DAGSTUHL
REPORTS

Volume 14, Issue 4, April 2024

Dagstuhl Reports, Vol. 14, 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
October, 2024

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.14.4.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.

Editorial Office
Michael Wagner *(Managing Editor)*
Michael Didas *(Managing Editor)*
Jutka Gasiorowski *(Editorial Assistance)*
Dagmar Glaser *(Editorial Assistance)*
Thomas Schillo *(Technical Assistance)*

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 24141

Network Calculus

Steffen Bondorf*[1], **Anne Bouillard***[2], **Markus Fidler***[3], **Jörg Liebeherr***[4], **and Lisa Maile**†[5]

1 **Ruhr-Universität Bochum, DE.** steffen.bondorf@rub.de
2 **Huawei Technologies – Boulogne-Billancourt, FR.** anne.bouillard@huawei.com
3 **Leibniz Universität Hannover, DE.** markus.fidler@ikt.uni-hannover.de
4 **University of Toronto, CA.** jorg@ece.utoronto.ca
5 **Universität Erlangen-Nürnberg, DE.** maile@ida.ing.tu-bs.de

Abstract

Network calculus is a versatile method for analysing queueing systems with applications in Internet Quality of Service (QoS), wireless networks, Ethernet with delay guarantees, real-time systems, and feedback control. Using min-plus or max-plus algebra and deterministic or stochastic bounds, this Dagstuhl Seminar aims to bring together the deterministic and stochastic network calculus community to discuss recent research, future directions, and collaboration.

The modelling power of network calculus allows it to represent different systems, making it applicable to a wide variety of queueing problems. Thus, it has been proposed in various contexts for new emerging technologies such as IEEE Time Sensitive Networking (TSN), IETF Deterministic Networking (DetNet), and 5G Ultra-Reliable Low Latency Communications (URLLC), with important applications in factory automation, aerospace onboard, and automotive in-vehicle networks.

The two communities of deterministic and stochastic network calculus have grown closer in recent years, for example, as deterministic network calculus results have been incorporated into stochastic network calculus, demonstrating the need for and value of strong collaboration between the communities.

Recent developments in network calculus algorithms include modular and optimisation approaches, parallelizable methods that improve performance bounds, machine learning techniques, but also the adaptation of network calculus for design automation and system configuration.

This report documents the programme, the new contributions, and the results of Dagstuhl Seminar 24141 “Network Calculus”.

Seminar April 1–4, 2024 – https://www.dagstuhl.de/24141

2012 ACM Subject Classification Computer systems organization → Real-time systems; Networks → Network performance analysis; Networks → Network performance modeling

Keywords and phrases age of information, effective bandwidths, network calculus, performance evaluation, queueing network

Digital Object Identifier 10.4230/DagRep.14.4.1

* Editor / Organizer

† Editorial Assistant / Collector

Network Calculus, *Dagstuhl Reports*, Vol. 14, Issue 4, pp. 1–22
Editors: Steffen Bondorf, Anne Bouillard, Markus Fidler, and Jörg Liebeherr
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Lisa Maile (Universität Erlangen-Nürnberg, DE)

Our Dagstuhl Seminar brought together leading experts in the field of network calculus to discuss recent research activities, identify future directions, and establish or strengthen cooperation. The seminar fostered collaborations and new ideas by focusing on diversity, incorporating perspectives from research and industry, varying levels and areas of expertise, and participants of different positions and ages. Attendees came from various scientific disciplines, including deterministic and stochastic network calculus, Age-of-Information, industrial communication, and wireless technologies. All combined under the need for Quality of Service (QoS) requirements. Participants contributed scientific presentations and workshops on their current and future research.

Network calculus is a theoretical framework for analyzing networks, mainly, but not limited to communication networks, but applications are also possible in the field of energy systems or even emergency call centers. By modeling data flows and systems as mathematical functions, network calculus allows the calculation of guaranteed (deterministic/stochastic) upper bounds. This analysis can be applied to various elements, including individual hops or entire systems. Additionally, network calculus has been used for certification purposes in avionic networks, such as the Airbus A380 [1], demonstrating its importance and applicability in designing safety-critical systems that require stringent QoS performance guarantees.

The seminar focused on several main areas: integrating algebras and performance metrics, exploring network topology and parallel systems, applying network calculus to emerging technologies, and developing algorithms and tools. New insights in these topics were presented, followed by group work on individual problems. These group sessions explored algorithmic challenges, mathematical and computational complications, and practical applications of network calculus.

In the context of algebras and performance metrics, discussions centered on representing network calculus as a systems theory under min-plus algebra, as well as integrating min-plus and max-plus algebras. This dual approach could provide a more comprehensive framework for analyzing complex network behaviors. A significant challenge addressed was the issue of rare perturbations, where deterministic network calculus currently lacks the ability to quantify the frequency or rarity of worst-case scenarios. Participants discussed the need for more nuanced metrics to distinguish between frequent low delays and rare high delays.

The seminar also highlighted the powerful modeling capabilities of network calculus in representing various systems, such as links, traffic shapers, and scheduling policies, which can be composed into arbitrary topologies. Advanced results from deterministic network calculus, like pay bursts and multiplexing only once phenomena, have recently been incorporated into stochastic network calculus, necessitating strong cooperation between the deterministic and stochastic communities. Parallel systems, particularly in the context of multi-path transport and fork-join models, were another central point, with discussions addressing the complexities and opportunities inherent in these configurations.

Emerging technologies such as Time-Sensitive Networking (TSN), Deterministic Networking (DetNet), and Ultra-Reliable Low Latency Communications (URLLC) were also a significant focus of the seminar. Network calculus currently receives a lot of research attention due to these emerging topics, which attract high interest from both industry and academia. These technologies are crucial for applications in factory automation, aerospace onboard systems, and automotive in-vehicle networks, where performance guarantees are critical. The seminar emphasized the integration of network calculus with these emerging technologies to ensure robust performance analysis and guarantees, supporting applications that meet stringent reliability and latency requirements.

Recent advancements in network calculus algorithms were another key theme of the seminar. Participants discussed combining modular and optimization approaches and developing highly parallelizable methods that iteratively improve performance bounds. The use of machine learning for appropriate decomposition in modular approaches was highlighted, allowing for more efficient and scalable solutions to complex network performance challenges. Additionally, the development of tools to rapidly disseminate novel results and facilitate extensive community-based research artifact evaluation and reproducibility verification was emphasized.

Discussions following each presentation helped identify open problems and challenges to be addressed by the community. Participants highlighted the importance of collaboration between the different network calculus communities and experts, the topics which are still open and where our community could focus in the future, and potentials to make network calculus more accessible to the public.

The seminar also included connecting personal and scientific discussions during a slightly rainy but enjoyable hike, long extended dinners, and quizzes to engage in team building. These social events provided additional opportunities for participants to engage and exchange ideas.

In summary, the Dagstuhl Seminar on network calculus facilitated significant discussions on advancing the field through interdisciplinary collaboration, integrating new technologies, and leveraging new methodologies for optimization. The insights and outcomes from the seminar pave the way for future research and development, ensuring network calculus receives recognition and remains a robust and versatile tool for network analysis and performance guarantees.

