodology) to graph processing in the digital continuum, and (iv) to identify flagship applications for holistic graph processing with real-world scale and societal impact.

Our future plans include high-visibility collaboration and dissemination, with mechanisms such as roadmap white-papers, networks of excellence, EU-level projects, and follow-up Dagstuhl Seminars.

2 Table of Contents

3 Overview of Talks

3.1 Democratizing Large-Scale Graph Analytics: From Supercomputing to Societal Impact

David A. Bader (NJIT – Newark, US)

In this talk, Distinguished Professor David A. Bader explores the evolution and impact of large-scale graph analytics, from his pioneering work in Linux supercomputing to today's democratization of massive data science capabilities. The presentation highlights how the open-source Arachne framework, built on Arkouda, enables researchers and organizations to process and analyze graphs containing terabytes of data through an accessible Python interface, while the heavy computational work occurs on powerful backend systems.

Bader discusses three critical application domains of this technology: national security (detecting malicious network activity through relationship patterns), computational neuroscience (analyzing connectomes containing millions of neurons and synapses), and scientometrics (mapping research collaboration networks). These examples demonstrate how graph analytics can solve complex problems that were previously inaccessible due to computational limitations.

The talk emphasizes the parallel between Bader's 1998 development of the Linux-based supercomputer – now the architecture for 100% of the world's top supercomputers with an estimated economic impact of $100 trillion – and his current mission to democratize access to sophisticated graph analytics capabilities. Through open-source tools and frameworks, Bader's work continues to bridge the gap between cutting-edge computing resources and real-world applications, making powerful data analysis accessible to researchers and practitioners across disciplines.

3.2 Graphs and Large Language Models: Vision or Reality ?

Angela Bonifati (Lyon 1 University & IUF, FR)

This talk explores the interplay of graph data models and Large Language Models (LLMs), addressing whether their integration is a current reality or an emerging vision. Graphs, as intuitive and powerful abstractions for modeling interconnected data, underpin a wide array of applications from social networks to scientific data analysis. The presentation dives into property graph transformations, emphasizing the need for expressive, declarative tools to enable robust, composable rule-based transformations and graph creation. The second part examines the role of LLMs in generating transformation rules and consistency constraints for property graphs. It discusses techniques for encoding graphs for LLM processing, compares performance across models (LLaMA-3, Mixtral), and highlights challenges in rule quality, scalability, and semantics. The talk concludes by outlining the promise of retrieval-augmented generation (RAG) for enhancing LLM capabilities with external graph-based knowledge, offering a roadmap for integrating symbolic and neural reasoning in future systems.

3.3 Structured data has a lot of value and we need to learn how to tap into it

Michael Cochez (VU Amsterdam, NL)

Recent advances in machine learning, especially with language models, have transformed many fields, dramatically raising expectations for artificial intelligence systems. Yet, these advances have also exposed a critical weaknesses: a lack of reliability and a need for a ridiculous amount of training data. In contrast, traditional AI methods offer more dependable but less flexible solutions, often limiting their broad applicability.

I am working in the intersection of these approaches which is sometimes called neuro-symbolic AI, particularly with knowledge graphs (KG). I investigate how graph neural networks can bridge the gap between discrete KGs and the statistical world of machine learning, enabling more robust and scalable solutions for tasks like structured and natural language question answering. My work often utilizes data from the medical and biomedical domains. During the seminar, I hope to discuss two sharpen my vision on the future of graph research. Among others, I am interesting in

1. Improving ML Trustworthiness: Many popular ML models are not trustworthy. Can the addition of graph data make a difference? Can we get to a point where we can provide formal guarantees (e.g., error bounds) to enhance trust in graph ML models?
2. Addressing issues in Multi-source KGs: There are challenges when attempting learning on unbalanced KGs, where a larger, potentially biased graph overshadows smaller, more specific ones. How could we overcome this?
3. Get to scalable graph ML models using both smart use of hardware and knowledge on what the content of the graph is.

3.4 From Graphs to Design Automation: Custom Systems for Data Analytics

Antonino Tumeo (Pacific Northwest National Lab. – Richland, US)

This talk discusses several co-design approaches to enable accelerated graph analytics across the whole continuum of computing. The talk first identifies the key issues in large scale graph processing, then reviews some algorithm-hardware co-design works to enable parallel graph processing on leadership computing systems with thousands of graphic processing units (GPUs). We then provide a full stack case study, the Graph Engine For Multithreaded Systems, which implements a scalable in-memory graph database for distributed high-performance computing clusters. The talk then briefly discusses the role of C++ distributed data structures libraries to design complex and malleable data models (including property graphs) and parallel (graph) algorithms. Finally, we discuss how to generate specialized accelerators for irregular applications and graph algorithms starting from shared memory parallel code descriptions with the SODA Synthesizer.

3.5 Two Graph-Related Topics

M. Tamer Özsu (University of Waterloo, CA)

I talk about two topics one of which focuses on graph processing algorithms and techniques, and the other on an interesting use of graphs. The first topic is querying streaming graphs which is difficult because it combines two difficult issues, namely the unboundedness and high velocity of arrivals of streaming data and the difficulty of graph querying. In this space we systematically analyzed query models with clear semantics, developed streaming graph algebra primitives that allowed logical plan generation. Transformation rules were developed to manipulate the logical plans as well as a cost-based optimization framework to find the better plans. All of this was implemented as a prototype over Calcite.

The second topic is graph-based vector indexing. In this space, vectors that are generated by an embedding model form a graph dataset over which an approximate nearest neighbour search (ANNS) is performed. To aid search, indexes are built on this dataset. Graph-based indexes are those that treat each vector as a vertex with edges representing relationships between these vectors. The search takes a query point, starts from an entry point vertex in this index and finds the approximate nearest neighbours of the query point. These indexes can be built using either a refinement approach (e.g., NNDescent) or an iterative approach (e.g., HNSW). Refinement-based approaches start with a random graph (of vectors) and iterativey improve the graph. They have fast index construction time, but lower queries-per-second than iterative approaches and they cannot do incremental inserts of new vectors. On the other hand, iterative approaches insert vectors one-at-a-time, performing an ANNS to determine which other vectors it should be connected to. They have high construction cost, but good query performance and can handle incremental inserts. Our work involves an index, called MIRAGE, that combines the good characteristics of both approaches. It has better construction time and higher queries-per-second.

4 Working groups

4.1 Holistic graph operations at scale, across the digital continuum

Andrea Bartolini (University of Bologna, IT), Duncan Bart (University of Twente – Enschede, NL), Dante Niewenhuis (VU Amsterdam, NL), Jože Rožanec (Jozef Stefan Institute – Ljubljana, SI), Daniël ten Wolde (CWI – Amsterdam, NL), Antonino Tumeo (Pacific Northwest National Lab. – Richland, US), and Nikolay Yakovets (TU Eindhoven, NL)

Introduction. Currently, we lack of a holistic view on graph processing at scale, that would comprehend the whole digital continuum. Therefore, it is key to identify what siloes do currently exist and the reasons behind their persistence as to design mechanisms that would enable graph operations at scale across the digital continuum.

Key ideas. Current approaches are fragmented, and there is no single graph abstraction that can express both temporal and spatial metadata. Furthermore, graph workflows are siloed, with few mechanisms that enable interoperability and expressiveness. A gap also exists between graph representations and how these could be mapped to hardware for efficient processing (e.g., mapping property graphs to GPUs is not trivial). A graph-first framework resembling a water cycle could be designed, considering (i) data streams and lakes deployed over a storage continuum and natively handling data bitemporality and associated with a semantic catalogue, (ii) mechanisms for scalable interfacing such as declarative queries that allow to express workflows and graph views, which are then translated by a (iii) compiler to stored procedures and executed on (iv) heterogueneous hardware across a compute continuum.

Conclusions. There is consensus that we need (a) a standard graph format that would reduce fragmentation and enable end-to-end graph processing, (b) a standardized representation as to allow to describe graph problems, (c) composable abstractions (frameworks that decouple graph modelling from performance concerns) providing a shared vocabulary between user needs and algorithm and system design, and (d) reimagine graph-processing hardware and systems (with a focus on sparse matrix computation and ensuring the storage and compute are co-located as to reduce the movement of data).

4.2 Applications at a real-world scale with near-term societal impact

Aydin Buluc (Lawrence Berkeley National Laboratory, US), Duncan Bart (University of Twente – Enschede, NL), Stefania Dumbrava (ENSIIE & TélécomSudParis, FR), Kamesh Madduri (Pennsylvania State University – University Park, US), Dante Niewenhuis (VU Amsterdam, NL), Jože Rožanec (Jozef Stefan Institute – Ljubljana, SI), Daniël ten Wolde (CWI – Amsterdam, NL), and Ana Lucia Varbanescu (University of Twente – Enschede, NL)

Introduction. To identify the future challenges and opportunities in graph processing, it is essential to identify what real-world applications with societal impact will look like. Understanding the requirements and characteristics of such applications helps identify the modeling and design choices required.

Key ideas. When developing applications at a real-world scale with near-term societal impact, it is crucial to consider how data is modeled into a graph, as this determines the approaches that become feasible for use and processing those graphs downstream. Key concerns related to (a) what methodologies are appropriate to extract data and create a graph, (b) how data should be represented in the graph – also addressing privacy and concerns where appropriate, (c) how graph systems should be designed to support multiple modeling choices, optimizations, and infrastructure capabilities without resulting in vendor lock-in, and (d) what approaches could be used for cleansing, completing, fixing, and processing those graphs. Promising application domains that could benefit from graph-based approaches include artificial data generation, healthcare, transportation, and logistics.

Conclusions. There is no perfect graph system. Data modeling choices, system optimizations, and infrastructure decisions depend heavily on application needs. Privacy-preserving modeling, federation, and distributed graph processing remain open research challenges. Future directions include better system/application co-design, reducing vendor lock-in, and making knowledge graph construction, processing, and graph-based applications development more accessible.

4.3 Adaptivity and optimization to ensure performance, scalability, and sustainability

Stefania Dumbrava (ENSIIE & TélécomSudParis, FR), Duncan Bart (University of Twente – Enschede, NL), Peter A. Boncz (CWI – Amsterdam, NL), Florina M. Ciorba (Universität Basel, CH), Dante Niewenhuis (VU Amsterdam, NL), Jože Rožanec (Jozef Stefan Institute – Ljubljana, SI), and Daniël ten Wolde (CWI – Amsterdam, NL)

Introduction. While performance, scalability, and sustainability are regarded as relevant, efforts in this direction remain fragmented, making it difficult to consolidate them in frameworks and systems that would benefit the end users.

Key ideas. Adaptivity and optimization are required to ensure performance, scalability, and sustainability. When speaking about scalability, we can consider weak scalability (scape up) or strong scalability (speed up), scale out (add more machines) or scale up (more resources inside a single machine). When contextualizing scalability w.r.t. graphs we consider scaling data (graph grows) and scaling workload (operations on graphs grow). When speaking about sustainability, we consider energy efficiency (brown, green energy), power consumption, carbon (embodied and operational), cost, required rare earth materials, and water consumption (for building and cooling). Finally, when considering performance, we can consider algorithmic or system performance, or performance related to productivity and usability of tools and systems. When considering algorithmic and system performance, we can characterize it with metrics that refer to execution time, response time, throughput, resource utilization, and performance portability. When considering performance from the productivity point of view, we can characterize it considering lines of code or external dependencies, but also through usability criteria, such as ease of deployment or ease of use of a given tool or system. When analyzing adaptivity and optimization to ensure performance, scalability, and sustainability, we observe that the graph ecosystem is fragmented. Many frameworks support only subsets of functionality and there is little convergence. That fragmentation is evident when considering supported functionality across graph tools and systems, where there is no single definition of correctness. Furthermore, such fragmentation makes it more difficult to select the right algorithm implementation (hardware and software) when considering complex algorithms.

Conclusions. To reduce fragmentation across the graph community, we propose focusing on standards rather than implementation details. Such standards could, in certain cases, replace the notion of correctness by specifying expectations over an algorithmic execution of system output. Furthermore, high-level abstractions, such as DSLs, could be useful to increase code portability and optimization across architectures. To ensure the best algorithms are available to everyone, it is necessary to promote best practices that provide algorithm and model implementations are usable beyond an academic endeavor. Furthermore, we require a community effort to catalog and benchmark algorithms, in order to understand their performance and available implementations. Artificial Intelligence can be leveraged to determine the optimal hardware and software implementation for a particular algorithm. We would ideally strive for models trained on small graphs whose inference can be extrapolated to large graphs. Explainable Artificial Intelligence can provide additional insights into understanding why a particular setup is suitable for a specific task.

4.4 Graph processing algorithms and workflows

Johannes Langguth (Simula Research Laboratory – Oslo, NO), Duncan Bart (University of Twente – Enschede, NL), Sanjukta Bhowmick (University of North Texas, US), Dante Niewenhuis (VU Amsterdam, NL), Jože Rožanec (Jozef Stefan Institute – Ljubljana, SI), Ingo Scholtes (Universität Würzburg, DE), and Daniël ten Wolde (CWI – Amsterdam, NL)

Introduction. Graph processing and graph applications have similar but different concerns. Nevertheless, ensuring synergies between those communities can help advance the field of graph processing at a higher pace.

Key ideas. The application and graph processing communities are similar, but have different concerns. The main objective of the graph database community is data management, while applications focus on performance. This discrepancy makes it challenging for the communities to align. For example, people are familiar with GraphDBs, but cannot extract the information they need due to low performance. Additionally, many communities work on the same challenges without considering progress made on them in a different community. Instead, we should combine forces. However, interoperability between the different communities is non-trivial. Creating a complete overarching system that satisfies all communities is challenging due to the variety in needs.

Conclusions. We need to focus on improving usability for all users. Current systems lack intuitive abstractions, especially for non-experts. We need to combine forces and avoid addressing problems that might have been solved in a different community. Additionally, we should focus more on data quality. Data cleaning is still a widespread bottleneck, but is understudied. There is also a responsibility mismatch: data producers have little incentive to clean data and what does clean data mean could be different depending on the downstream requirements.