References

1 F. Francés, C. Fraboul, and J. Grieu, *Using Network Calculus to optimize the AFDX network*, ERTS 2006: 3rd European Congress ERTS Embedded real-time software, 2006.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

3 Overview of Talks

3.1 Quasi-Deterministic Burstiness Bound for Aggregate of Independent, Periodic Flows

Anne Bouillard (Huawei Technologies – Boulogne-Billancourt, FR)

Joint work of Hossein Tabatabaee, Anne Bouillard, Jean-Yves Le Boudec
Main reference Seyed Mohammadhossein Tabatabaee, Anne Bouillard, Jean-Yves Le Boudec: "Quasi-Deterministic Burstiness Bound for Aggregate of Independent, Periodic Flows", in Proc. of the Quantitative Evaluation of Systems – 20th International Conference, QEST 2023, Antwerp, Belgium, September 20-22, 2023, Proceedings, Lecture Notes in Computer Science, Vol. 14287, pp. 205–220, Springer, 2023.
URL https://doi.org/10.1007/978-3-031-43835-6_15

Time-sensitive networks require timely and accurate monitoring of the status of the network. To achieve this, many devices send packets periodically, which are then aggregated and forwarded to the controller. Bounding the aggregate burstiness of the traffic is then crucial for effective resource management. In this paper, we are interested in bounding this aggregate burstiness for independent and periodic flows. A deterministic bound is tight only when flows are perfectly synchronized, which is highly unlikely in practice and would be overly pessimistic.

We compute the probability that the aggregate burstiness exceeds some value. When all flows have the same period and packet size, we obtain a closed-form bound using the Dvoretzky–Kiefer–Wolfowitz inequality. In the heterogeneous case, we group flows and combine the bounds obtained for each group using the convolution bound. Our bounds are numerically close to simulations and thus fairly tight. The resulting aggregate burstiness estimated for a non-zero violation probability is considerably smaller than the deterministic one: it grows in square root of n log n, instead of n, where n is the number of flows.

3.2 Dynamics of energy storage systems with self-discharge

Almut Burchard (University of Toronto, CA)

Joint work of Almut Burchard, Majid Raeis, Jörg Liebeherr

Energy storage is a crucial component of the smart grid, since it provides the ability to buffer transient fluctuations of the energy supply from renewable sources. Even without a load, energy storage systems experience a reduction of the stored energy through self-discharge. This talk presents analysis of the self-discharge phenomenon using a queueing system model, which we refer to as leakage queue. When the average net charge is positive, we discover that the leakage queue operates in one of two regimes: a leakage-dominated regime and a capacity-dominated regime. We find that in the leakage-dominated regime, the stored energy stabilizes at a point that lies well below the storage capacity. The predictions are validated in a numerical example where the energy supply resembles a wind energy source.

3.3 Automata-Theoretic Characterizations of Real-Time Calculus

Samarjit Chakraborty (University of North Carolina at Chapel Hill, US)

Real-Time Calculus (RTC) [2, 7] has proven to be a general framework for analyzing a variety of distributed and heterogeneous real-time systems that employ different scheduling policies. By drawing principles from the theory of Network Calculus, RTC uses *arrival curves* to model how computation and communication tasks in a real-time system are triggered, or in other words, the workload in a system. Similarly, *service curves* are used to model how these tasks are served, or the computation and communication bandwidth available to a task. With the output of a computation or communication task *triggering* a subsequent task on the same or a different resource, by using the theory of max plus and min plus algebra, the framework helps in analyzing how workload flows through a network of computation and communication resources and derives the timing properties of the tasks mapped onto these resources. With this analysis being compositional, RTC has been found to be effective in analyzing the timing properties of large networked embedded systems in scalable fashion.

The analysis in RTC uses algebraic techniques and is "functional" in nature, and does not allow the modeling of "state" in a straightforward manner. This talk discussed how timing properties of a cyber-physical system (CPS) can be modeled using principles of RTC, but using a corresponding automata-theoretic model. Representing an *arrival curve* as a finite automaton allows access to a rich set of automata-theoretic modeling and verification tools, including model checking. Currently, the control strategy in a CPS is determined independent of the characteristics of the platform running the software-implementation of the controller. The control strategy typically includes a sampling period and a required sensor-to-actuator delay that the engineer implementing the controller has to guarantee using suitable scheduling techniques. While such a design flow follows the principle of "separation of concerns" and distinguishes between a model and its implementation, it is increasingly turning out to be overly conservative. This is because the strict separation between design and implementation carries no information on how much deviation in the timing behavior determined at the controller design stage is safe. As a result, the notion of safety has been synonymous with "meeting all deadlines." This is a difficult goal to meet, given the complexities of modern implementation platforms and the increasing volume of software code that is implemented on them in various domains such as automotive or robotics.

To address this, the talk outlined a notion of *safety*, where any trajectory of the closed-loop system (*i.e.,* plant + controller) in its state space is considered to be safe as long as it is contained within a predefined safety pipe around its nominal trajectory. Such a nominal trajectory could be the system trajectory when the controller is subjected to an ideal timing behavior (*i.e.*, it experiences no deadline misses). The question is to then determine which timing behaviors result in safe trajectories? For this, deadline hit/miss patterns are represented as RTC arrival curve-like *weakly-hard* constraints. Whether a chosen weakly-hard constraint is safe or not is determined through a safe but approximate reachability analysis of the closed-loop system [5, 10]. It is then noted that any weakly-hard constraint – that represents a set of binary strings capturing patterns of deadline hits or misses experienced by a control task – can be represented by an equivalent finite automaton. Since the union of regular languages is also regular, a collection of finite automata representing safe timing timing behaviors of the closed-loop system can be represented by a single finite automaton. Now, consider a set of controllers that need to be implemented on the same shared platform, with their utilization being greater than 100% if *all* of their deadlines are to be met. Is it possible to schedule them such that each controller misses some deadlines, but are nevertheless *safe*, following the notion of safety outlined above?

The talk outlined a method to address this question. It showed that the product of the automaton corresponding to the safe timing behaviors of the individual closed-loop systems contains such schedules. More details on this may be found in [8, 9]. Checking whether a given weakly-hard constraint is safe, involves solving a reachability analysis problem, which is computationally very expensive. This was addressed by solving a safe but an *approximate* reachability analysis problem, which can be pessimistic, *i.e.*, return "unsafe" for weakly-hard constraints that are actually safe. This was addressed by using stochastic hypothesis testing based statistical verification techniques [4]. These are computationally less expensive and are less pessimistic, but provide parametrizable probabilistic guarantees.

A natural question that arises in the context of using automata-theoretic representations of RTC is: *why not use formalisms like timed automata?* This is because they have also proven to be very effective for the modeling and verification of real-time systems and are amenable to model checking. The talk concluded by discussing that RTC provides a "count-based abstraction" that is useful and sufficient in many setups. Here, it is sufficient to record *how many deadlines were missed?* Recording the precise times at which the individual deadlines were missed or when tasks were triggered – as would be done by formalisms like timed automata – would not be necessary. The schedule synthesis techniques outlined in [8, 9] from the automata representations of RTC-based weakly-hard constraints are also simpler than those that would be necessary if formalisms like timed automata would be used to represent the safe timing behaviors of the different controllers sharing computation and communication resources. Other automata-theoretic representations of RTC may be found in [1, 3, 6].

References

1 A. Bouillard, L. T. X. Phan, and S. Chakraborty. Lightweight modeling of complex state dependencies in stream processing systems. In *15th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2009.

2 S. Chakraborty, S. Künzli, and L. Thiele. A general framework for analysing system properties in platform-based embedded system designs. In *Design, Automation and Test in Europe Conference and Exposition (DATE)*, 2003.

3 S. Chakraborty, L. T. X. Phan, and P. S. Thiagarajan. Event count automata: A state-based model for stream processing systems. In *26th IEEE Real-Time Systems Symposium (RTSS)*, 2005.

4 B. Ghosh, C. Hobbs, S. Xu, F. D. Smith, J. H. Anderson, P. S. Thiagarajan, B. Berg, P. S. Duggirala, and S. Chakraborty. Statistical verification of autonomous system controllers under timing uncertainties. *Real Time Syst.*, 60(1):108–149, 2024.

5 C. Hobbs, B. Ghosh, S. Xu, P. S. Duggirala, and S. Chakraborty. Safety analysis of embedded controllers under implementation platform timing uncertainties. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 41(11):4016–4027, 2022.

6 L. T. X. Phan, S. Chakraborty, and P. S. Thiagarajan. A multi-mode real-time calculus. In *29th IEEE Real-Time Systems Symposium (RTSS)*, 2008.

7 L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2000.

8 S. Xu, B. Ghosh, C. Hobbs, P. S. Thiagarajan, and S. Chakraborty. Safety-aware flexible schedule synthesis for cyber-physical systems using weakly-hard constraints. In A. Takahashi, editor, *28th Asia and South Pacific Design Automation Conference (ASPDAC)*, 2023.

9 S. Xu, B. Ghosh, C. Hobbs, P. S. Thiagarajan, P. Joshi, and S. Chakraborty. Safety-aware implementation of control tasks via scheduling with period boosting and compressing. In *29th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2023.

10 A. Yeolekar, R. Metta, C. Hobbs, and S. Chakraborty. Checking scheduling-induced violations of control safety properties. In *20th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, volume 13505 of *Lecture Notes in Computer Science*. Springer, 2022.

3.4 Impact of AS6802 Synchronization Protocol on Time- Triggered and Rate-Constrained traffic

Anaïs Finzi (TTTech Computertechnik – Wien, AT)

TTEthernet is an Ethernet-based synchronized network technology compliant with the AFDX standard. It supports safety-critical applications by defining different traffic classes: Time-Triggered (TT), Rate-Constrained (RC), and Best-Effort traffic. The synchronization is managed through the AS6802 protocol, which defines so-called Protocol Control Frames (PCFs) to synchronize the local clock of each device. In this presentation, we analyze the synchronization protocol to assess the impact of the PCFs on TT and RC traffic.

We propose a method to decrease the impact of PCFs on TT and a new Network Calculus model to compute RC delay bounds with the influence of both PCF and TT traffic. We finish with a performance evaluation to i) assess the impact of PCFs, ii) show the benefits of our method in terms of reducing the impact of PCFs on TT traffic and iii) prove the necessity of taking the PCF traffic into account to compute correct RC worst-case delays and provide a safe system.

3.5 Exploiting Minimal Arrival Curves To Deal With Negative Service Curves

Anja Hamscher (RPTU – Kaiserslautern, DE) and Vlad-Cristian Constantin (RPTU – Kaiserslautern, DE)

Joint work of Anja Hamscher, Vlad-Cristian Constantin, Jens B. Schmitt
Main reference Anja Hamscher, Vlad-Cristian Constantin, Jens B. Schmitt: "Extending Network Calculus To Deal With Partially Negative And Decreasing Service Curves", CoRR, Vol. abs/2403.18042, 2024.
URL https://doi.org/10.48550/ARXIV.2403.18042

When considering multiple flow scenarios, strictness of service curves is required to obtain a residual service curve for a single flow. Without strictness, the residual service curve exhibits an interval over which it is negative and decreasing, i.e., it is not in $\mathcal{F}_0^{\uparrow}$ anymore. However, this assumption is needed to calculate performance bounds. Introducing a lower bound on the arrivals to a system, the inherent issue can be avoided.

In our talk, we discuss the issue at hand and show how the existing performance bounds can be adjusted to work with service curves that are not in $\mathcal{F}_0^{\uparrow}$. We also discuss potential applications.

3.6 Towards a Calculus for Wireless Networks: Recent Advances and What's Next

Yuming Jiang (NTNU – Trondheim, NO)

In this talk, a quick recap on the "Wireless Network Calculus" talk at Dagstuhl Seminar 15112 on Network Calculus is first provided. Then, the focus is on introducing some fundamental advances in the topic since that talk. They include the light-tailed property of wireless channel capacity, the dependence structure and copula analysis of wireless channel capacity, and Spatial Network Calculus for wireless networks. The third part is devoted to reliable performance, e.g. 99% packet delivery ratio, in wireless mesh or massive machine-type communications (mMTC). Experimental results from real word IoT wireless networks are used to demonstrate that there is a huge gap. To bridge, an idea is introduced at the end. It is to coordinate transmission among nodes such that the transmission nodes satisfy certain constraints with which the required reliability can be ensured from Spatial Network Calculus analysis.

Main references for further reading:

- F. Sun and Y. Jiang. A Statistical Property of Wireless Channel Capacity: Theory and Application. IFIP Performance 2017
- F. Sun and Y. Jiang. Stochastic Dependence in Wireless Channel Capacity: A Hidden Resource. arXiv 1711.10363 (2017/2019)
- K. Feng and F. Baccelli. Spatial Network Calculus and Performance Guarantees in Wireless Networks. IEEE Trans. Wireless Communications. 23(5): 5033-5047 (2024)

3.7 Time Sensitive Networks and Network Calculus

Jean-Yves Le Boudec (Jouxtens-Mézery, CH)

Joint work of Jean-Yves Le Boudec, Ahlem Mifdaoui, Ehsan Mohammadpour, Eleni Stai, Ludovic Thomas

Time Sensitive Networks offer guarantees on worst-case delay, worst-case delay variation and zero congestion loss; in addition, they provides mechanisms for packet duplication in order to hide residual losses due to transmission errors. They find applications in many areas such as factory automation, embedded and vehicular networks, audio-visual studio networks, and in the front-hauls of cellular wireless networks. In this talk we describe recent network-calculus results that can be used to analyze time sensitive networks with components such as packet ordering and duplicate removal functions, schedulers, regulators, dampers. We explain why clock non-idealities matter and how to take them into account.