4.5 Massive graph creation with generative and analytical approaches

Nikolay Yakovets (TU Eindhoven, NL), Bogdan Arsintescu (Microsoft Corp. – Mountain View, US), Duncan Bart (University of Twente – Enschede, NL), Jože Rožanec (Jozef Stefan Institute – Ljubljana, SI), Juan F. Sequeda (data.world – Austin, US), and Daniël ten Wolde (CWI – Amsterdam, NL)

Introduction. Real-world data is oftentimes not directly obtainable in a graph shape/format. This data can be transformed into a graph using a schema or ontology. In some cases, graphs are needed that cannot be obtained from real-world data. In these cases, the graphs must be synthetically generated.

Key ideas. Graph generation can be divided into two categories. Creating graphs from existing data sources (with an inherent graph structure, like social networks, or other forms such as time series data) or synthetic graph generation. Creating graphs from existing data is essential for real-world applications. A significant challenge in generating graphs from existing data sources is the data modeling problem, i.e., defining a suitable schema/ontology. Creating a good-quality graph is essential for the quality and reliability of graph applications. However, there is currently no clear definition of what constitutes a "good" graph, which also heavily depends on the specific application at hand. Synthetic graph generation is crucial for benchmarking or stress testing applications under immense computational loads, and it can also be critical in ensuring anonymity when handling sensitive data that could identify individuals, such as diseases that affect only a small number of people. Generating synthetic graphs that resemble real-world graphs for a specific use case is a significant challenge to solve.

Conclusions. A catalogue should be created with metrics and qualities that distinguish between "good" and "bad" graphs extracted from real data and ontologies. This could include credibility/traceability, entropy, or statistical properties (degree distribution, clustering, and path lengths, among others). We consider automatic ontology detection is a crucial research direction for streamlining graph generation from real-world data without human intervention. Generative AI could be a promising research direction for the generation of synthetic graphs.

5 Open problems

5.1 Open problems regarding Holistic Graph-Processing Systems to Enable Real-World Scale and Societal Impact

Jože Rožanec (Jozef Stefan Institute – Ljubljana, SI)

The recent seminar successfully identified and delineated several open problems within the domain of holistic graph-processing systems. Addressing these challenges is critical for enabling systems that can operate at real-world scale and maximize societal impact. From the issues discussed, we have prioritized and selected five problems of scientific and engineering importance, which are briefly described below.

- **Application-centric graph usage matrix:** the goal is to establish a structured analytical framework to systematically characterize graph utilization across diverse application domains, drawing inspiration from the foundational work of Asanovic et al. [1]. This analysis must extend beyond a mere quantification of how intensely are graph technologies utilized and must rigorously assess usage across four critical dimensions: (i) graph generative approaches and graph analytics, (ii) graph processing algorithms and workflows, (iii) graph operations at scale across the digital continuum, and (iv) adaptivity and optimization to ensure performance, scalability, and sustainability. Furthermore, a quantitative characterization should be developed to map applications across the trade-off space defined by accuracy, performance, and portability. This will facilitate the creation of a workflow model to guide the selection of appropriate algorithms. The matrix will be complemented by a curated, diverse collection of exemplars: documented instances of successful algorithm-implementation pairs focusing on algorithmic building blocks that drive high performance, productivity, and portability within real-world application contexts.

- **Workload abstraction and scalable reference architecture:** building upon the insights derived from the application-centric graph usage matrix, the objective is to define formal workload abstractions (Basic Graph Operations). These Basic Graph Operations must be composable elements for the construction of complex algorithms, workflows, and scalable, interacting data processing pipelines. Furthermore, we propose the development of a reference architecture for scalable graph processing. This architecture must leverage the defined workload abstractions and directly address the needs identified by the application-centric graph usage matrix. The design must weight distributed computing architectures and single-node architectures, considering their respective strengths and weaknesses and educate how processing demands can be framed to achieve the desired outcomes.
- **The Memex Architecture: integrating graphs, metalearning, and causality:** we aim to lay the foundation for an architecture that would allow to build a holistic knowledge graph and leverage it for self-improvement. The knowledge graph will semantically link information regarding system execution processes, resource performance metrics, identified system anomalies, underlying infrastructure properties, and crucially, causal relationships among these elements. We envision that this architecture can be leveraged for metalearning, enabling the system to continuously self-optimize and improve its operational efficiency over time.
- **Hardware acceleration and graph compression techniques:** a systematic analysis is required to determine the short-term and long-term implications of emerging hardware accelerators on the efficiency of holistic graph operations. Concurrently, there is a necessity to catalogue and evaluate graph compression techniques that can be efficiently processed by these accelerators.
- **Scientific benchmarking standardization:** perceive a significant opportunity to establish a new generation of rigorous, scientifically grounded graph processing benchmarks. These benchmarks could be built upon the foundations laid by the application-centric graph usage matrix, the formal characterization of algorithms, and the capabilities of emerging accelerators. The primary goal is to introduce a transparent methodology for comparing and contrasting algorithms, datasets, processing workloads, and the heterogeneous underlying computing infrastructure used for their execution.

As an academic and professional community focused on advancing holistic graph-processing systems, we consider addressing these open problems is fundamental to realizing the next generation of scalable and sustainable systems capable of generating significant real-world and societal-scale impact.

References

1 Asanovic, Krste and Bodik, Ras and Catanzaro, Bryan and Gebis, Joseph and Husbands, Parry and Keutzer, Kurt and Patterson, David and Plishker, William and Shalf, John and Williams, Samuel Webb. *The landscape of parallel computing research: A view from berkeley*. 2006

Participants

- Bogdan Arsintescu
Microsoft Corp. – Mountain View, US
- David A. Bader
NJIT – Newark, US
- Duncan Bart
University of Twente – Enschede, NL
- Andrea Bartolini
University of Bologna, IT
- Sanjukta Bhowmick
University of North Texas, US
- Peter A. Boncz
CWI – Amsterdam, NL
- Angela Bonifati
Lyon 1 University & IUF, FR
- Aydin Buluc
Lawrence Berkeley National Laboratory, US
- Kuan-Hsun Chen
University of Twente – Enschede, NL
- Florina M. Ciorba
Universität Basel, CH
- Michael Cochez
VU Amsterdam, NL
- Khuzaima Daudjee
University of Waterloo, CA
- Stefania Dumbrava
ENSIIE & TélécomSudParis, FR
- Aaron Eberhart
metaphacts – Walldorf, DE
- Brian Elvesaeter
SINTEF – Oslo, NO
- Reza Farahani
Alpen-Adria-Universität Klagenfurt, AT
- Peter Haase
Metaphacts – Walldorf, DE
- Kathrin Hanauer
Universität Wien, AT
- Alexandru Iosup
VU Amsterdam, NL
- Johannes Langguth
Simula Research Laboratory – Oslo, NO
- Michelle Li
Harvard University – Boston, US
- Andrew Lumsdaine
Pacific Northwest National Laboratory – Seattle, US & University of Washington – Seattle, US
- Kamesh Madduri
Pennsylvania State University – University Park, US
- Timothy G. Mattson
Human Learning Group – Ocean Park, US
- Martin Molan
Comtrade AI – Ljubljana, SI
- Dante Niewenhuis
VU Amsterdam, NL
- M. Tamer Özsu
University of Waterloo, CA
- Radu Prodan
Alpen-Adria-Universität Klagenfurt, AT
- Matei Ripeanu
University of British Columbia – Vancouver, CA
- Jože Rožanec
Jozef Stefan Institute – Ljubljana, SI
- Ingo Scholtes
Universität Würzburg, DE
- Daniel Thilo Schroeder
SINTEF – Oslo, NO
- Juan F. Sequeda
data.world – Austin, US
- Daniël ten Wolde
CWI – Amsterdam, NL
- Antonino Tumeo
Pacific Northwest National Lab. – Richland, US
- Ana Lucia Varbanescu
University of Twente – Enschede, NL
- Laurentiu Vasiliu
Peracton – Galway, IE
- Hannes Voigt
Neo4j – Leipzig, DE
- Nikolay Yakovets
TU Eindhoven, NL

Report from Dagstuhl Seminar 25172

Information Exchange in Software Verification

Dirk Beyer[*1], **Marieke Huisman**[*2], **Jan Strejček**[*3], **and Heike Wehrheim**[*4]

1 LMU Munich, DE. dirk.beyer@sosy.ifi.lmu.de
2 University of Twente – Enschede, NL. m.huisman@utwente.nl
3 Masaryk University – Brno, CZ. strejcek@fi.muni.cz
4 Universität Oldenburg, DE. heike.wehrheim@uni-oldenburg.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 25172 *Information Exchange in Software Verification*. The term "software verification" refers to the procedure of deciding the correctness of software with respect to (user-supplied or predefined) specifications. In general, software verification is an undecidable problem. Despite this undecidability, software verification is a very active research field with contributions of researchers from several areas such as theorem proving, deductive verification, static analysis, and automatic verification. The analysis techniques developed in these subareas are often complementary with respect to the type of software and specifications they can efficiently handle.

The objective of this Dagstuhl Seminar was to bring together people working in these different subareas to discuss and advance ways of having tools and techniques cooperate on the task of software verification.

Seminar April 22–25, 2025 – https://www.dagstuhl.de/25172
2012 ACM Subject Classification Software and its engineering → Formal methods; Theory of computation → Logic and verification
Keywords and phrases Competitions and Benchmarks, Data-Flow Analysis, Deductive Verification, Formal Verification, Model Checking
Digital Object Identifier 10.4230/DagRep.15.4.92
Edited in cooperation with Zsófia Ádám and Paulína Ayaziová

1 Executive Summary

Dirk Beyer (LMU München, DE)
Marieke Huisman (University of Twente – Enschede, NL)
Jan Strejček (Masaryk University – Brno, CZ)
Heike Wehrheim (Universität Oldenburg, DE)

The term "software verification" refers to the procedure of deciding the correctness of software with respect to (user-supplied or predefined) specifications. In general, software verification is an undecidable problem. Despite this undecidability, software verification is a very active research field with contributions of researchers from several areas such as theorem proving, deductive verification, static analysis, and automatic verification. The analysis techniques developed in these subareas are often complementary with respect to the type of software and specifications they can efficiently handle.

* Editor / Organizer

The objective of this Dagstuhl Seminar was to bring together people working in these different subareas to discuss and advance ways of having tools and techniques cooperate on the task of software verification. Specific focus was given to the exchange of information between tools. In the seminar, we spent one morning session on talks about existing formats for exchange and a second morning session with talks on existing cooperations. The afternoon sessions and the third morning was dedicated to breakout groups on specific topics. In particular, during breakout groups we discussed the following questions:

- Which kind of information can be exchanged?
- How can we make tools provide their (partial) results about program correctness to other tools?
- How can we make tools use (partial) results of other tools during their analysis?
- Which formats are adequate for information exchange?

As an outcome of the discussions, we agreed on new experiments for more advanced exchange of information between tools for deductive verification and software model checking. We have also outlined several new formats for information exchange, in particular an intermediate format for programs and a format for exchanging information about abstraction precision. Action items for further development towards these formats and in other directions have been formulated and agreed by participants.

2 Table of Contents

3 Overview of Talks

3.1 StaRVOOrS – The Force Awakens Again

Wolfgang Ahrendt (Chalmers University of Technology – Göteborg, SE)

Joint work of Wolfgang Ahrendt, Jesús Mauricio Chimento, Gordon J. Pace, Gerardo Schneider
Main reference Wolfgang Ahrendt, Jesús Mauricio Chimento, Gordon J. Pace, Gerardo Schneider: "Verifying data- and control-oriented properties combining static and runtime verification: theory and tools", Formal Methods Syst. Des., Vol. 51(1), pp. 200–265, 2017.
URL https://doi.org//10.1007/S10703-017-0274-Y

Static verification techniques are used to analyse and prove properties about programs before they are executed. Many of these techniques work directly on the source code and are used to verify data-oriented properties over all possible executions. The analysis is necessarily an over-approximation as the real executions of the program are not available at analysis time. In contrast, runtime verification techniques have been extensively used for control-oriented properties, analysing the current execution path of the program in a fully automatic manner. In this talk, we present a novel approach in which data-oriented and control-oriented properties may be stated in a single formalism amenable to both static and dynamic verification techniques. The specification language we present to achieve this that of ppDATEs, which enhances the control-oriented property language of DATEs, with data-oriented pre/postconditions. For runtime verification of ppDATE specifications, the language is translated into a DATE. We give a formal semantics to ppDATEs, which we use to prove the correctness of our translation from ppDATEs to DATEs. We show how ppDATE specifications can be analysed using a combination of the deductive theorem prover KeY and the runtime verification tool LARVA. Verification is performed in two steps: KeY first partially proves the data-oriented part of the specification, simplifying the specification which is then passed on to LARVA to check at runtime for the remaining parts of the specification including the control-oriented aspects. We show the applicability of our approach on two case studies.

3.2 FM-Tools: Find, Use, and Conserve Tools for Formal Methods

Dirk Beyer (LMU Munich, DE)

Main reference Dirk Beyer: "FM-Tools: Find, Use, and Conserve Tools for Formal Methods". 2024
URL https://www.sosy-lab.org/research/pub/2024-Podelski65.Find_Use_and_Conserve_Tools_for_Formal_Methods.pdf

The research area of formal methods has made enormous progress in the last 20 years, and many tools exist to apply formal methods to practical problems. Unfortunately, many of these tools are difficult to find and install, and often they are not executable due to missing installation requirements. The findability and wide adoption of tools, and the reproducibility of research results, could be improved if all major tools for formal methods were conserved and documented in a central repository of tools for formal methods (cf. FAIR principles).

This paper describes a solution to this problem: Collect and maintain essential data about tools for formal methods in a central repository, called FM-Tools, which is available at `https://gitlab.com/sosy-lab/benchmarking/fm-tools`. The repository contains metadata, such as which tools are available, which versions are advertized for each tool, and what command-line arguments to use for default usage. The actual tool executables are stored in tool archives at Zenodo, and for technically deep documentation, references point to archived publications or project web sites. Two communities, which are concerned with software verification and testing, already adopted the FM-Tools repository for their comparative evaluations.

3.3 Contract-LIB: A Proposal for a Common Interchange Format for Software System Specification

Gidon Ernst (LMU Munich, DE), Mattias Ulbrich (KIT – Karlsruher Institut für Technologie, DE)

Joint work of Gidon Ernst, Wolfram Pfeifer, Mattias Ulbrich
Main reference Gidon Ernst, Wolfram Pfeifer, Mattias Ulbrich: "Contract-LIB: A Proposal for a Common Interchange Format for Software System Specification", in Proc. of the Leveraging Applications of Formal Methods, Verification and Validation. Specification and Verification - 12th International Symposium, ISoLA 2024, Crete, Greece, October 27-31, 2024, Proceedings, Part III, Lecture Notes in Computer Science, Vol. 15221, pp. 79–105, Springer, 2024.
URL https://doi.org//10.1007/978-3-031-75380-0_6

Interoperability between deductive program verification tools is a well-recognized long-standing challenge. In this paper we propose a solution for a well-delineated aspect of this challenge, namely the exchange of abstract contracts for possibly stateful interfaces that represent modularity boundaries. Interoperability across tools, specification paradigms, and programming languages is achieved by focusing on abstract implementation-independent behavioral models. The approach, called Contract-LIB in reminiscence of the widely-successful SMT-LIB format, aims to standardize the language over which such contracts are formulated and provides clear guidance on its integration with established methods to connect high-level specifications with code-level data structures. We demonstrate the ideas with examples, define syntax and semantics, and discuss the rationale behind key design decisions.

3.4 Proofs on Inductive Predicates in Why3

Jean-Christophe Filliâtre (CNRS – Gif-sur-Yvette, FR)

Joint work of Jean-Christophe Filliâtre, Andrei Paskevich, Henri Saudubray
URL https://www.why3.org/

This talk presents an extension of the Why3 tool to enable proofs by induction on instances of inductive predicates. It consists of a new pattern matching construction to analyze the form of such a derivation, on the one hand, and a new notion of variant to justify the termination of a recursive function that proceeds according to the size of a derivation, on the other hand. We show how this extension can be implemented conservatively, with almost no changes to Why3's verification condition generator.

3.5 K2 and VMT

Alberto Griggio (Bruno Kessler Foundation – Trento, IT)

Joint work of Alberto Griggio, Alessandro Cimatti, Stefano Tonetta, Martin Jonáš
Main reference Alberto Griggio, Martin Jonás: "Kratos2: An SMT-Based Model Checker for Imperative Programs", in Proc. of the Computer Aided Verification - 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part III, Lecture Notes in Computer Science, Vol. 13966, pp. 423–436, Springer, 2023.
URL https://doi.org//10.1007/978-3-031-37709-9_20
Main reference Alessandro Cimatti, Alberto Griggio, Stefano Tonetta: "The VMT-LIB Language and Tools", in Proc. of the 20th Internal Workshop on Satisfiability Modulo Theories co-located with the 11th International Joint Conference on Automated Reasoning (IJCAR 2022) part of the 8th Federated Logic Conference (FLoC 2022), Haifa, Israel, August 11-12, 2022, CEUR Workshop Proceedings, Vol. 3185, pp. 80–89, CEUR-WS.org, 2022.
URL https://ceur-ws.org/Vol-3185/extended9547.pdf

In this talk we present two languages for the representation of formal verification problems of infinite-state systems, called VMT and K2. VMT stands for Verification Modulo Theories, and is an extension of SMT-LIB for the representation of symbolic transition systems and properties over them, expressed either as invariants (must hold for all reachable states) or as more general properties in Linear Temporal Logic (LTL). VMT was designed for ease of use and to facilitate interoperability among model checking tools operating over symbolic transition systems, possibly using background theories for representing infinite-state systems. More information is available at `https://vmt-lib.fbk.eu/`. Also the K2 language is conceptually an extension of SMT-LIB; unlike VMT, however, K2 allows to represent imperative programs with multiple procedures. The language, which is the input format of the Kratos2 symbolic model checker, has been used successfully in different research and technology transfer projects with industrial partners as a simple but powerful intermediate language for verification. More information is available at `https://kratos.fbk.eu/`.

3.6 Deductive Validation: Witnesses from Automatic Verifiers for Deductive Verifiers

Matthias Heizmann (Universität Stuttgart, DE)

Joint work of Matthias Heizmann, Dominik Klumpp, Frank Schüssele, Marian Lingsch-Rosenfeld
Main reference Matthias Heizmann, Dominik Klumpp, Marian Lingsch Rosenfeld, Frank Schüssele: "Correctness Witnesses with Function Contracts", CoRR, Vol. abs/2501.12313, 2025.
URL https://doi.org//10.48550/ARXIV.2501.12313

We discuss the possibility to use deductive verification tools for validating correctness witnesses. Perhaps, surprisingly, the annotations provided by automatic verifiers are often not sufficient for deductive verifiers. In this talk we identify five challenges: inductiveness, function contracts, expressiveness of expressions, gotos, and undefined behavior.

3.7 AutoSV-Annotator: Integrating Deductive and Automatic Software Verification

Marieke Huisman (University of Twente – Enschede, NL)

Joint work of Lukas Armborst, Dirk Beyer, Marieke Huisman, Marian Lingsch-Rosenfeld

Software model checking and deductive software verification have complementary strengths and weaknesses: software model checkers are more straight-forward to use, as they analyze the program without user input; but they do not yet support complicated data structures and expressive specifications. In contrast, deductive verifiers can verify expressive specifications and complex data structures modularly, but they require the user to specify the program behavior in detail, which is a time-consuming process. Due to their differing nature, the two approaches usually remain separate. However, for industrial usage, one requires both: ease of use as well as expressiveness. Therefore, we present AutoSV-Annotator, a toolchain that integrates the two approaches for C programs. The toolchain allows a user to iteratively refine the deductive annotations in a C program, calling a model checker to supplement the annotations at each iteration, guided by the already existing annotations. We show that our tool is able to annotate and prove many tasks from the SV-Benchmarks set. Our results show that the two strategies can indeed benefit from each other.

3.8 Information Exchange with CoVeriTest and Predicate Maps

Marie-Christine Jakobs (LMU Munich, DE)

Joint work of Dirk Beyer, Marie-Christine Jakobs
Main reference Marie-Christine Jakobs: "Reusing Predicate Precision in Value Analysis", in Proc. of the Integrated Formal Methods - 17th International Conference, IFM 2022, Lugano, Switzerland, June 7-10, 2022, Proceedings, Lecture Notes in Computer Science, Vol. 13274, pp. 63–85, Springer, 2022.
URL https://doi.org//10.1007/978-3-031-07727-2_5

In this talk, I present two application areas for information exchange, test-case generation and cooperation in verificaton. I start to describe how I exchange information for test-case generation with CoVeriTest, a cooperative test-case generator based on the configurable program analysis framework CPAchecker. After a brief reminder on test-case generation with verifiers, I present the circular composition used by CPAchecker and give ideas on how information exchanged on test goals but also the explored state space. Thereafter, I continue discussing general application scenarios and concrete approaches for exchanging predicate maps, CPAchecker's format to specify the abstraction level of CPAchecker's predicate analysis, i.e., which predicates to use where. I give an overview of producer and consumers of predicate maps and provide an idea how to exchange information between predicate and value analysis. On the one hand, I sketch how to extract information from predicate maps for value analysis, in particular to derive the relevant variables to track. On the other hand, I indicate how to use information from the value analysis to generate predicate maps. I conclude the talk with the lessons learnt from the presented exchange approaches.

3.9 SMT-LIB for Exchange of Information Between Verifiers

Martin Jonáš (Masaryk University – Brno, CZ)

URL https://smt-lib.org/

SMT-LIB is a widely-used format for specification of first-order formulas over various logical theories, used by a majority of current state-of-the-art SMT solvers. In this talk, I provide a high-level overview of the format with focus on the exchange of information between software verifiers. The talk also introduces several concepts, such as annotation mechanism, that are used by various software-verification languages based on SMT.

3.10 Specification and Verification with Frama-C/MetAcsl

Nikolai Kosmatov (Thales Research & Technology – Palaiseau, FR)

Joint work of Adel Djoudi, Martin Hána, Nikolai Kosmatov
Main reference Adel Djoudi, Martin Hána, Nikolai Kosmatov: "Formal Verification of a JavaCard Virtual Machine with Frama-C", in Proc. of the Formal Methods - 24th International Symposium, FM 2021, Virtual Event, November 20-26, 2021, Proceedings, Lecture Notes in Computer Science, Vol. 13047, pp. 427–444, Springer, 2021.
URL https://doi.org//10.1007/978-3-030-90870-6_23

In the Frama-C ecosystem, tool collaboration is often based on ACSL annotations. Various kinds of more complex properties are translated into basic ACSL annotations that can then be verified using different tools. Examples of properties include relational properties, temporal logic properties, test objectives, etc. In this talk we show how global (high-level) properties are specified and verified using the MetAcsl plugin of Frama-C. Such properties include security properties (isolation, confidentiality, integrity) and are highly relevant for security certification. This approach was successfully used by Thales to formally verify its JavaCard Virtual Machine for the highest level of security certification (EAL7) of Common Criteria.

3.11 Towards Scalable and Distributed Software Verification

Thomas Lemberger (LMU Munich, DE)

Joint work of Dirk Beyer, Matthias Kettl, Thomas Lemberger
Main reference Dirk Beyer, Matthias Kettl, Thomas Lemberger: "Decomposing Software Verification using Distributed Summary Synthesis", Proc. ACM Softw. Eng., Vol. 1(FSE), pp. 1307–1329, 2024.
URL https://doi.org//10.1145/3660766

There are many approaches for automated software verification, but they are either imprecise, do not scale well to large systems, or do not sufficiently leverage parallelization. We propose an approach to decompose one large verification task into multiple smaller, connected verification tasks, based on blocks in the program control flow. For each block, summaries (block contracts) are computed – based on independent, distributed, continuous refinement by communication between the blocks. The approach iteratively synthesizes preconditions to assume at the block entry (computed from postconditions received from block predecessors, i.e., which program states reach this block) and violation conditions to check at the block exit (computed from violation conditions received from block successors, i.e., which program states lead to a specification violation). This separation of concerns leads to an architecture in which blocks can be analyzed in parallel, as separate verification problems. Whenever new information is available from other blocks, the verification can decide to restart with this new information. We realize our approach on the basis of configurable program analysis and implement it for the verification of C programs in the widely used verifier CPAchecker.

3.12 From Requirements to Heterogenous Verification: Where Is the Toolchain?

Rosemary Monahan (Maynooth University, IE)

Joint work of Marie Farrell, Matt Luckcuck, Rosemary Monahan, Conor Reynolds, Oisín Sheridan
Main reference Marie Farrell, Matt Luckcuck, Rosemary Monahan, Conor Reynolds, Oisín Sheridan: "Adventures in FRET and Specification", CoRR, Vol. abs/2503.24040, 2025.
URL https://doi.org//10.48550/ARXIV.2503.24040

Knowing which properties to specify and subsequently verify is one of the biggest bottlenecks in the use of Formal Methods [1]. Requirements are often written in natural-language, which can be ambiguous and is not generally amenable to formal verification. As a result, requirements elicitation and formalisation becomes even more important for understanding what to specify and verify.

Tools such as NASA's Formal Requirements Elicitation Tool (FRET) have been developed to bridge the gap between natural-language requirements and the logics normally used for formal specification [2]. FRET provides a structured natural-language, called FRETISH, from which temporal logic specifications and other verification conditions can be derived. FRET has been used in many use cases to support both elicitation and formalisation of requirements. The case studies that we report on are an aircraft engine software controller, a mechanical lung ventilator, a rover carrying out an inspection task and an algorithm for autonomously grasping spent rocket stages. We discuss how the FRETISH requirements were elicited and subsequently used for specifying properties to be verified in various formal methods (model-checking, theorem proving and runtime verification approaches) encouraging a framework for interoperability.

The overarching aim of each of these case studies was different. Specifically, in the aircraft engine controller we focused our efforts on eliciting and formalising requirements in conversation with an industrial partner (Collins Aerospace) [3]. In the mechanical lung ventilator we formalised requirements that were supplied in the ABZ case study documentation with a view to developing a formal model of the system in Event-B [4]. For the inspection rover [5], the integration of various verification artefacts into an assurance case were explored, with a hazard analysis used to define the requirements that were formalised in FRET and subsequently verified against models of the system using CoCoSim and Event-B. In the autonomous grasping case study [6], the authors modelled the code of the system using Dafny, and generated a suite of runtime monitors using ROSMonitoring that were able to identify requirement violations at run time.

Tools like FRET are invaluable when communicating with engineers and developers both in academia and industry. This was evidenced by the aircraft engine controller and autonomous grasping case studies. In both, FRET's diagrammatic and natural-language semantics helped case study providers to consider their system from different perspectives. This undoubtedly provided a bridge for these developers, providing traceability from natural-language to formalised requirements and verification conditions. This approach to demonstrating formal methods in a user-friendly way that leverages existing requirements engineering approaches encouraged our collaborators to consider using similar formal tools in the future.

Throughout these case studies, we used FRET not only as an elicitation/formalisation tool but also as a way to manage the requirements that we were defining. We used the support that FRET provides for defining parent-child relationships between requirements, which allowed us to group related requirements together. Currently this parent-child link in FRET is informal in the sense that the user can assign a parent to a requirement without any formal/automatic checks that the requirements are related in some way. In future FRET developments it may be useful to formalise this relationship to allow the user to gradually refine requirements and provide a way to verify the traceability that we have documented between them.

In these case studies, some of the integration was via existing tool-supported translations (FRET to CoCoSpec) but the others were manual translations (FRET to Event-B, Dafny and ROSMonitoring). In the future, we hope that we can provide a more logically-founded and systematic approach to integrating these formal methods. Mathematical frameworks like UTP [7] or institution theory [8] may have roles to play here. However, less formal approaches will also be beneficial including the definition of workflows to combine and use various methods alongside one another.

This work was partially supported by the Royal Academy of Engineering and EPSRC grant EP/Y001532/1, as well as Maynooth University's Hume Doctoral Award. We thank our co-authors and collaborators on each of the case studies: Stylianos Basagiannis, Georgios Giantamidis, Vassilios Tsachouridis, Hamza Bourbouh, Anastasia Mavridou, Irfan Sljivo, Guillaume Brat, Louise Dennis, Michael Fisher, Nikos Mavrakis, Angelo Ferrando, Yang Gao and Clare Dixon.

References

1 Kristin Yvonne Rozier. *Specification: The biggest bottleneck in formal methods and autonomy*. In Verified Software. Theories, Tools, and Experiments: 8th International Conference, VSTTE 2016, Toronto, ON, Canada, July 17–18, 2016, Revised Selected Papers 8, pages 8–26. Springer, 2016

2 Dimitra Giannakopoulou, Thomas Pressburger, Anastasia Mavridou, and Johann Schumann. *Automated formalization of structured natural language requirements*. Information and Software Technology, 137:106590, 2021.