References

1 Ehsan Mohammadpour, Eleni Stai, Jean-Yves Le Boudec: Improved Network-Calculus Nodal Delay-Bounds in Time-Sensitive Networks. IEEE/ACM Trans. Netw. 31(6): 2902-2917 (2023)

2 Ehsan Mohammadpour, Jean-Yves Le Boudec: On Packet Reordering in Time-Sensitive Networks. IEEE/ACM Trans. Netw. 30(3): 1045-1057 (2022)

3 Ludovic Thomas, Jean-Yves Le Boudec: On Time Synchronization Issues in Time-Sensitive Networks with Regulators and Nonideal Clocks. Proc. ACM Meas. Anal. Comput. Syst. 4(2): 27:1-27:41 (2020)

4 Ehsan Mohammadpour, Jean-Yves Le Boudec: Analysis of Dampers in Time-Sensitive Networks With Non-Ideal Clocks. IEEE/ACM Trans. Netw. 30(4): 1780-1794 (2022)

3.8 Network Calculus Characterization of Congestion Control

Harvinder Lehal (University of Toronto, CA)

Joint work of Harvinder Lehal, Natchanon Luangsomboon, Jörg Liebeherr
Main reference Harvinder Lehal, Natchanon Luangsomboon, Jörg Liebeherr: "Network Calculus Characterization of Congestion Control for Time-Varying Traffic", CoRR, Vol. abs/2403.15303, 2024.
URL https://doi.org/10.48550/ARXIV.2403.15303

Models for the dynamics of congestion control generally involve systems of coupled differential equations which assume that traffic sources saturate the maximum transmissions allowed by the congestion control method. This is not suitable for studying congestion control of intermittent but bursty traffic sources. In this talk, we presented a characterization of congestion control for arbitrary time-varying traffic that applies to rate-based as well as window-based congestion control. We leverage the capability of network calculus to precisely describe the input-output relationship at network elements for arbitrary source traffic and show that our characterization can closely track the dynamics of even complex congestion control algorithms.

3.9 Decentralized Reservation Protocols in Time-Sensitive Networking: Applying Network Calculus Without Central Network Overview

Lisa Maile (Universität Erlangen-Nürnberg, DE)

Joint work of Lisa Maile, Dominik Voitlein, Alexej Grigorjew, Kai-Steffen Jens Hielscher, Reinhard German
Main reference Lisa Maile, Dominik Voitlein, Alexej Grigorjew, Kai-Steffen Jens Hielscher, Reinhard German: "On the Validity of Credit-Based Shaper Delay Guarantees in Decentralized Reservation Protocols", in Proc. of the 31st International Conference on Real-Time Networks and Systems, RTNS 2023, Dortmund, Germany, June 7-8, 2023, pp. 108–118, ACM, 2023.
URL https://doi.org/10.1145/3575757.3593644

Resource reservation is a fundamental mechanism for ensuring quality of service in time-sensitive networks. In the Ethernet technology Time-Sensitive Networking, this can be done decentrally through new resource reservation protocols. For reservation, the standards assume a maximum worst-case latency that is limited at each hop. However, we show that these worst-case latency bounds are not safe. To address this, we propose an alternative to the current standards to allow reservation of time-sensitive traffic with reliable latency guarantees, using Network Calculus. The talk is based on a publication for the Credit-Based Shaper forwarding mechanism, but extends that work to generalise the solution for different forwarding mechanisms and to determine the forwarding parameters, such as the reserved bandwidth.

3.10 Statistical Age-of-Information Bounds for Parallel Systems

Mahsa Noroozi (Leibniz Universität Hannover, DE)

Joint work of Markus Fidler, Jaya Prakash Champati, Joerg Widmer, Mahsa Noroozi
Main reference Markus Fidler, Jaya Prakash Champati, Joerg Widmer, Mahsa Noroozi: "Statistical Age-of-Information Bounds for Parallel Systems: When Do Independent Channels Make a Difference?", IEEE J. Sel. Areas Inf. Theory, Vol. 4, pp. 591–606, 2023.
URL https://doi.org/10.1109/JSAIT.2023.3328766

This work contributes tail bounds of the age-of-information of a general class of parallel systems and explores their potential. Parallel systems arise in relevant cases, such as in multi-band mobile networks, multi-technology wireless access, or multi-path protocols, just to name a few. Typically, control over each communication channel is limited, and random service outages and congestion cause buffering that impairs the age-of- information. The parallel use of independent channels promises a remedy, since outages on one channel may be compensated for by another. Surprisingly, for the well-known case of M|M|1 queues we find the opposite: pooling capacity in one channel performs better than a parallel system with the same total capacity. A generalization is not possible since there are no solutions for other types of parallel queues at hand. In this work, we prove a dual representation of age-of-information in min-plus algebra that connects to queueing models known from the theory of effective bandwidth/capacity and stochastic network calculus. Exploiting these methods, we derive tail bounds of the age-of-information of G|G|1 queues. Tail bounds of the age-of-information of independent parallel queues follow readily. In addition to parallel classical queues, we investigate Markov channels where, depending on the memory of the channel, we show the true advantage of parallel systems. We continue to investigate this new finding and provide insight into when capacity should be pooled in one channel or when independent parallel channels perform better. We complement our analysis with simulation results and evaluate different update policies, scheduling policies, and the use of heterogeneous channels that are most relevant for the latest multi-band networks.

3.11 Equivalent versions of Total Flow Analysis and SAIHU tool

Stéphan Plassart (University Savoie Mont Blanc – Annecy, FR)

Joint work of Stéphan Plassart, Jean-Yves Le Boudec, Chun-Tso Tsai, Seyed Mohammadhossein Tabatabaee
Main reference Stéphan Plassart, Jean-Yves Le Boudec: "Equivalent Versions of Total Flow Analysis", CoRR, Vol. abs/2111.01827, 2021.
URL https://arxiv.org/abs/2111.01827
Main reference Chun-Tso Tsai, Seyed Mohammadhossein Tabatabaee, Stéphan Plassart, Jean-Yves Le Boudec: "Saihu: A Common Interface of Worst-Case Delay Analysis Tools for Time-Sensitive Networks", CoRR, Vol. abs/2303.14565, 2023.
URL https://doi.org/10.48550/ARXIV.2303.14565

Total Flow Analysis (TFA) is a method for conducting the worst-case analysis of time sensitive networks without cyclic dependencies. In networks with cyclic dependencies, Fixed-Point TFA introduces artificial cuts, analyses the resulting cycle-free network with TFA, and iterates. If it converges, it does provide valid performance bounds. In the first part of this talk, I show that the choice of the specific cuts used by Fixed-Point TFA does not affect its convergence nor the obtained performance bounds, and that it can be replaced by an alternative algorithm that does not use any cut at all, while still applying to cyclic dependencies. In the second part, I present Saihu, a common python interface for worst-case delay analysis. Currently it integrates the 3 most used worst-case network analysis tools : xTFA, DiscoDNC, and Panco. Saihu provides a general interface that enables defining the networks in a single XML or JSON file and executing all tools simultaneously without any adjustment for individual tools. Saihu also exports analysis results into formatted reports automatically. Saihu is modular, allowing other worst-case analysis tools to be integrated in the future.