3 Marie Farrell, Matt Luckcuck, Oisín Sheridan, and Rosemary Monahan. *FRETting About Requirements: Formalised Requirements for an Aircraft Engine Controller*. In Requirements Engineering: Foundation for Software Quality, pages 96–111. Springer, 2022.

4 Marie Farrell, Matt Luckcuck, Rosemary Monahan, Conor Reynolds, and Oisín Sheridan, *Fretting and formal modelling: A mechanical lung ventilator*. In International Conference on Rigorous State Based Methods, 2024.

5 Hamza Bourbouh, Marie Farrell, Anastasia Mavridou, Irfan Sljivo, Guillaume Brat, Louise A. Dennis, and Michael Fisher. *Integrating Formal Verification and Assurance: An Inspection Rover Case Study*. In NASA Formal Methods, pages 53–71. Springer, 2021.

6 Marie Farrell, Nikos Mavrakis, Angelo Ferrando, Clare Dixon, and Yang Gao. *Formal Modelling and Runtime Verification of Autonomous Grasping for Active Debris Removal*. Frontiers in Robotics and AI, 2022.

7 Charles Anthony Richard Hoare. *Unified theories of programming*. In Mathematical methods in program development, pages 313–367. Springer, 1997.

8 Joseph A. Goguen and Rod M. Burstall. *Institutions: Abstract model theory for specification and programming*. Journal of the ACM, 39(1):95–146, 1992.

3.13 Cooperation via Splitting

Cedric Richter (Carl von Ossietzky Universität Oldenburg, DE)

Joint work of Cedric Richter, Marek Chalupa, Marie-Christine Jakobs, Heike Wehrheim
Main reference Cedric Richter, Marek Chalupa, Marie-Christine Jakobs, Heike Wehrheim: "Cooperative Software Verification via Dynamic Program Splitting", in Proc. of the 47th IEEE/ACM International Conference on Software Engineering, ICSE 2025, Ottawa, ON, Canada, April 26 - May 6, 2025, pp. 2087–2099, IEEE, 2025.
URL https://doi.org//10.1109/ICSE55347.2025.00092

The goal of cooperative software verification is to share the complexity of the verification task between verifiers. While there exists many approaches to cooperation between software verifiers, they often require specialized interfaces for communication. This hinders the application of off-the-shelf verification tools in a cooperative setting which do not support the required interfaces for information exchange. One idea to circumvent the problem is by encoding the verification progress directly into the program. Still, a verifier that communicates its verification progress is required.

In this talk and in the accompanying paper, we propose dynamic program splitting as a way to record the verification progress of off-the-shelf verification tools. Dynamic program splitting starts by running a given verifier. If the verifier cannot solve the task within a given time, dynamic program splitting then splits the task into multiple parts. The verifier is then executed on each subtask. This process continues on the unverified parts of the verification task, continuously splitting subtasks that are not yet solved. In the end, all unsolved parts are merged together to form a residual verification task. This task can directly be processed by a second verifier. The key advantage of our approach is that off-the-shelf verifiers can be directly be used without requiring specialized interfaces, as the main communication is performed on the task itself. Our evaluation shows that being able to use off-the-shelf tools can be a key advantage in cooperative verification.We also report on further experimental results and discuss open problems.

3.14 MoXI: An Intermediate Language to Spur Reproducible and Comparable Model Checking Research

Kristin Yvonne Rozier (Iowa State University – Ames, US)

Joint work of Kristin Yvonne Rozier, Rohit Dureja, Ahmed Irfan, Chris Johannsen, Karthik Nukala, Natarajan Shankar, Cesare Tinelli, Moshe Y. Vardi
Main reference Kristin Yvonne Rozier, Rohit Dureja, Ahmed Irfan, Chris Johannsen, Karthik Nukala, Natarajan Shankar, Cesare Tinelli, Moshe Y. Vardi: "MoXI: An Intermediate Language for Symbolic Model Checking", in Proc. of the Model Checking Software - 30th International Symposium, SPIN 2024, Luxembourg City, Luxembourg, April 8-9, 2024, Proceedings, Lecture Notes in Computer Science, Vol. 14624, pp. 26–46, Springer, 2024.
URL https://doi.org//10.1007/978-3-031-66149-5_2
Main reference Chris Johannsen, Karthik Nukala, Rohit Dureja, Ahmed Irfan, Natarajan Shankar, Cesare Tinelli, Moshe Y. Vardi, Kristin Yvonne Rozier: "The MoXI Model Exchange Tool Suite", in Proc. of the Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 14681, pp. 203–218, Springer, 2024.
URL https://doi.org//10.1007/978-3-031-65627-9_10

As symbolic model checking algorithms have grown more powerful and popular, their implementations have grown increasingly closed-source and specialized. This move away from standardization has stunted symbolic model checking research: there was not a way to build upon or adapt closed-source tools to compare with new algorithms; there was not a way to compare state-of-the-art back-end model checking algorithms because their implementations reasoned over models in different input languages; and replication of previous research results proved challenging due to hidden implementation details. After extensive input from the model checking research community, we have the intermediate language MoXI, the Model eXchange Interlingua, built to stimulate symbolic model checking research. Publicly available MoXI translators enable comparability with previous research results. Adding a new back-end model checking algorithm and comparing it to previously published algorithms is now as easy as creating a MoXI translator to/from the new algorithm. One can now compare any number of back-end implementations over inputs in any number of different model or specification languages by compiling through MoXI. Accordingly, the public collection of translators through MoXI is quickly growing. We will highlight the state of the art in reproducible and comparable model checking research and point to community resources to continue the groundswell.

The GitHub organization provides full artifacts: https://github.com/ModelChecker.

3.15 ACSL: Behavioral Interface Specification Language for C Code

Julien Signoles (Université Paris-Saclay, CEA, List, FR)

Main reference Louis Gauthier, Virgile Prevosto, Julien Signoles: "A Semantics of Structures, Unions, and Underspecified Terms for Formal Specification", in Proc. of the 2024 IEEE/ACM 12th International Conference on Formal Methods in Software Engineering (FormaliSE), Lisbon, Portugal, April 14-15, 2024, pp. 100–110, ACM, 2024.
URL https://doi.org//10.1145/3644033.3644380

ACSL is a Behavioral Interface Specification Language for C Code. It is independent from any tool, even if Frama-C, a framework for analyses of C code, provides a reference implementation. ACSL is meant to express precisely and unambiguously the expected behavior of a piece of C code. It is close to other contract-based languages, e.g., JML for Java, even if there are differences that mainly comes from the underlying programming language. Typically, ACSL must support specification of (low-level) memory properties, which are critical for C code. This short talk introduces the main constructs of ACSL, illustrated by a few small examples.

References

1 P. Baudin, P. Cuoq, J.-C. Filliâtre, C. Marché, B. Monate, Y. Moy, V. Prevosto. ANSI/ISO C Specification Language Reference Manual. https://frama-c.com/download/acsl.pdf

2 A. Blanchard. Introduction to C program proof with Frama-C and its WP plugin. https://allan-blanchard.fr/publis/frama-c-wp-tutorial-en.pdf

3 A. Blanchard, C. Marché, V. Prevosto. Formally Expressing what a Program Should Do: The ACSL Language. In Guide to Software Verification with Frama-C: Core Components, Usages and Applications, Springer Cham, 2024

3.16 Correctness Verification Witnesses 2.0

Jan Strejček (Masaryk University – Brno, CZ)

Joint work of Paulína Ayaziová, Dirk Beyer, Marian Lingsch Rosenfeld, Martin Spiessl, Jan Strejček
Main reference Paulína Ayaziová, Dirk Beyer, Marian Lingsch Rosenfeld, Martin Spiessl, Jan Strejček: "Software Verification Witnesses 2.0", in Proc. of the Model Checking Software - 30th International Symposium, SPIN 2024, Luxembourg City, Luxembourg, April 8-9, 2024, Proceedings, Lecture Notes in Computer Science, Vol. 14624, pp. 184–203, Springer, 2024.
URL https://doi.org//10.1007/978-3-031-66149-5_11

Verification witnesses are now widely accepted objects used not only to confirm or refute verification results, but also for general exchange of information among various tools for program verification. The original format for witnesses is based on GraphML, and it has some known issues including a semantics based on control-flow automata, limited tool support of some format features, and a large size of witness files. The version 2.0 of the witness format is based on YAML and overcomes the above-mentioned issues. The presentation focuses on correctness witnesses in format version 2.0 and it also mentions some extensions planned for version 2.1.

3.17 Paradigm-Spanning Collaboration: The Karlsruhe Java Verification Suite

Mattias Ulbrich (KIT – Karlsruher Institut für Technologie, DE)

Joint work of Jonas Klamroth, Florian Lanzinger, Wolfram Pfeifer, Mattias Ulbrich
Main reference Jonas Klamroth, Florian Lanzinger, Wolfram Pfeifer, Mattias Ulbrich: "The Karlsruhe Java Verification Suite", in Proc. of the The Logic of Software. A Tasting Menu of Formal Methods - Essays Dedicated to Reiner Hähnle on the Occasion of His 60th Birthday, Lecture Notes in Computer Science, Vol. 13360, pp. 290–312, Springer, 2022.
URL https://doi.org//10.1007/978-3-031-08166-8_14

Deductive verification of sophisticated properties of sophisticated data structures is a difficult task. Proof endeavours in such contexts still have many proof obligation which do not require the expressiveness of a deductive verifier.

In the talk I presented the Karlsruhe Java Verification Suite, a collection of verification tool around the deductive verification engine KeY. The bounded model checker JJBMC extends the checker JBMC by JML specifications and the Property Checker uses the Java Checker Framework to capture object properties and provides a lightweight method to annotate formal guarantees throughout an application.

The tools cooperate via JML and Java assertions. We show that if every assertion in a program is handled by at least one sound verification tool all other tools can change assertions into helpful assumptions.

3.18 Reducers to Aid Information Exchange

Heike Wehrheim (Universität Oldenburg, DE)

Joint work of Dirk Beyer, Marie-Christine Jakobs, Thomas Lemberger, Heike Wehrheim
Main reference Dirk Beyer, Marie-Christine Jakobs, Thomas Lemberger, Heike Wehrheim: "Reducer-based construction of conditional verifiers", in Proc. of the 40th International Conference on Software Engineering, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018, pp. 1182–1193, ACM, 2018.
URL https://doi.org//10.1145/3180155.3180259

Despite recent advances, software verification remains challenging. To solve hard verification tasks, we need to leverage not just one but several different verifiers employing different technologies. To this end, we need to exchange information between verifiers. Conditional model checking was proposed as a solution to exactly this problem: The idea is to let the first verifier output a condition which describes the state space that it successfully verified and to instruct the second verifier to verify the yet unverified state space using this condition. However, most verifiers do not understand conditions as input.

In this talk and the accompanying paper, we propose the usage of an off-the-shelf construction of a conditional verifier from a given traditional verifier and a reducer. The reducer takes as input the program to be verified and the condition, and outputs a residual program whose paths cover the unverified state space described by the condition. This program can then be given to the second verifier for completing the verification. We report on results of some experiments and discuss open questions.

3.19 Bridging Hardware and Software Formal Verification

Zsófia Ádám (Budapest University of Technology & Economics, HU), Dirk Beyer (LMU Munich, DE)

Joint work of Zsófia Ádám, Dirk Beyer, Po-Chun Chien, Nian-Ze Lee, Nils Sirrenberg
Main reference Zsófia Ádám, Dirk Beyer, Po-Chun Chien, Nian-Ze Lee, Nils Sirrenberg: "Btor2-Cert: A Certifying Hardware-Verification Framework Using Software Analyzers", in Proc. of the Tools and Algorithms for the Construction and Analysis of Systems - 30th International Conference, TACAS 2024, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2024, Luxembourg City, Luxembourg, April 6-11, 2024, Proceedings, Part III, Lecture Notes in Computer Science, Vol. 14572, pp. 129–149, Springer, 2024.
URL https://doi.org//10.1007/978-3-031-57256-2_7

Computing systems demand meticulous verification to ensure their correct functionality. Formal methods have been successfully applied in real-world scenarios and delivered correctness guarantees with mathematical rigor. The challenges to formal methods have escalated due to increasing interactions among diverse components, e.g., software programs, hardware circuits, and cyber-physical devices, therefore, a cross-disciplinary approach is necessary among separate research communities. Although the research communities for formal methods share similar concepts and techniques, their alignment with distinct computational models creates gaps between the communities. In this talk, we introduce our attempts to bridge the gap between software and hardware analysis, including transferring verification approaches and translating verification tasks. By systematically cross-applying their techniques and extracting valuable insights into analyzing heterogeneous computing systems, we aim to unify verification approaches for comprehensive system-level verification.

4 Working Groups

4.1 Breakout Group: The Goals of the Seminar from the Perspective of Software Model Checking Community

Jan Strejček (Masaryk University – Brno, CZ)

Joint work of Zsófia Ádám, Paulína Ayaziová, Marek Chalupa, Daniel Dietsch, Matthias Heizmann, Marie-Christine Jakobs, Thomas Lemberger, Cedric Richter, Jan Strejček, Heike Wehrheim

The breakout group consisted mainly of developers of automatic software verification tools. The group formulated the following answers to questions given by organizers.

What do we want to achieve with the exchange formats? The group wants to support the cooperation between (static and dynamic) verifiers and/or program analyses (that can provide, e.g., points-to information). It should also facilitate incremental verification, validation of verification results, human/AI interaction with verification tools, and support human reasoning about the program and found property violations.

Which objects do we want to exchange? The group came up with the following objects: invariants (including object invariants and transition invariants), useful ghost code, abstract reachability graphs, precision sufficient to prove the task, function contracts/summaries, error conditions, ranking functions, abstract contracts (independent of the implementation), assumptions for verifiers, verification sub-tasks (that were not proven yet), memory-specific information (e.g., shape graphs), information about synchronization between threads (e.g., execution graphs or lock order), sets of specific program paths (e.g., abstract counterexamples), and meta-information about the provenance of the presented objects.

What are suitable formats to use for the exchange? Besides the formats that are already widely used by the community (C programs, witness formats), the group suggested using (E-)ACSL and introducing specific predicates to describe some of the objects identified above. The group also discussed whether the information retrieval should be static or query-based and whether the information should be exchanged within the program (internally) or aside (externally). Here the group did not agree on answers as each option has its pros and cons.