3.12 Time-stationary and Event- based Age-of-Information: A Palm calculus bridge

Amr Rizk (Universität Duisburg-Essen, DE)

Joint work of Amr Rizk, Jean-Yves Le Boudec
Main reference Amr Rizk, Jean-Yves Le Boudec: "A Palm Calculus Approach to the Distribution of the Age of Information", IEEE Trans. Inf. Theory, Vol. 69(12), pp. 8097–8110, 2023.
URL https://doi.org/10.1109/TIT.2023.3326381

A key metric to express the timeliness of status updates in latency-sensitive networked systems is the age of information (AoI), i.e., the time elapsed since the generation of the last received informative status message. State-of-the-art approaches to analyzing the AoI rely on queueing models that are composed of one or many queuing systems endowed with service order, e.g., FIFO, LIFO, or last-generated-first-out order. A major difficulty arising in these analysis methods is capturing the AoI under message reordering when the delivery is non-preemptive and non-FIFO, i.e., when messages can overtake each other and the reception of informative messages may obsolete some messages that are underway.

This talk which is based on our work [1] describes the derivation of computable exact formulations for the distribution of AoI in non-preemptive, non-FIFO systems where the main ingredients of our analysis are Palm calculus and time inversion.

References

1 Amr Rizk and Jean-Yves Le Boudec. *Palm Calculus Approach to the Distribution of the Age of Information*. In IEEE Transactions on Information Theory, vol. 69, no. 12, pp. 8097-8110, 2023, doi: 10.1109/TIT.2023.3326381.

3.13 Logic-Based Formal Analysis of Network Performance

Mina Tahmasbi Arashloo (University of Waterloo, CA)

In recent years, there has been a trend toward using programs written in high-level domain-specific languages to specify network packet processing behavior. This has opened up possibilities to use program analysis techniques to formally reason about network properties such as reachability and absence of forwarding loops. Most of the existing work in this area focuses on reasoning about the functional correctness of computer networks. In this talk, we present our recent efforts in using logic-based formal analysis techniques to reason about network performance properties such as throughput, latency, starvation, and fairness, and its potential synergies with network calculus. In particular, we discuss (1) how using logic-based formal analysis can help relax some of the assumptions and simplifications one may need to make in network calculus about modeling the arrival pattern of traffic, the packet processing logic, and packet loss, and (2) how network-calculus-like abstractions can help reduce the significant computational overhead that logic-based formal analysis of performance properties is prone to in comparisons with functional-correctness properties.

3.14 Factors Limiting the Modularity of xTFA

Ludovic Thomas (CNRS – Villers-lès-Nancy, FR)

Joint work of Ludovic Thomas, Ahlem Mifdaoui, Jean-Yves Le Boudec

Experimental modular Total-Flow Analysis (xTFA) is an open-source experimental Python tool for computing end-to-end latencies in time-sensitive networks. Through a modular conception based on computational blocks and flow-state partitions, xTFA supports a wide variety of models, including redundancy functions, interleaved regulators, cyclic dependencies, and clock non-idealities. But the modularity of the tool is limited because of the assumptions that the different models make and the variety of these assumptions. This talk discusses the origin of these limitations and proposes open question towards a classification of assumptions for service-curves properties in time-sensitive networks.

3.15 Performance and Scaling of Parallel Systems with Blocking Start and/or Departure Barriers

Brenton Walker (Leibniz Universität Hannover, DE)

Parallel systems divide jobs into smaller tasks that can be serviced by many workers at the same time. Some parallel systems have blocking barriers that require all of their tasks to start and/or depart in unison. This is true of many parallel Machine Learning workloads, and popular Apache Spark engine has, for this reason, recently added support for Barrier Execution Mode (BEM).

The drawback of these barriers is reduced performance and stability, compared to equivalent non-blocking systems. We derived analytical expressions for the stability for BEM systems and extend results from Network Calculus to derive waiting and sojourn time bounds.

Our results show that for a given utilization (ρ) and number of workers (s), there is an optimal degree of parallelism (k) that balances waiting time and execution time to minimize sojourn times. This complements prior work on so called ”tiny tasks”, where $\frac{k}{s} >> 1$, to show that among systems with barriers, the unstable Split-Merge model ($k = s$) is the anomaly, and taking $\frac{k}{s} < 1$ or $\frac{k}{s} > 1$ is actually much better.

3.16 Improving algorithms and software for DNC

Raffaele Zippo (University of Pisa, IT)

Joint work of Raffaele Zippo, Paul Nikolaus, Giovanni Stea, Andrea Trasacco
Main reference Raffaele Zippo, Giovanni Stea: "Nancy: An efficient parallel Network Calculus library", SoftwareX, Vol. 19, p. 101178, 2022.
URL https://doi.org/10.1016/J.SOFTX.2022.101178
Main reference Raffaele Zippo, Paul Nikolaus, Giovanni Stea: "Extending the network calculus algorithmic toolbox for ultimately pseudo-periodic functions: pseudo-inverse and composition", Discret. Event Dyn. Syst., Vol. 33(3), pp. 181–219, 2023.
URL https://doi.org/10.1007/S10626-022-00373-5
Main reference Raffaele Zippo, Paul Nikolaus, Giovanni Stea: "Isospeed: Improving (min, +) Convolution by Exploiting (min, +)/(max, +) Isomorphism", in Proc. of the 35th Euromicro Conference on Real-Time Systems, ECRTS 2023, July 11-14, 2023, Vienna, Austria, LIPIcs, Vol. 262, pp. 12:1–12:24, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.
URL https://doi.org/10.4230/LIPICS.ECRTS.2023.12

In this talk, I present results and work in progress improving the existing software and algorithms for DNC computations, particularly those built on the Nancy computational library. First, I discuss the optimizations stemming from the duality between (min,+) and (max,+), which we call isospeed algorithms. Then, I present some work in progress that may be useful to the community: Nancy HTTP wrappers, to use the library from other languages; Nancy.Interactive to improve the testing and teaching workflow; Nancy.Expressions, opening up new ways to optimize computations transparently to the user; and DNC Benchmarking, a project aimed at benchmarking DNC implementations to highlight the impact of optimizations on runtime. Finally, a note on the need for practical examples to highlight, especially to those outside our community, where the hard computations are in DNC, and thus improve the chances of publication of this kind of results.

4 Working groups

4.1 Network Calculus and Machine Learning/Artificial Intelligence

Michael Beck (University of Winnipeg, CA)

Our group discussed three areas in which machine learning and network calculus overlap, all of them related to hard optimization problems. These are i) optimal transformation of flows to allow the analysis of feedforward networks with deterministic network calculus (DNC); ii) optimal cutting of flows to break cyclic dependencies in arbitrary networks; iii) determining the schedule in IEEE 802.1Qbv time-aware schedulers.

4.1.1 Optimal topology manipulation in feedforward networks

To apply the benefits of a Pay-Multiplexing-Only-Once (PMOO) analysis in DNC the corresponding network topology must be a nested feed-forward network. This means that the service elements in the network can be ordered $1, \ldots, n$ such that i) for each flow all nodes it visits are traversed in increasing order and ii) for any two flows F_1, F_2 we have either

$$s_{F_1} > e_{F_2} \text{ or } e_{F_1} < s_{F_2} \tag{1}$$

or

$$\mathcal{N}(F_1) \subseteq \mathcal{N}(F_2) \text{ or } \mathcal{N}(F_1) \supseteq \mathcal{N}(F_2), \tag{2}$$

where s_F, e_F, and $\mathcal{N}(F)$ denote the first, last, and all nodes a flow F traverses respectively.

In networks that have interleaving flows instead, i.e. are not nested, the PMOO analysis is not directly applicable. Since PMOO leads to generally better bounds, it is thus desirable to modify the topology, this is done by "cutting" one or more flows. For example, consider a three node network with the two flows $\mathcal{N}(F_1) = \{1, 2\}$ and $\mathcal{N}(F_2) = \{2, 3\}$. In this case it is possible to consider flow F_1 leaving the network after traversal of node 1 and re-entering it at node 2, i.e., an output bound of flow F_1 is computed after being served at node 1 and this output bound serves as a new arrival bound for a flow F_1' with $\mathcal{N}(F_1') = \{2\}$. Alternatively, flow F_2 could be cut after traversing node 2. The resulting performance bounds are generally not the same leaving a decision problem for the practitioner, which does not have an obvious analytical solution even for modest network sizes.

An alternative to cutting flows is to prolong flows to create a nested topology. In the above example we could also extend either of the two flows to traverse all three nodes leading to two more possible topologies, which are nested and serve as an upper bound for the original network with respect to calculable performance guarantees. In more complex networks several steps of flow prolongations and flow cuttings will have to be performed to reach a nested topology.

Finding an optimal sequence for these steps in general feed-forward networks is a problem of high complexity. To find such a sequence a machine learning approach can be applied, which is based on graph neural networks (GNN). For this methodology to be applied the network must be represented as a graph which represents possible cuts of flows (a cutting graph) or prolongation of flows (a prolongation graph). Creating a single graph that captures both, cuts and prolongations, has – to the authors knowledge – not been reported, yet.

Prolongation and cutting graphs are very similar to each other, thus for brevity we describe here only cutting graphs. They consists of four types of nodes: i) server nodes, ii) flow nodes, iii) ordering nodes, and iv) decision nodes. Together with edges and node labels the cutting graph is a one-to-one mapping of a network. The cutting graph is created as follows:

- Each server of the network is represented as a server node with a label that represents the server's service guarantee. The server nodes are connected via edges in tandem according to their natural ordering $1, \ldots, n$ (see the first condition of feedforward topologies at the beginning of this subsection).
- Each flow of the network is represented as a flow node with a label that represents the flow's (upper) arrival guarantee.
- For each flow a number of ordering nodes, equal to the number of servers the flow traverses, is added to the graph. The ordering nodes are labeled by the hop count of the flow at each server and are connected via edges to the flow node and the respective server node.
- For each flow and each consecutive pair of ordering nodes a decision node is added to the graph and connected with three edges to the pair of ordering nodes and the respective flow node. These nodes are initially unlabeled.

For a visualization of a network and its corresponding cutting graph see [1].

Having such a cutting graph at hand a GNN can be used to determine labels to the ordering nodes. These labels will be binary and represent the decision to perform the cut or not. To train the GNN a supervised learning approach has been followed in [1] that requires training data for which the optimal solution is already known. Due to the complexity of the problem the such generated training data only contains graphs of relatively small size for which an optimal solution can still be obtained. Our group discussed the idea whether a model trained on such data can generalize successfully to networks of larger size. This research question holds for both cutting graphs and prolongation graphs.

Since, optimal solutions for larger topologies are out of reach the problem could also be approached by unsupervised machine learning methods instead. The intermediate obtained performance bound will serve as feedback to the unsupervised methodology, for example, as a fitness function in a genetic algorithm.

4.1.2 Optimal cutting in cyclic networks

Very related to the above is the question of which cuts to perform to break cyclic flow dependencies in non-feedforward networks. In such a network there exists at least one cycle of servers which can be traversed just by following the direction of one or more flows. Such cycles prevent the direct application of the PMOO analysis. Which flows to cut at which positions to achieve optimal performance guarantees is not obvious. The application of supervised or unsupervised machine learning methods similar to those described above should be investigated.

4.1.3 Time-Aware Scheduling in Time Sensitive Networks

The last topic discussed by our group was how to determine an optimal schedule for IEEE 802.1Qbv schedulers. More precisely, given

- an overall service description for the service element,
- arrival curves for each flow (each flow enters one of the 8 queues, with more than one flow per queue possible),
- the length of the cyclic schedule, and
- prioritization between queues,

how should the queue's gates be opened and closed to meet the delay requirements for each of the 8 traffic classes. To note here is, that the opposite problem of deriving delay bounds from a given schedule is relatively easy.

Since the schedule can be arbitrarily structured we face another hard optimization problem. We see an opportunity to deploy machine learning methods for synthesizing a schedule according to the delay objectives. An unsupervised methodology seems appropriate here, with a feedback that is based on the margins by which the delay requirements are (not) met. Due to the continuous introduction and ceasing of flows the gate schedules must be created online, which in turn, puts a limit on the complexity of the models that can be used.

References

1 Fabien Geyer, Steffen Bondorf, *Graph-based deep learning for fast and tight network calculus analyses*, IEEE Transactions on Network Science and Engineering, 8, 1, p. 75-88. 2020

4.2 Network Calculus versus other types of real-time network analysis

Anne Bouillard (Huawei Technologies – Boulogne-Billancourt, FR) and Jörg Liebeherr (University of Toronto, CA)

Recent studies have highlighted similarities between different analyses approaches for real-time systems:

- The equivalence between Real-Time Calculus (RTC) and the variable capacity node in Network Calculus (NC) was established over a decade ago.
- More recent research identifies similarities with Response Time Analysis (RTA) and even stream models.

Despite these similarities, significant differences in the approaches remain. Investigating the potential of NC formalism could be beneficial.

4.2.1 Comparative Analysis: Other Methods vs. Network Calculus

- NC does not support multicore analysis.
- Packet losses are seldom considered in NC (there are only a few works about it)
- NC lacks Directed Acyclic Graph (DAG) tasks to represent task patterns.
- NC also does not consider behaviors depending on a state, that could be represented by an automaton

4.2.2 Focus on RTC and NC Comparison

The similarities have been established quite long ado, since RTC is a variation of NC and has been defined to be applied to real-time systems, whereas NC was previously targeting communication systems. The primary difference lies in the time domain (non-negative reals in NC vs. the whole real number in RTC), and this makes a difference, especially when considering lower arrival curves (which is doable, but not often done in NC).

Another difference in the approach rather than in the theory is that NC will abstract the model by simpler curves (a few token-buckets), here RTC will consider stair-case curves (based on the packet level).

Some generalization have also been considered, like interleaved of packets of different type and differentiated service among them.

We also pointed a difference in the approach: while NC is analysis a network when the scheduling parameters are fixed (allowing modular or global approaches), the RTC is modular and in order not to be too pessimistic in the composition, it chooses the scheduler.

4.2.3 A Common Open Issue: Back Pressure

Back-pressure model analysis remains unresolved. Some have looked at the model (for NoC or wormhole routing analysis), and failed to have an accurate analysis. Further theoretical developments may provide solutions.