4.2 Breakout Group: Exchange Between Deductive Verification

Marieke Huisman (University of Twente – Enschede, NL)

Joint work of Wolfgang Ahrendt, David Cok, Gidon Ernst, Jean-Christophe Filliâtre, Marieke Huisman, Nikolai Kosmatov, Robert Mensing, Rosemary Monahan, Simmo Saan, Julien Signoles, Alexander Stekelenburg, Mattias Ulbrich

This breakout group had participants that are working on deductive verifiers. The discussion concentrated mainly on exchange between deductive verification tools. We started by discussing how interoperability involving deductive verifiers could be good. Concrete examples where heterogeneous verification, where different tools target different properties, validation and cross-checking of verification results, and combinations with automatic tools that could be used first to resolve the basic properties, or to infer specifications.

We then discussed what would be the appropriate level of exchange. One obvious option is to exchange concrete examples, i.e., programs with specifications. Exchanging intermediate verification results would be interesting, but also much more challenging. One important candidate is the exchange of axiomatisations of data structures and logical theories, that are used in specifications, possibly based on format inspired by SMT-LIB.

Finally, we concluded the discussion by identifying areas where already some exchange is happening between deductive verifiers, such as ContractLib, and the language constructs for which exchange could take place, such as well-encapsulated objects, lemmas etc.

In a second session, the breakout group continued by identifying a list of concrete short-term actions that could be taken to experiment with information exchange: sharing axiomatisations, work on a common example together, and organise a hackathon/specathon, to make progress on the examples, share examples of verified programs, and add tools to FM-Tools.

In the long term, we believe that these actions should lead to support for specificaton exchange, annotation exchange and proof exchange. We also identified a concrete list of semantical issues that we would have to agree upon to enable this.

4.3 Breakout Group: Intermediate Language for Software Verification

Dirk Beyer (LMU Munich, DE)

Joint work of Dirk Beyer, Gidon Ernst, Alberto Griggio, Martin Jonáš, Viktor Malík, Robert Mensing, Kristine Yvonne Rozier, Jan Strejček, Vesal Vojdani

The state of the art in software verification is to use programming languages like C and Java, possibly annotated using ACSL, JML, and other annotation languages, to describe the verification tasks. The two breakout sessions discussed the need to exchange verification tasks and the motivation to create a new, community-based, intermediate language (IL) that is easy to support and process by tools. The requirements for the language are as follows: The IL should

(a) have a formal semantics, such that comparative evaluations of verification algorithms does not depend on interpretations, and the same should hold for witness validation,

(b) support interoperability, such that frontends and backends are exchangable,

(c) support verification and witness validation by transformation,

(d) be independent from the higher-level input language (such as C and Java) and independent from verification technology used in the backend,
(e) retain more structure than MoXI, VMT, and CHC (control flow, procedures, sequential composition, ...) to express tasks before encoding to verification conditions by a particular approach,
(f) be based on well-understood programming concepts and consolidate existing languages in that spectrum (e.g., Boogie, Why3, Viper),
(g) support witness validation that is compilable to a simple first-order check (i.e., SMT) without further inference, for all properties, and
(h) appeal to use-cases in both software model checking and deductive verification.

The objects to be exchanged by the language shall be (1) programs, (2) properties, and (3) witnesses (counterexamples and invariants). The group suggests that the new IL should be based on a text format (not binary) and use a well-known base syntax (e.g., Lisp-based). As a concrete outcome, it was agreed that the language would be defined before autumn 2025 and that a track for the language shall be established by the SV-COMP competition organizers.

4.4 Breakout Group: Decomposition of the Verification Task

Heike Wehrheim (Universität Oldenburg, DE)

Joint work of Ádám, Zsófia; Wolfgang Ahrendt, Paulína Ayaziová, Marek Chalupa, Daniel Dietsch, Matthias Heizmann, Marie-Christine Jakobs, Thomas Lemberger, Cedric Richter, Simmo Saan, Vesal Vojdani, Heike Wehrheim

A verification task is a program and a (correctness) specification. The purpose of decomposition is to divide a verification task into several subtasks which can be given to (different) software verifiers for solving them. The verifier's results then at the end need to be combined. Thus, we do not just need decomposition, but also merging of results (for which we however did not have time in the discussion).

We basically identified two sorts of decomposition:

Based on program structure Decomposition based on program structure might split programs on branches or along paths. We decided to not discuss this longer as there are already a number of proposals around.

Based on partial results The second sort of decomposition we considered is the one based on a partial results, i.e., one verifier has already inspected the verification task and has obtained some partial information about it. This partial information might have come along because of a non-finished (full) proof attempt or as a result of a specific query (e.g., "give me all points-to information").

As sorts of partial results we identified (a) conditions (as in Conditional Model Checking), (b) annotated Control-Flow-Graphs/Automata (this is what e.g. the verifier Goblint could produce) or (c) specific assumptions under which the verification has succeeded (e.g., knowledge about no non-null pointers or predicates about input variables). Potential formats of such assumptions have then been further discussed in another breakout group.

4.5 Breakout Group: Exchange of Partial Verification Results

Zsófia Ádám (Budapest University of Technology & Economics, HU) and Thomas Lemberger (LMU Munich, DE)

Joint work of Zsófia Ádám, Wolfgang Ahrendt, Paulína Ayaziová, Marek Chalupa, Daniel Dietsch, Thomas Lemberger, Heike Wehrheim

Automated and deductive verifiers perform well for different challenges. This motivates different methods of cooperation, which often require the sharing of (partial) verification results. However, no widely adopted exchange format between automated and deductive verifiers exists. The working group on exchange of partial verification results focused on three aspects of a successful exchange format:

(1) The type of information that may be exchanged. Most notably:
- Local or global assumptions that the verifier assumed;
- invariants of various types: global invariants, location-bound invariants, loop invariants, object invariants, and resource invariants; and
- properties that were established through the analysis, such as safety of assertions or nullability.

(2) Existing related formats that serve as examples of information exchange in verification. For example the Software Witness 2.0 format and (E-)ACSL. In this context the working group also discussed various pitfalls in exchange formats: undefined behavior in the expression language and how to precisely define elements and corresponding locations tool-independently.

(3) The terminology to use and biases in data representation that may be introduced through certain terminology. For example, the terms *assumed* (compared to *assert*) and *asserted* (compared to *assume*) express that the information is not instructions for a potential consumer, but that it is a description of the performed work and established truths; independent of the consumer's use case.

Going forward, we propose a wider discussion on the above concepts and ideas to establish a consumer-independent exchange format for partial verification results.

4.6 Breakout Group: Exchange Between Deductive Tools (DV) and Fully Automatic Software Verifiers (SV)

Marieke Huisman (University of Twente – Enschede, NL)

Joint work of David Cok, Gidon Ernst, Jean-Christophe Filliâtre, Alberto Griggio, Matthias Heizmann, Marieke Huisman, Nikolai Kosmatov, Robert Mensing, Rosemary Monahan, Cedric Richter, Julien Signoles, Alexander Stekelenburg, Mattias Ulbrich

In this breakout group, we discussed how fully automatic software verifiers could be combined with deductive verifiers. We first discussed several ongoing efforts in this direction. There are several ways in which the combination can work: deductive verifiers can prove some parts that are hard for automated software verifiers, software verifiers can prove dedicated properties that are useful for the deductive verifiers. We can also use the combination to speed up the overall verification process.

One particular challenge is that deductive verifiers often uses specifications with quantifiers. Not all software verifiers can reason about those, but there are several tricks to work around this, e.g., by encoding the quantified variable as a non-deterministic variable, but this encoding hinders the exchange.

We agreed on several follow-up actions, in particular to look at some concrete examples and look what could be achieved. We also intend to collect the existing approaches for integration in a joint document, so everybody could study them. Eventually, we plan to organise some online events to discuss the experiences with some of these concrete examples.

4.7 Breakout Group: Information Exchange Between Automatic SW Verifiers

Jan Strejček (Masaryk University – Brno, CZ)

Joint work of Dirk Beyer, Marie-Christine Jakobs, Martin Jonáš, Viktor Malík, Kristin Yvonne Rozier, Simmo Saan, Jan Strejček, Vesal Vojdani

The group went through (a part of) the list of objects for exchange between verification tools identified by Wednesday's breakout group of the software model checking community (see Section 4.1). Using that list, we suggested further extensions of the witness format 2.0, in particular more general ghost code, `modifies` clause in function contracts, and support of a fragment of ACSL.

In the second part of the session, the group focused on designing a format for exchanging precision (i.e., used predicates and relevant variables) that was used during the verification run. We agreed on the objects that should represent the precision: side-effect-free and terminating C expressions together with their scope (global, local, or a particular function). We also decided to use YAML and adopt some parts of the witness format 2.0. Finally, we have formulated some action items aiming to establish the precision exchange format.

Participants

- Zsófia Ádám
Budapest University of Technology & Economics, HU
- Wolfgang Ahrendt
Chalmers University of Technology – Göteborg, SE
- Paulína Ayaziová
Masaryk University – Brno, CZ
- Dirk Beyer
LMU München, DE
- Marek Chalupa
IST Austria – Klosterneuburg, AT
- David Cok
Safer Software Consulting – Rochester, US
- Daniel Dietsch
Universität Freiburg, DE & Qt Group – Espoo, FI
- Gidon Ernst
LMU München, DE
- Jean-Christophe Filliâtre
CNRS – Gif-sur-Yvette, FR
- Alberto Griggio
Bruno Kessler Foundation – Trento, IT
- Matthias Heizmann
Universität Stuttgart, DE
- Marieke Huisman
University of Twente – Enschede, NL
- Marie-Christine Jakobs
LMU München, DE
- Martin Jonáš
Masaryk University – Brno, CZ
- Nikolai Kosmatov
Thales Research & Technology – Palaiseau, FR
- Thomas Lemberger
LMU München, DE
- Viktor Malík
Brno University of Technology, CZ
- Robert Mensing
University of Twente – Enschede, NL
- Rosemary Monahan
Maynooth University, IE
- Cedric Richter
Carl von Ossietzky Universität Oldenburg, DE
- Kristin Yvonne Rozier
Iowa State University – Ames, US
- Simmo Saan
University of Tartu, EE
- Julien Signoles
Université Paris-Saclay, CEA, List, FR
- Alexander Stekelenburg
University of Twente – Enschede, NL
- Jan Strejček
Masaryk University – Brno, CZ
- Mattias Ulbrich
KIT – Karlsruher Institut für Technologie, DE
- Vesal Vojdani
University of Tartu, EE
- Heike Wehrheim
Universität Oldenburg, DE

Report from Dagstuhl Seminar 25181

Learned Predictions for Data Structures and Running Time

Inge Li Gørtz[*1], **Benjamin J. Moseley**[*2], **Shikha Singh**[*3], and **Sergei Vassilvitskii**[*4]

1 **Technical University of Denmark - Lyngby, DK.** inge@dtu.dk
2 **Carnegie Mellon University - Pittsburgh, US.** moseleyb@andrew.cmu.edu
3 **Williams College - Williamstown, US.** ss32@williams.edu
4 **Google - New York, US.** sergeiv@gmail.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 25181 "Learned Predictions for Data Structures and Running Time". The focus of the seminar was applying the new algorithms-with-predictions framework to improve worst-case running time of algorithms and data structures.

This seminar brought together researchers from the data structures, combinatorial optimization and learned predictions communities to address the challenges of adopting learned machine-learned predictions for improving running time guarantees.

Seminar April 27 – May 2, 2025 – https://www.dagstuhl.de/25181
2012 ACM Subject Classification Theory of computation
Keywords and phrases algorithms with predictions, approximation algorithms, beyond-worst-case analysis, data structures, learning-augmented algorithms
Digital Object Identifier 10.4230/DagRep.15.4.112

1 Executive Summary

Inge Li Gørtz
Benjamin J. Moseley
Shikha Singh
Sergei Vassilvitskii

Traditionally, the performance of algorithms and data structures is measured in terms of the worst-case analysis of their *cost*, e.g. running time, approximation ratio or memory usage. Algorithms designed in the worst-case model are optimized to perform well against all instances, even carefully contrived adversarial instances. While the worst-case model leads to extremely strong guarantees, it can be a pessimistic predictor of the algorithm's performance on instances typically seen in practice. In particular, it is frequently observed that the theoretically best worst-case algorithm may not necessarily be the fastest algorithm in experimental evaluations. An emerging line of work, called **algorithms with predictions** or learning-augmented algorithms, uses machine learning to design improved algorithms that circumvent worst-case lower bounds. The majority of prior work in this area has focused on improving the quality of the solution (that is, the competitive ratio) of optimization algorithms. How to leverage learned predictions to speed up the *running time* of optimization algorithms and data structures has received less attention.

[*] Editor / Organizer

Seminar Overview

The goal of this Dagstuhl Seminar was to develop the theoretical foundations of using predictions for faster data structures and optimization. It brought together researchers from different but complementary areas of data structures, combinatorial optimization and learning theory.

The initial days of the seminar consisted of a small number of survey style talks (45 mins) focused on the key areas of learning-augmented and worst-case data structures, machine learning for NP hard problems as well as learning-augmented streaming algorithms. The rest of the schedule included shorter technical presentations, unstructured collaboration time, open-problem working groups as well as group discussions. Overall, the seminar was successful in bridging the gap between the different communities as well as creating new technical collaborations among the participants.

Technical Themes

Overall, the seminar converged on the following key themes.

Beyond-Worst-Case Data Structures. A major theme was how predictions can be used to improve the running time of fundamental data structures such as sorted arrays, priority queues and search trees. The main techniques that were discussed in the talks were: how prediction decomposition can be used for designing learned data structures, how history independence has been surprisingly successful at improving worst-case data structure performance, and how techniques such as compression, sketching and adapting to workload patterns can be leveraged for faster practical performance.

Warm-starting offline optimization. Traditional optimization algorithms assume that problems are solved from scratch each time. However, algorithms are often run on different correlated data sets over time. These prior runs can encode information that can be used to improve run times in the future. An intuitive approach to speed up an algorithm is to start with some previously computed solution, referred to as a *warm start*. Warm start is commonly used by heuristics to improve empirical performance. This seminar focused on how several recent results that provide a theoretical foundation of warm-start heuristics in the algorithms-with-predictions model.