4.2.4 Takeaway

It might be an opportune time to draft a review paper on *20 Years of RTC*.

4.2.5 List of Acronyms:

- DAG: Directed Acyclic Graph
- NC: Network Calculus
- NoC: Network on Chip
- RTA: Response Time Analysis
- RTC: Real-Time Calculus

4.3 Core of the Network Calculus Theory

Marc Boyer (ONERA – Toulouse, FR)

The group discussed different versions of service, which offered different points of view (Adaptive service, Real-Time Calculus – RTC, strict service, etc.).

Some discussions also on the equivalence between (min,+) and (max,+) dioids. Rather than using them one at a time, one can think about (min-plus) and (max-plus) at the same time. Starting points can be found in the 1992 book by Baccelli/Cohen/Olsder/Quadrat [1]. Since convolution is a Minkowski sum, an exploration of algebras of Minkowski systems may lead to progress. The Moreau conjugacy a generalisation of legendre transform, has also been mentioned.

We also speak about the question of rare perturbation. The deterministic network calculus is designed to bound the worst case, but not to quantify how rare/frequent are this worst cases. Considering a sequence of message, all having a delay of 1ms except one out of 1000 that have a delay of 10ms. Then network calculus is not able to make the difference with a sequence with all delays being 10ms. And this is not related to the bounding methods, even the definition of delay is not able to capture it.

Additional, short discussion have been done on how to model real systems with notion of mode and mode change and on interesting non-linear systems.

References

1 F. Baccelli, G. Cohen, G.J. Olsder, J.P. Quadrat. *Synchronization and Linearity: An algebra for discrete event systems*. ISBN: 047193609X. John Wiley & Sons Ltd, 1992

4.4 Network Calculus Tools

Lisa Maile (Universität Erlangen-Nürnberg, DE) and Anja Hamscher (RPTU – Kaiserslautern, DE)

4.4.1 Introduction

This section outlines the discussion held on NC tools and their functionalities, mainly “Nancy” and “Saihu”.

4.4.2 New "Nancy" Functionality

Nancy [2] is a versatile C# library specifically designed for Network Calculus computations. It implements min-plus and max-plus operators and can manage ultimately pseudo-periodic piecewise affine curves. Nancy is released under the MIT license and developed by Raffaele Zippo and Giovanni Stea from the University of Pisa, Italy.

The new functionality of Nancy includes the ability to embed Nancy in other languages through HTTP+JSON. During the discussion, a live demonstration showcased how this integration works. Feedback on the current implementation addressed several points:

- Compatibility is not limited to Windows.
- Integration into existing projects is a key motivation for this development.
- Nancy utilizes Plotly for plotting, which can be reimplemented as needed.
- The need for garbage collection and storage management for large networks was highlighted as an area requiring further investigation.

4.4.3 Presentation of "Saihu"

Saihu [1] is a Python interface that integrates several major worst-case network analysis tools, including the tools xTFA (supports TFA), DiscoDNC (supports LUDB, PMOO, SFA), and Panco (supports PLP and ELP). These tools are frequently used in time-sensitive network analysis. Saihu provides a general interface that allows users to define networks in a single XML or JSON file and execute all tools simultaneously without needing to adjust individual tools.

Additionally, Saihu exports analysis results into formatted reports automatically and offers automatic network generation significantly reducing the required work of users. The modular nature of the package allows for the incorporation of more tools in the future. During the discussion, the following points were emphasized:

- Extension for new tools can be achieved using a common JSON format.
- The syntax of JSON files needs to be compatible with various tools to ensure smooth integration.

4.4.4 Further Aspects

The user experience of the tools was discussed, focusing on ease of use, integration capabilities, and the overall satisfaction of users with the functionalities provided by both Nancy and Saihu.

A comprehensive list of tools is available on Wikipedia, which serves as a valuable resource for users looking to explore different tools available for network analysis and other related tasks.

One of the issues discussed was the challenge of allocating time and resources for tool development in an academic setting. The results of programming efforts alone are often difficult to publish, highlighting a significant hurdle for researchers and developers in academia.

The development of a benchmark was proposed as a beneficial step to compare different tools effectively. Such benchmarks would provide standardized criteria for evaluating tool performance and capabilities, facilitating better decision-making for users and developers.

4.4.5 Conclusion

The discussion provided valuable insights into the functionalities and future directions for tools like Nancy and Saihu. Addressing user feedback, compatibility, and resource challenges will be crucial for the continued development and adoption of these tools in various applications.

References

1 Tsai, Chun-Tso et al. "Saihu: A Common Interface of Worst-Case Delay Analysis Tools for Time-Sensitive Networks." ArXiv abs/2303.14565 (2023).

2 R. Zippo and G. Stea, "Nancy: An efficient parallel network calculus library", SoftwareX, vol. 19, Jul. 2022.

Participants

- Michael Beck
University of Winnipeg, CA
- Steffen Bondorf
Ruhr-Universität Bochum, DE
- Anne Bouillard
Huawei Technologies –
Boulogne-Billancourt, FR
- Marc Boyer
ONERA – Toulouse, FR
- Peter Buchholz
TU Dortmund, DE
- Almut Burchard
University of Toronto, CA
- Georg Carle
TU München – Garching, DE
- Samarjit Chakraborty
University of North Carolina at
Chapel Hill, US
- Vlad-Cristian Constantin
RPTU – Kaiserslautern, DE
- Hugo Daigmorte
RealTime-at-Work – Nancy, FR
- Markus Fidler
Leibniz Universität
Hannover, DE
- Anaïs Finzi
TTTech Computertechnik –
Wien, AT
- Stéphane Gaubert
INRIA & CMAP, Ecole
polytechnique – Palaiseau, FR
- Damien Guidolin–Pina
RealTime-at-Work – Nancy, FR
- Anja Hamscher
RPTU – Kaiserslautern, DE
- Max Helm
TU München – Garching, DE
- Kai-Steffen Jens Hielscher
Universität Erlangen-
Nürnberg, DE
- Yuming Jiang
NTNU – Trondheim, NO
- Kai Lampka
NXP Semiconductors –
München, DE
- Jean-Yves Le Boudec
Jouxtens-Mézery, CH
- Harvinder Lehal
University of Toronto, CA
- Jörg Liebeherr
University of Toronto, CA
- Lisa Maile
Universität Erlangen-
Nürnberg, DE
- Mahsa Noroozi
Leibniz Universität
Hannover, DE
- Xi Peng
Huawei Technologies –
Hong Kong, HK
- Stéphan Plassart
University Savoie Mont Blanc –
Annecy, FR
- Amr Rizk
Universität Duisburg-Essen, DE
- Giovanni Stea
University of Pisa, IT
- Mina Tahmasbi Arashloo
University of Waterloo, CA
- Ludovic Thomas
CNRS – Villers-lès-Nancy, FR
- Brenton Walker
Leibniz Universität
Hannover, DE
- Kui Wu
University of Victoria, CA
- Raffaele Zippo
University of Pisa, IT