Dynamic and streaming algorithms with predictions. Predictions can be specially useful in dynamic and streaming algorithms to help with the uncertainty with future inputs. The seminar focused on how predicting future updates can be used to efficiently "lift" fully offline algorithms to the online setting. This technique proved to be specially effective in the area of dynamic graph problems where the existing worst-case algorithms have slow update times (polynomial in input size). Similarly in the streaming setting, predictions about future updates enabled efficient sketching algorithms. Finally, the seminar also focused on how dynamic algebraic algorithms have been leveraged predictions to bypass worst-case lower bounds.

Learned combinatorial optimization. Finally, the seminar focused on how correlations in the input distributions can be exploited to design more efficient data-driven algorithms for optimization problems. These techniques have been applied to a variety of domains such as configuring integer programming algorithms, using heat maps and neural net techniques for improved algorithms for NP hard problems such as TSP, and using predicted distributions for portfolio optimization.

Takeaways and Conclusion

We believe the Dagstuhl Seminar was successful in meeting our as well as the participants expectations. The post-seminar survey was highly positive. Attendees rated the scientific quality of the seminar very highly (median 10/11) and said that it inspired new research ideas, interdisciplinary insights, and potential collaborations, particularly due to its successful integration of communities working on predictions and data structures.

In the future, we would like to include more PhD students as well as start working group activities earlier in the week. Overall, we were very happy with the seminar. We would like to thank the entire Schloss Dagstuhl team for making the seminar possible and ensuring that everything from initial planning, to logistics, meals, etc. were impeccably smooth.

2 Table of Contents

3 Overview of Talks

3.1 Approximation Algorithms with Predictions

Antonios Antoniadis (University of Twente, NL)

We initiate a systematic study of utilizing predictions to improve over approximation guarantees of classic algorithms, without increasing the running time. We propose a systematic method for a wide class of optimization problems that ask to select a feasible subset of input items of minimal (or maximal) total weight. This gives simple (near-)linear time algorithms for, e.g., Vertex Cover, Steiner Tree, Min-Weight Perfect Matching, Knapsack, and Clique. Our algorithms produce optimal solutions when provided with perfect predictions and their approximation ratios smoothly degrade with increasing prediction error. With small enough prediction error we achieve approximation guarantees that are beyond reach without predictions in the given time bounds, as exemplified by the NP-hardness and APX-hardness of many of the above problems. Although we show our approach to be optimal for this class of problems as a whole, there is a potential for exploiting specific structural properties of individual problems to obtain improved bounds; we demonstrate this on the Steiner Tree problem. We conclude with an empirical evaluation of our approach.

3.2 Sorting and Priority Queues with Predictions

Christian Coester (University of Oxford, GB)

Joint work of Xingjian Bai, Ziyad Benomar, Christian Coester

The talk discusses two of the most fundamental problems in algorithms and data structures through the lens of algorithms-with-predictions: sorting and priority queues. We consider different types of predictions and show how these can be used to improve running time and comparison complexity of algorithms. In one setting, the algorithm is given predictions about the rank of items, and in another setting, the algorithm has access to quick-and-dirty comparisons to complement much slower exact comparisons. We obtain simple algorithms with optimal guarantees and favorable experimental performance in sorting tasks as well as when applied to Dijkstra's shortest path algorithm.

3.3 Ski Rental With Distributional Predictions

Michael Dinitz (Johns Hopkins University - Baltimore, US)

Ski rental was one of the first problems to be considered in the algorithms with predictions literature. We revisit this problem from the perspective of distributional predictions. We give an algorithm with optimal additive loss as a function of the earth mover's distance between the predicted distribution and the true distribution, and show both analytically and experimentally that in many natural settings distributional predictions allow for significantly better performance than traditional point predictions.

3.4 Scenario-Based Robust Optimization of Tree Structures

Christoph Dürr (Sorbonne University - Paris, FR)

Joint work of Christoph Dürr, Spyros Angelopoulos, Alex Elenter, Georgii Melidi
Main reference Spyros Angelopoulos, Christoph Dürr, Alex Elenter, Georgii Melidi: "Scenario-Based Robust Optimization of Tree Structures", in Proc. of the AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA, pp. 26895–26903, AAAI Press, 2025.
URL https://doi.org/10.1609/AAAI.V39I25.34894

We initiate the study of tree structures in the context of scenario-based robust optimization. Specifically, we study Binary Search Trees (BSTs) and Huffman coding, two fundamental tech- niques for efficiently managing and encoding data based on a known set of frequencies of keys. Given k different scenarios, each defined by a distinct frequency distribution over the keys, our objective is to compute a single tree of best-possible performance, relative to any scenario. We consider, as performance metrics, the competitive ratio, which compares multiplicatively the cost of the solution to the tree of least cost among all scenarios, as well as the regret, which induces a similar, but additive comparison. For BSTs, we show that the problem is NP-hard across both metrics. We also show how to obtain a tree of competitive ratio $\lceil \log_2(k+1) \rceil$, and we prove that this ratio is optimal. For Huffman Trees, we show that the problem is, likewise, NP-hard across both metrics; we also give an algorithm of regret $\lceil \log_2 k \rceil$, which we show is near-optimal, by proving a lower bound of $\lfloor \log_2 k \rfloor$. Last, we give a polynomial-time algorithm for computing Pareto-optimal BSTs with respect to their regret, assuming scenarios defined by uniform distributions over the keys. This setting captures, in particular, the first study of fairness in the context of data structures. We provide an experimental evaluation of all algorithms. To this end, we also provide mixed integer linear program formulation for computing optimal trees.

3.5 From compression to learning, one more step in data structure design

Paolo Ferragina (Sant'Anna School of Advanced Studies - Pisa, IT)

Key-value stores and search engines are posing a continuously growing need to efficiently store, retrieve and analyze massive sets of keys under the many and different requirements posed by users, devices, and applications. Such a new level of complexity could not be properly handled by known data structures, so that academic and industrial researchers started recently to devise new approaches that integrate classic data structures with various kinds of advanced techniques drawn from data compression, computational geometry and machine learning, hence originating what are currently called "Learned Data Structures". In this talk, I'll survey the evolution of these kinds of data structures, discuss their theoretical and experimental performance, highlight their limits, and point out new challenges worth of future research.

3.6 Nearly Optimal List Labeling

Hanna Komlós (NYU - New York, US)

Joint work of Michael A. Bender, Alex Conway, Martín Farach-Colton, Hanna Komlós, Michal Koucký, William Kuszmaul, Michael E. Saks
Main reference Michael A. Bender, Alex Conway, Martín Farach-Colton, Hanna Komlós, Michal Koucký, William Kuszmaul, Michael E. Saks: "Nearly Optimal List Labeling", in Proc. of the 65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024, pp. 2253–2274, IEEE, 2024.
URL https://doi.org/10.1109/FOCS61266.2024.00132

The list-labeling problem is a fundamental data structural primitive which captures the basic task of storing a dynamically changing set of up to N elements in sorted order in an array of size $M = \Theta(N)$. The goal is to support online insertions and deletions while moving elements around in the array as little as possible. Despite over four decades of study, there has until recently been a longstanding gap between the upper and lower bounds for list labeling ($O(\log^2(N))$ and $\Omega(\log(N))$ element moves per operation, respectively). This gap was narrowed in 2022 by Bender et al. with a randomized history-independent algorithm with $O(\log^{1.5}(N))$ cost, and they showed a matching lower bound for history-independent list-labeling algorithms. In recent work, we nearly close the gap by providing an $\tilde{O}(\log(N))$ algorithm which leverages the workload's history of operations. We do this by building a randomized predictor of future operations that performs well on all inputs in expectation. Combining this with a prediction-augmented algorithm, we achieve an improved worst-case upper bound.

3.7 Faster Data Structures via History Independence

William Kuszmaul (Carnegie Mellon University - Pittsburgh, US)

This talk will survey a recent trend through which history independence – a widely-studied privacy property in the data-structures literature – has been used as an algorithmic tool, not just for privacy, but for building faster and better data structures.

3.8 Data Driven Solution Portfolios

Silvio Lattanzi (Google - Barcelona, ES)

Joint work of Marina Drygala, Silvio Lattanzi, Andreas Maggiori, Miltiadis Stouras, Ola Svensson, Sergei Vassilvitskii
Main reference Marina Drygala, Silvio Lattanzi, Andreas Maggiori, Miltiadis Stouras, Ola Svensson, Sergei Vassilvitskii: "Data-Driven Solution Portfolios", in Proc. of the 16th Innovations in Theoretical Computer Science Conference, ITCS 2025, January 7-10, 2025, Columbia University, New York, NY, USA, LIPIcs, Vol. 325, pp. 46:1–46:15, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.
URL https://doi.org/10.4230/LIPICS.ITCS.2025.46

In this paper, we consider a new problem of portfolio optimization using stochastic information. In a setting where there is some uncertainty, we ask how to best select k potential solutions, with the goal of optimizing the value of the best solution. More formally, given a combinatorial problem Π, a set of value functions V over the solutions of Π, and a distribution D over V, our goal is to select k solutions of Π that maximize or minimize the expected value of the *best* of those solutions. For a simple example, consider the classic knapsack problem: given a universe of elements each with unit weight and a positive value, the task is to select r elements maximizing the total value. Now suppose that each element's weight comes from a (known) distribution. How should we select k different solutions so that one of them is likely to yield a high value?

In this work, we tackle this basic problem, and generalize it to the setting where the underlying set system forms a matroid. On the technical side, it is clear that the candidate solutions we select must be diverse and anti-correlated; however, it is not clear how to do so efficiently. Our main result is a polynomial-time algorithm that constructs a portfolio within a constant factor of the optimal.

3.9 Learning Link Deletions

Quanquan C. Liu (Yale University - New Haven, US)

The main bottleneck in designing efficient dynamic algorithms is the unknown nature of the update sequence. In particular, there are some problems, like triconnectivity, planar digraph all pairs shortest paths, k-edge connectivity, and others, where the separation in runtime between the best offline or partially dynamic solutions and the best fully dynamic solutions is polynomial, sometimes even exponential. In this talk, we introduce the predicted-updates dynamic model, one of the first beyond-worst-case models for dynamic algorithms, which generalizes a large set of well-studied dynamic models including the offline dynamic, incremental, and decremental models to the fully dynamic setting when given predictions about the update times of the elements. In the most basic form of our model, we receive a set of predicted update times for all of the updates that occur over the event horizon. We give a novel framework that "lifts" offline divide-and-conquer algorithms into the fully dynamic setting with little overhead. Using this, we are able to interpolate between the offline and fully dynamic settings; when the ℓ_1 error of the prediction is linear in the number of updates, we achieve the offline runtime of the algorithm (up to poly log n factors). The second part of the talk will feature experimental results on how feasible it is to obtain these predictions in real-world graphs. We discuss new results that learn deletion times in dynamic networks using a very simple linear regression framework based on a set of features we find for our testing datasets.

3.10 Learned Data Structures via Decomposition

Samuel McCauley (Williams College - Williamstown, US)

This talk is about recent results for how predictions can improve online algorithms for list labelling, online cycle detection, and incremental shortest paths. In each case, the main idea of the technique is to use predictions to decompose the input into smaller pieces which can be solved more efficiently. I will discuss how this technique works, as well as how to use it in the future: both the potential to apply it to new problems, and common challenges that may arise.

3.11 Machine Learning for Better Algorithms

Benjamin J. Moseley (Carnegie Mellon University - Pittsburgh, US)

This talk explores the emerging area of algorithms with predictions, also known as learning-augmented algorithms. These methods incorporate machine-learned predictions into algorithmic design, allowing the algorithms to adapt to input distributions and achieve improved performance in runtime, space, or solution quality. The presentation will highlight recent advances in leveraging predictions to enhance the efficiency of discrete algorithms. The goal is to overview the breadth of the uses of the model in numerous areas, showcasing prior work and examples.

3.12 From heatmaps to TSP tours

Adam Polak (Bocconi University - Milan, IT)

Joint work of Adam Polak, Marek Eliáš, Fabrizio Grandoni, Eleonora Vercesi

This will be a talk about the travelling salesperson problem. I will show how to improve upon the approximation guarantees of the classic Christofides algorithm by using per-edge predictions as to whether the edge is part of the optimal TSP tour. Our algorithm can be seen as a tool for turning a probabilistic heatmap, produced by a deep neural network, into a discrete solution – which is a crucial step in pipelines of modern neural network solvers for TSP and other combinatorial optimization problems. I will present preliminary empirical results comparing our algorithm to other typical solutions for that step.

3.13 Learning-Augmented Mechanism Design

Golnoosh Shahkarami (MPI für Informatik - Saarbrücken, DE)

In the strategic facility location problem, a set of agents report their locations in a metric space and the goal is to use these reports to open a new facility, minimizing an aggregate distance measure from the agents to the facility. However, agents are strategic and may misreport their locations to influence the facility's placement in their favor. The aim is to design truthful mechanisms, ensuring agents cannot gain by misreporting. This problem was recently revisited through the learning-augmented framework, aiming to move beyond worst-case analysis and design truthful mechanisms that are augmented with (machine-learned) predictions. The focus of this prior work was on mechanisms that are deterministic and augmented with a prediction regarding the optimal facility location. In this paper, we provide a deeper understanding of this problem by exploring the power of randomization as well as the impact of different types of predictions on the performance of truthful learning-augmented mechanisms. We study both the single-dimensional and the Euclidean case and provide upper and lower bounds regarding the achievable approximation of the optimal egalitarian social cost.

3.14 Streaming algorithms with predictions

Ali Vakilian (TTIC - Chicago, US)

In this talk, I'll provide an overview of streaming algorithms with predictions. I'll begin with results for vector-based problems (e.g., frequency estimation, norm estimation, low-rank approximation), and then focus on the use of recent epsilon-accurate predictions in graph streaming settings, specifically for the Max-Cut problem.

3.15 Dynamic Algebraic Algorithms with Predictions

Jan van den Brand (Georgia Institute of Technology - Atlanta, US)

Joint work of Jan van den Brand, Sebastian Forster, Yasamin Nazari, Adam Polak
Main reference Jan van den Brand, Sebastian Forster, Yasamin Nazari, Adam Polak: "On Dynamic Graph Algorithms with Predictions", in Proc. of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024, pp. 3534–3557, SIAM, 2024.
URL https://doi.org/10.1137/1.9781611977912.126

Dynamic algebraic algorithms are data structures that maintain properties of matrices and vector spaces, such as matrix inverse, determinant, rank, etc. They are the main subroutine behind the best algorithms for many dynamic graph problems, such as maintaining reachability, maximum matching size, triangle detection, and many more.

In this talk, we will see techniques that accelerate dynamic algebraic algorithms when predictions about future updates are available. Via reductions, this allows to maintain fully dynamic triangle detection, maximum matching, single-source reachability, in $O(n^{\omega-1} + n\eta_i)$ worst-case update time. Here η_i denotes how much earlier the i-th update occurs than predicted, and $\omega\tilde{2}.372$ is the matrix multiplication exponent. For comparison, there are conditional lower bounds that prove no non-trivial data structures are possible without predictions.

3.16 Machine learning for online matching and integer programming

Ellen Vitercik (Stanford University, US)

When designing algorithms for computational problems in practice, there is often ample data about the application domains in which these algorithms will operate. This data presents a valuable opportunity: it can be leveraged via machine learning (ML) to improve the performance of existing optimization algorithms, help practitioners select among different algorithms, and guide the discovery of entirely new algorithms. However, this promise hinges on several core challenges, including: How to align deep learning architectures with algorithmic tasks, Whether to integrate ML into existing algorithms or train them end-to-end for combinatorial tasks, How to supervise ML models on (NP-hard) algorithmic problems, where ground-truth labels are computationally expensive to obtain, and Whether we can provide robust theoretical guarantees for the resulting algorithms. This talk will showcase solutions through two case studies. The first uses graph neural networks to learn near-optimal online matching policies that generalize beyond their training regime. The second shows how large language models can configure integer programming solvers with only a handful of training instances.

3.17 Beyond-worst-case Data Structures

Sebastian Wild (University of Liverpool, GB)

Lazy search trees [1] are a comparison-based sorted dictionary that smoothly interpolates between binary search trees and efficient priority queues automatically, depending on use. In particular they support faster insertions if many elements are not queried. Lazy B-trees bring this trade-off to external memory. Lazy search trees (of either kind) achieve this interpolation by a beyond-worst-case view of operations on sorted dictionaries, in particular by taking the ranks of queries into account and avoiding insertions to be always preceded by a query. The talk also suggested considering predictions of upcoming queries towards further improvements.

References

1 Sandlund, B., and Wild, S. *Lazy search trees*. In 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS) (pp. 704-715). IEEE.

3.18 CPMA: An Efficient Batch-Parallel Set Without Pointers

Helen Xu (Georgia Institute of Technology - Atlanta, US)

Main reference Brian Wheatman, Randal C. Burns, Aydin Buluç, Helen Xu: "CPMA: An Efficient Batch-Parallel Compressed Set Without Pointers", in Proc. of the 29th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming, PPoPP 2024, Edinburgh, United Kingdom, March 2-6, 2024, pp. 348–363, ACM, 2024.
URL https://doi.org/10.1145/3627535.3638492

This talk introduces the batch-parallel Compressed Packed Memory Array (CPMA), a compressed, dynamic, ordered set data structure based on the Packed Memory Array (PMA). Traditionally, batch-parallel sets are built on pointer-based data structures such as trees because pointer-based structures enable fast parallel unions via pointer manipulation. When compared with cache-optimized trees, PMAs were slower to update but faster to scan. The batch-parallel CPMA overcomes this tradeoff between updates and scans by optimizing for cache-friendliness. On average, the CPMA achieves 3x faster batch-insert throughput and 4x faster range-query throughput compared with compressed PaC-trees, a state-of-the-art batch-parallel set library based on cache-optimized trees. We further evaluate the CPMA compared with compressed PaC-trees and Aspen, a state-of-the-art system, on a real-world application of dynamic-graph processing. The CPMA is on average 1.2x faster on a suite of graph algorithms and 2x faster on batch inserts when compared with compressed PaC-trees. Furthermore, the CPMA is on average 1.3x faster on graph algorithms and 2x faster on batch inserts compared with Aspen. At the end, I will provide some hot takes (opinions expressed here are only my own) on how I think these areas relate to learned indices.

4 Panel discussions

4.1 Industry Research Discussion

Panelists: *Sergei Vassilvitskii (Google – New Yor, US), Rob Johnson (Broadcom – San Jose, US) and Cindy Phillips (Sandia National Labs – Albuquerque, US)*

During the coffee and cake break on Tuesday, we held an informal panel discussion on industry research. Participants asked our panelists questions about their experiences working in their current roles. The discussion was meant to mentor students and junior academics and to offer practical advice for those considering or engaging careers outside academia.

4.2 Women^{+} in Theoretical Computer Science

During the coffee break on Thursday, we held an informal gathering for all self-identifying women and non-binary participants at the workshop. The goal was to provide mentorship and networking opportunity for the junior members. The topics of discussion included teaching and advising challenges as well as career trajectories in academia and industry.

Participants

- Antonios Antoniadis
University of Twente, NL
- Martin Aumüller
IT University of Copenhagen, DK
- Ioana Oriana Bercea
KTH Royal Institute of Technology – Stockholm, SE
- Philip Bille
Technical University of Denmark – Lyngby, DK
- Gerth Stølting Brodal
Aarhus University, DK
- Christian Coester
University of Oxford, GB
- Alexander Conway
Cornell Tech – New York, US
- Michael Dinitz
Johns Hopkins University – Baltimore, US
- Christoph Dürr
Sorbonne University – Paris, FR
- Martin Farach-Colton
NYU – New York, US
- Paolo Ferragina
Sant'Anna School of Advanced Studies – Pisa, IT
- Inge Li Gørtz
Technical University of Denmark – Lyngby, DK
- John Iacono
ULB – Brussels, BE
- Sungjin Im
University of California – Santa Cruz, US
- Riko Jacob
IT University of Copenhagen, DK
- Rob Johnson
Broadcom – San Jose, US
- Hanna Komlós
NYU – New York, US
- László Kozma
FU Berlin, DE
- William Kuszmaul
Carnegie Mellon University – Pittsburgh, US
- Silvio Lattanzi
Google – Barcelona, ES
- Thomas Lavastida
University of Texas – Dallas, US
- Alexander Lindermayr
Universität Bremen, DE
- Quanquan C. Liu
Yale University – New Haven, US
- Samuel McCauley
Williams College – Williamstown, US
- Ulrich Carsten Meyer
Goethe University – Frankfurt am Main, DE
- Benjamin J. Moseley
Carnegie Mellon University – Pittsburgh, US
- Yasamin Nazari
VU Amsterdam, NL
- Kim Thang Nguyen
University of Grenoble, FR
- Debmalya Panigrahi
Duke University – Durham, US
- Cynthia A. Phillips
Sandia National Labs – Albuquerque, US
- Adam Polak
Bocconi University – Milan, IT
- Rajeev Raman
University of Leicester, GB
- Golnoosh Shahkarami
MPI für Informatik – Saarbrücken, DE
- Shikha Singh
Williams College – Williamstown, US
- Clifford Stein
Columbia University – New York, US
- Ola Svensson
EPFL – Lausanne, CH
- Ali Vakilian
TTIC – Chicago, US
- Jan van den Brand
Georgia Institute of Technology – Atlanta, US
- Sergei Vassilvitskii
Google – New York, US
- Ellen Vitercik
Stanford University, US
- Sebastian Wild
University of Liverpool, GB
- Chenyang Xu
East China Normal University – Shanghai, CN
- Helen Xu
Georgia Institute of Technology – Atlanta, US

Report from Dagstuhl Seminar 25182

Challenges and Opportunities of Table Representation Learning

Carsten Binnig[*1], **Julian Martin Eisenschlos**[*2], **Madelon Hulsebos**[*3], **and Frank Hutter**[*4]

1 **TU Darmstadt, DE.** carsten.binnig@cs.tu-darmstadt.de
2 **Google Research – Zürich, CH.** eisenjulian@gmail.com
3 **CWI – Amsterdam, NL.** madelonhulsebos@gmail.com
4 **Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE.** fh@cs.uni-freiburg.de

Abstract

The growing volume and importance of structured data have sparked increasing interest in Table Representation Learning (TRL), an emerging field that leverages neural models to learn abstract, general-purpose representations for tabular data to support a wide range of downstream tasks such as tabular prediction, table question answering, tabular data cleaning, and many more. This seminar gathered the different communities (ML, NLP, IR, DB) who work on this topic to discuss the challenges & long-term vision of this field.

From the organizers: Carsten Binnig, Julian Eisenschlos, Madelon Hulsebos, Frank Hutter.

Seminar April 27 – May 2, 2025 – https://www.dagstuhl.de/25182

2012 ACM Subject Classification Computing methodologies → Natural language processing; Computing methodologies → Neural networks; Information systems → Data management systems

Keywords and phrases applications of table representation learning, benchmarks and datasets for table representation learning, pre-trained (language) models for tables and databases, representation and generative learning for data management and analysis

Digital Object Identifier 10.4230/DagRep.15.4.126

1 Executive Summary

Carsten Binnig (TU Darmstadt, DE)
Julian Martin Eisenschlos (Google Research – Zürich, CH)
Madelon Hulsebos (CWI – Amsterdam, NL)
Frank Hutter (Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE)

The Dagstuhl Seminar 25182, held from April 27 to May 2, 2025, brought together researchers from machine learning, natural language processing, information retrieval, and databases to discuss the challenges and vision for Table Representation Learning (TRL). As structured data continues to grow in volume and importance, TRL aims to build representations that enable downstream tasks such as prediction, question answering, data preparation. The seminar served as a forum to share long-term visions, highlight challenges, and discuss research directions that bridge across these diverse communities.

[*] Editor / Organizer

We opened the seminar with a series of opinionated tutorials that laid out the research landscape. Carsten Binnig, taking a database systems perspective, argued that TRL could help eliminate what he termed the "data tax," "query tax," and "tuning tax" that currently burden users of relational databases. By automating tasks such as query authoring, data cleaning, and performance optimization, TRL could significantly reduce entry barriers to database use. Julian Eisenschlos followed with an NLP-centered overview of table question answering. He reviewed benchmarks and encoding strategies, emphasizing the trade-offs between interpretability and computational cost. He highlighted how pre-training tasks and generation-based understanding link table reasoning with broader modalities such as charts and infographics, while also pointing to the challenges of training models in a post-LLM landscape. Madelon Hulsebos shifted the focus to table semantics, stressing that much of TRL happens "before generating insights" She argued that understanding table- and column-relationships is foundational for data preparation, search and retrieval, and predictive modeling. Her talk questioned whether billion-parameter general-purpose models are necessary for tabular tasks, or whether modular, specialized systems might be more effective. Finally, Frank Hutter presented recent advances in deep learning for tabular prediction. After reviewing early methods, he discussed TabPFN and its extensions, which aim to overcome previous limitations. His talk underscored TRL's growing ability to rival or surpass traditional tabular learning methods while also pointing out open challenges in scaling, context integration, and generalization.

The opinionated tutorials were complemented by shorter impulse talks that addressed specific aspects of TRL. Paolo Papotti reviewed transformer-based adaptations for tabular data, covering innovations in inputs, internals, outputs, and pretraining. Gaël Varoquaux analyzed architectural challenges in building table foundation models, particularly around heterogeneity and invariances. Michael Cochez argued for tighter links between graph learning and TRL to handle complex relations and incomplete data. Xue Effy Li emphasized that meaningful table representation requires integrating contextual information such as metadata, documentation, and world knowledge. Gerardo Vitagliano demonstrated the promise of multimodal pipelines that integrate genomic, imaging, textual, and tabular data for scientific discovery, while exposing current limitations. Andreas Müller shared a vision for deep integration of human feedback, LLMs, databases and table foundation models into agentic systems. Shuaichen Chang discussed the challenges of deploying Text-to-SQL in real-world settings, where ambiguity, noise, and mixed data require robust solutions. Finally, Fatma Özcan showed how long-context reasoning and multi-agent systems can improve Text-to-SQL robustness.

After the talks, we divided into working groups to explore research problems in more depth. The Multi-Modal Data Analysis group examined how to query and process data spanning text, images, genomics, and audio, emphasizing the need for new operators, adaptive indexing, and cost-aware query planning. The Predictive ML and Context group discussed how to integrate metadata, external knowledge, and domain expertise into statistical tabular prediction, proposing hybrid architectures that combine foundation models with agentic systems. The Conversational Analytics group envisioned natural language interfaces that go beyond text-to-SQL by supporting explanations, causal reasoning, and iterative dialogue with human oversight. The Architectures for Table Foundation Models group debated whether a universal foundation model for tables is achievable, weighing trade-offs between adaptability, semantic grounding, and efficiency.

Reflecting on the seminar's discussions, we identified a few cross-cutting themes that seem promising for future research. First, context, whether in the form of metadata, domain knowledge, or multimodal signals, was consistently identified as crucial for making TRL robust and useful. Second, the limitations of current benchmarks remain a bottleneck, as they fail to capture the complexity of real-world tabular reasoning. Finally, participants questioned the pursuit of monolithic "one-size-fits-all" tabular models, favoring instead modular or hybrid systems that can flexibly combine database principles, machine learning, and human expertise. We concluded the seminar with a shared recognition that while TRL has achieved significant progress, its long-term promise lies in deeply integrating methods across disciplines to build more adaptive, interpretable, and context-aware tabular intelligence.

2 Table of Contents

3 Overview of Talks

3.1 An opinionated talk on TRL from a database lens OR Rethinking databases in the Age of LLMs/AI

Carsten Binnig (TU Darmstadt, DE)

Relational databases have been a success story since their incarnation in the 1970s. However, using these systems has extreme high overhead for users. First, extracting data from potentially unstructured or unclear sources and transform those in clean structured schema cause high overheads (= data tax). Moreover, authoring queries using SQL has similarly high overheads as queries (query tax) involve many tables and SQL requires from users to be highly precise in how tables are combined, which attributes are queries, etc.

Finally, relational databases require high overhead of managed tuning to provide high performance (tuning tax).

In this talk, I explore the opportunities of TRL to tackle these overheads and help users with query authoring (text-to-SQL), data engineering (e.g. automated cleaning) and learned tuning.

3.2 Impulse Talk: Text-to-SQL and Table Analysis in the Wild.

Shuaichen Chang (Amazon Web Services – New York, US)

In this talk, we explore the challenges faced by Text-to-SQL systems in real-world applications. These systems must operate effectively over complex and large-scale databases and abstract tasks.

We highlight key issues such as robustness to semantically equivalent variations in databases and user query, and the need to handle noisy user queries, including ambiguous and unanswerable questions.

Additionally, we discuss a particularly realistic setting where structured tables are intertwined with unstructured text, requiring integrated reasoning across both data types for effective data analysis. Crucially, there is no one-size-fit-all solution in such setting. Effective system design must be tailored to the data characteristics.

Looking forward, we envision that AI agent automatically build such systems based on specific data and task.

3.3 Impulse talk: Relational learning and reasoning for table representation leaning

Michael Cochez (VU Amsterdam, NL)

When data becomes complex, including complex relations exist between entities, it becomes more natural to represent as a graph, Besides, in the graph representation learning field, people have experience with constraints. Tasks in this field include attribute and missing link prediction, but also more complex query answering. Especially in cases where the graph is incomplete, which in real world settings is a ways the case, this is interesting. In this setting, we also answer questions like "Give me all scientists that have worked A in 3 or more universities, in at least 2 countries", which requires joins, counting, iterating, and aggregation. They have also limitations regarding the quality of benchmarks and scalability of methods. Overall, I want to see more collaborations between the graph and the table so fields.

3.4 An Opinionated Overview of Deep Learning for Table Question Answering: An NLP Perspective

Julian Martin Eisenschlos (Google Research – Zürich, CH)

Through this talk I will define visual language and TR utility, with a focus on table understanding. I will introduce the main benchmarks and the ways the community has tackled them over the last couple of years. We'll deep dive into the way tables are encoded through embedding, transfer, and the semantics of the answer trading off interpretability and cost of acquisition. We will also touch on the more trend pre-training tasks: how generation-driven understanding and how understanding table helps in chart and infographic understanding. To close, we will highlight what are the training challenges in a world post LLMs.

3.5 An opinionated talk on Table Representation Learning: on everything before you get the insight

Madelon Hulsebos (CWI – Amsterdam, NL)

Getting insights from structured data comprises many subsequent steps. In this talk I reflect on a critical construct that enables downstream insights coming from table semantics. Particular column semantics, regarding a single column, relationship across columns and tables, are instrumental for data integration, cleaning, search, as well as question answering and predictive modeling.

Reflecting on the performance of small and specialized models versus billion-parameter models for type prediction raises the general question whether we need high-capacity general-purpose tabular models for every task. Do we need to work towards these models, or think more about compound systems (with mixtures, modularity, etc.) for tabular tasks?

The talk continues with a building block for advancing TRL: data & metrics to measure realistic performance on real-world tasks & data. What are the right datasets, tasks, metrics to push TRL further? Early insights show that tabular analytical reasoning capabilities of LLMs and specialized tabular models are far behind.

I conclude with the shared vision of end-to-end tabular analysis: should this be multi-agent systems that abstract humans away? No. We need intersection with domain expertise. But it is a dot on the horizon…

3.6 An Opinionated Overview of Deep Learning for Tabular Prediction

Frank Hutter (Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE)

In this talk, I will motivate the tabular prediction problem to the TRL community and discuss the state of deep learning for solving it. Specifically, I will briefly discuss early tabular deep learning methods, explain TabPFN in some detail, and then focus on the many extensions that have been introduced to move beyond the limitations of TabPFN.

3.7 Impulse Talk: Beyond Tables: Context Matters

Xue Li (CWI – Amsterdam, NL)

Tabular Data rarely exist in isolation – they are embedded in broader ecosysten of documentations, meta data, domain knowledge, and even world knowledge. In this talk, I suggest that meaningful table representation requires incorporating global context for downstream tasks such as TableDA, Text. etc. Global context can include source documentation, organizing organizational processes, and even common-sense or parametric knowledge stored in large Language models. I highlight recent approaches including RAG, Graph RAG. and prompting – based techniques for retrieving and integrating external context, and discuss why these solutions are far from sufficient.. I outline key technical challenger, meta- questions about context retrieval and utilization, and opportunities towards more holistic table understanding.

3.8 Impulse Talk: Transformer-based Table Representation Learning

Paolo Papotti (EURECOM – Biot, FR)

In the last few years, the NLP community witnessed advances in neural representations of free-form text with transformer-based architectures for producing language models (LMs). Given the importance of knowledge in relational tables, recent efforts extended LMs by developing neural representations for tabular data. In this talk, I present these proposals covering extensions to the original transformer in terms of input, internals (such as attention), output, and pre-training task.

3.9 Impulse Talk: Architectures for tabular learning

Gaël Varoquaux (INRIA Saclay – Île-de-France, FR)

If we want to solve tabular learning problems, what are the constraints and advances in architectures for table foundation models? We want to construct representations that can be reused to facilitate learning and/or transfer learning. For this goal, we need to think about representations of tables and how to represent mixed data types, as well as how to learn with these representations.

3.10 Impulse talk: Beyond tables, Multimodal pipelines

Gerardo Vitagliano (MIT – Cambridge, US)

Real-world data science is often complex and insights or outputs derive from data collected in a wide variety of modalities. In this talk, we present an example of insightful science based on a multimodal research scenario: how do researchers find and characterize tumor progression using cancer patient's big clinical records?

We discuss how to process this information, recalling process data in a variety of modalities: genomic, imaging (MRI included), tabular, and free text formats. Through several examples of data processing, we outline the limitations of state-of-the-art multimodal processing systems, as well as processing approximations (approximate query processing, data summarization, embeddings, indexing).

We outline three macro-areas of research: multimodal embeddings, retrieval strategies, integration and processing joint modalities.

Within these systems we envision all of these components will play a fundamental role to enable complex pipelines and integrative analysis of multimodal data. State-of-the-art AI models process data of various modalities, including text, image and sound. However, the context size of these models is severely limited, making it impossible to directly apply them to large data collections with millions of entries. Even if it is possible to directly apply models to large data collections, it is often prohibitively expensive.

Our goal was to create data processing systems that scale up multimodal data processing to large data collections, leveraging techniques such as approximate processing (processing a carefully selected data subset to obtain an approximated result), caching (reusing results from prior queries to answer new queries more efficiently), and compression. Compressing data to make processing with AI models cheaper. In addition, we plan to create a benchmark that allows us to evaluate such systems for multimodal data processing, according to metrics such as result quality and monetary processing fees.

3.11 Impulse Talk: Text-to-SQL – Long Context

Fatma Özcan (Google – San Jose, US)

Text-to-SQL is challenging in that a natural language question is inherently ambiguous, while SQL generation requires a precise understanding of complex data schema and semantics. One solution is to provide sufficient contextual information. In this talk, I present a detailed study on the performance and the latency trade-offs long context offered by Google's Gemini model, showing the impact of various contextual information, including example column values, user-provided hints, in-context examples, SQL documentation, and relevant schema. I then describe CHASE-SQL, a novel multi-agent solution that uses LLMs to generate diverse SQL candidates using three different approaches, choosing the final answer by a selection agent.

4 Working groups

4.1 Working group: Multi-Modal Data Analysis

Michael Cochez (VU Amsterdam, NL), Tianji Cong (University of Michigan – Ann Arbor, US), Andreas Kipf (TU Nürnberg, DE), Olga Ovcharenko (TU Berlin, DE), Fatma Özcan (Google – San Jose, US), Shivam Sharma (TU Darmstadt, DE), Immanuel Trummer (Cornell University – Ithaca, US), and Gerardo Vitagliano (MIT – Cambridge, US)

Multi-modal data analysis addresses the challenge of querying and processing data across diverse modalities – numbers, text, images, audio, video, graphs, genomics, and time series – often within relational tables. Multimodality is not simply a matter of semantic variation; it requires system-level design to handle distinct representations and access patterns. Modern representation learning allows late fusion, reducing the need for early manual integration.

A multi-modal database system must address storage, querying, and processing. Storage may involve tables with references to external modalities, adaptive indexing, and multimodal database cracking. Querying may take the form of SQL, natural language, or operator pipelines, and requires specialized operations such as cross-modal joins, transformations (for example, speech-to-text), and multimodal search. Because multimodal analysis is computationally expensive, optimization strategies are essential: approximations, pruning redundant modalities, batching, caching, and cost-aware query planning.

Applications span e-commerce recommendations, bioacoustics, healthcare analytics, multimedia search, and scientific domains. Benchmarks are crucial, ideally combining real datasets with diverse query types and measuring not only accuracy but also cost, latency, and scalability. The overarching insight is that multimodal relational learning is achievable, but only with new operators, optimization techniques, and dedicated benchmarks that acknowledge the trade-offs between cost and accuracy.

4.2 Working group: Predictive ML and Context

Frank Hutter (Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE), Katharina Eggensperger (Universität Tübingen, DE), Myung Jun Kim (INRIA Saclay – Île-de-France, FR), Xue Li (CWI – Amsterdam, NL), Lennart Purucker (Universität Freiburg, DE), and Sebastian Schelter (TU Berlin, DE)

Predictive machine learning on tabular data raises the question of how best to incorporate context and world knowledge. Current approaches exist along a spectrum: at one end, large language models can be fine-tuned on company data and prompted with customer information; at the other, traditional pipelines engineer features from available sources and train standard supervised models. Both approaches show limitations – pre-trained models often cannot disambiguate column semantics, while purely tabular methods struggle without interaction and ignore implicit meanings. In-between approaches such as CARTE, TARTE, and TabPFN attempt to combine graph or probabilistic structures with tabular input.

Context encompasses many forms: metadata, additional text, provenance, external knowledge bases, and domain-specific constraints. Agents can act as junior data scientists, augmenting tables, searching for relevant data, and performing feature engineering. A hybrid vision emerges: a world model provides semantic grounding, a statistical foundation model delivers predictive accuracy, and agentic systems serve as the glue between them. While the long-term goal may be a unified end-to-end model, a modular design currently appears more viable. Benchmarks and interfaces remain underdeveloped, highlighting a research challenge in marrying contextual world knowledge with robust statistical learning.

4.3 Working group: Converstaional Analytics

Andreas Müller (Microsoft Corp. – Mountain View, US), Carsten Binnig (TU Darmstadt, DE), Shuaichen Chang (Amazon Web Services – New York, US), Madelon Hulsebos (CWI – Amsterdam, NL), and Anupam Sanghi (TU Darmstadt, DE)

Conversational analytics seeks to enable dialogue-driven data analysis in natural language, extending far beyond text-to-SQL translation. The aim is not only to retrieve results but also to provide explanations, reason about causes, and support human decision-making. A central question concerns whether systems should rely on explicit planning – through a formal algebra of operations – or on step-by-step reasoning by large models. Planning and orchestration resemble challenges in robotics, where reinforcement learning and tool management are key.

A conversational algebra can capture the steps of analysis: retrieving data, identifying granularity, selecting measures, reshaping data, choosing analytical methods, applying guardrails, and validating outcomes. Human involvement is essential for reviewing intermediate results, correcting assumptions, and guiding decisions, while automated components orchestrate tools and workflows. Risks include hallucinated results, invalid causal inferences, and misleading reasoning traces, underscoring the need for transparency and verifiability.

Existing benchmarks such as Spider 2.0 and DiscoveryBench offer starting points but fail to capture the full complexity of reasoning. Progress in this area requires richer benchmarks and formal abstractions, combined with human-in-the-loop oversight to ensure reliability and explainability.

4.4 Working group: Architectures for Table Foundation Models

Gaël Varoquaux (INRIA Saclay – Île-de-France, FR), Vadim Borisov (tabularis.ai – Tübingen, DE), Julian Martin Eisenschlos (Google Research – Zürich, CH), Floris Geerts (University of Antwerp, BE), Filip Gralinski (Snowflake – Warsaw, PL), Tassilo Klein (SAP SE – Walldorf, DE), Paolo Papotti (EURECOM – Biot, FR), and Liane Vogel (TU Darmstadt, DE)

Table foundation models (TFMs) sit at the intersection of multiple research traditions: statistical estimation in machine learning, relational structures in databases, and semantic grounding in NLP. The guiding question is whether a single general-purpose foundation model can cover all tasks involving tabular data, or whether domain-specific variants will remain necessary. Trade-offs span adaptability versus cost, the diversity of supported inputs, and the role of world knowledge alongside statistical learning.

Architectural choices extend beyond transformer-based approaches to include graph neural networks, symbolic reasoning components, and modules specialized for numbers and time series. Key issues include which invariances should be respected, such as row and column order, and how to scale computation for very large tables. Pretraining raises questions of data sourcing: synthetic datasets may capture structural patterns, while real and domain-specific data provide semantic grounding. Evaluation of TFMs requires a multidimensional benchmark framework covering adaptability, scalability, modality handling, semantic understanding, and efficiency. The vision is not to treat TFMs as scaled-up language models but as flexible, principled architectures that integrate statistical inference, semantic reasoning, and database knowledge.

Participants

- Carsten Binnig
TU Darmstadt, DE
- Vadim Borisov
tabularis.ai – Tübingen, DE
- Shuaichen Chang
Amazon Web Services – New York, US
- Michael Cochez
VU Amsterdam, NL
- Tianji Cong
University of Michigan – Ann Arbor, US
- Katharina Eggensperger
Universität Tübingen, DE
- Julian Martin Eisenschlos
Google Research – Zürich, CH
- Floris Geerts
University of Antwerp, BE
- Filip Gralinski
Snowflake – Warsaw, PL
- Madelon Hulsebos
CWI – Amsterdam, NL
- Frank Hutter
Prior Labs – Freiburg, DE & ELLIS Institute Tübingen, DE & Universität Freiburg, DE
- Myung Jun Kim
INRIA Saclay – Île-de-France, FR
- Andreas Kipf
TU Nürnberg, DE
- Tassilo Klein
SAP SE – Walldorf, DE
- Xue Li
CWI – Amsterdam, NL
- Andreas Müller
Microsoft Corp. – Mountain View, US
- Fatma Özcan
Google – San Jose, US
- Olga Ovcharenko
TU Berlin, DE
- Paolo Papotti
EURECOM – Biot, FR
- Lennart Purucker
Universität Freiburg, DE
- Anupam Sanghi
TU Darmstadt, DE
- Sebastian Schelter
TU Berlin, DE
- Shivam Sharma
TU Darmstadt, DE
- Immanuel Trummer
Cornell University – Ithaca, US
- Gaël Varoquaux
INRIA Saclay – Île-de-France, FR
- Gerardo Vitagliano
MIT – Cambridge, US
- Liane Vogel
TU Darmstadt, DE