Report from Dagstuhl Seminar 24151

Methods and Tools for the Engineering and Assurance of Safe Autonomous Systems

Elena Troubitsyna[*1], **Ignacio J. Alvarez**[*2], **Philip Koopman**[*3], **and Mario Trapp**[*4]

1 **KTH Royal Institute of Technology – Stockholm, SE.** elenatro@kth.se
2 **Intel – Hillsboro, US.** ignacio.j.alvarez@intel.com
3 **Carnegie Mellon University – Pittsburgh, US.** koopman.cmu@gmail.com
4 **TU München, DE.** mario.trapp@cit.tum.de

Abstract

Autonomous systems rely increasingly on Artificial Intelligence (AI) and Machine Learning (ML) for implementing safety-critical functions. It is widely accepted that the use of AI/ML is disruptive for safety engineering methods and practices. Hence, the problem of safe AI for autonomous systems has received a significant amount of research and industrial attention over the last few years. Over the past decade, multiple approaches and divergent philosophies have appeared in the safety and ML communities. However, real-world events have clearly demonstrated that the safety assurance problem cannot be resolved solely by improving the performance of ML algorithms. Hence, the research communities need to consolidate their efforts in creating methods and tools that enable a holistic approach to safety of autonomous systems. This motivated the topic of our Dagstuhl Seminar – exploring the problem of engineering and safety assurance of autonomous systems from an interdisciplinary perspective. As a result, the discussions of achievements and challenges spanned over a broad range of technological, organizational, ethical and legal topics summarized in this document.

Seminar April 7–12, 2024 – https://www.dagstuhl.de/24151

2012 ACM Subject Classification Computing methodologies → Artificial intelligence; Computer systems organization → Dependable and fault-tolerant systems and networks; Computer systems organization → Embedded systems; Hardware → Safety critical systems; Software and its engineering

Keywords and phrases ai, safety assurance, safety-critical autonomous systems, simulation-based verification and validation, software engineering

Digital Object Identifier 10.4230/DagRep.14.4.23

[*] Editor / Organizer

Methods and Tools for the Engineering and Assurance of Safe Autonomous Systems, *Dagstuhl Reports*, Vol. 14, Issue 4, pp. 23–41
Editors: Ignacio J. Alvarez, Philip Koopman, Mario Trapp, and Elena Troubitsyna
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Elena Troubitsyna (KTH Royal Institute of Technology – Stockholm, SE)
Ignacio J. Alvarez (Intel – Hillsboro, US)
Philip Koopman (Carnegie Mellon University – Pittsburgh, US)
Mario Trapp (TU München, DE)

Introduction

The examples of modern autonomous systems include self-driving cars, UAV (drones), underwater vehicles, various industrial and home service robots. In general, autonomous systems are intended to operate without human intervention over prolonged time periods, perceive their operating environment and adapt to internal and external changes.

For example, a self-driving car gathers information from camera and lidar to detect, e.g., pedestrians on the road and plan collision avoidance maneuvers, slowing down or breaking, i.e., avoid hazards. The perception functions process the inputs of various sensors and generate the internal model of the operating environment. By relying on this model, the decision functions plan and execute the actions required to achieve the goals of the mission. In general, they follow the generic "sense-understand-decide-act" behavioral pattern, which is also traditionally adopted in robotics.

Both sensing and decision making usually rely on Artificial Intelligence (AI), in particular Machine Learning (ML). While AI and ML algorithms have already been used in robotics for several decades, their use in safety-critical systems is fairly new and currently not appropriately addressed by safety engineering neither from technological, nor from organizational and legal points of view.

The problem of safe AI has received a significant amount of research and industrial attention over the last few years, but there has been a divergence in the approaches taken by the safety and the ML communities. Moreover, it has become clear that the safety assurance problems cannot be resolved by improving the ML algorithms alone. Hence, the research communities should consolidate their efforts in creating methods and tools enabling a holistic approach to safety of autonomous systems.

This motivated the topic of our Dagstuhl Seminar – exploring the problem of engineering and assuring safety of autonomous systems from an interdisciplinary perspective. A group of experts from avionics, automotive, machine learning, simulation, verification and validation and safety engineering reviewed the current academic state-of-the-art, industry practices and standardization to determine the latest achievement and challenges in developing and safety assurance for autonomous systems over a broad range of technological, organizational, ethical and legal perspectives.

As a result, the discussions of achievements and challenges in developing and assuring safety of autonomous systems spanned over a broad range of technological, organizational, ethical and legal topics.

Organisation of the seminar

The seminar brought together researchers and practitioners from different disciplines and application domains. Since, currently, the innovation in autonomous systems is strongly led by industry, a significant number of participants were industrial engineers, who not only shared their best practices but also identified unsolved research problems. In constructive debates, we discussed the results of applying and experimenting with various techniques for engineering safe autonomous systems and identified open research challenges.

To facilitate an open discussion between the participants, and analyze the problem of engineering safe autonomous systems from different points of view, before the seminar, we identified the following general discussion themes:

- Role of formal methods in engineering and assurance of safe autonomous systems
- Regulatory, assurance and standards for safety-critical autonomous systems
- Safety of AI-based system versus normal technical system safety
- Safety and security interactions
- Risk acceptance for autonomous systems

This report presents the summaries of the discussions focused on the specific topics within these themes.

We would like to acknowledge the supporting contributors – the session chairs and scribes that helped to collect the information for this report: Magnus Albert (SICK AG – Waldkirch, DE), Ensar Becic (National Transportation Savety Board, US), Nicolas Becker (Stellantis France – Poissy, FR), Simon Burton (Gerlingen, DE), Radu Calinescu (University of York, GB), Betty H. C. Cheng (Michigan State University – East Lansing, US), Krzysztof Czarnecki (University of Waterloo, CA), Niels De Boer (Nanyang TU – Singapore, SG), Lydia Gauerhof (Bosch Center for AI – Renningen, DE), Jérémie Guiochet (LAAS – Toulouse, FR), Hans Hansson (Mälardalen University – Västerås, SE), Aaron Kane (Edge Case Research – Pittsburgh, US), Lars Kunze (University of Oxford, GB), Jonas Nilsson (NVIDIA Corp. – Santa Clara, US), Nick Reed (Reed Mobility – Wokingham, GB), Jan Reich (Fraunhofer IESE – Kaiserslautern, DE), Martin Rothfelder (Siemens – München, DE), Philippa Ryan (University of York, GB), Fredrik Sandblom (Zenseact AB – Gothenburg, SE), Stefano Tonetta (Bruno Kessler Foundation – Trento, IT), Kim Wasson (Joby Aviation – Santa Cruz, US), and William H. Widen (University of Miami – Coral Gables, US). In the spirit of Chatham House Rules that prevailed in the meeting, we are not attributing any particular written text to any particular person.

2 Table of Contents

3 Role of Formal Methods in Engineering and Assurance of Safe Autonomous Systems

3.1 Formal methods for AI-enabled systems

Model-driven and formal approaches are often used in the development of safety-critical systems. Hence, it is natural to discuss the perspectives of using formal methods in engineering safe autonomous systems. We observed that traditional formal methods, such as model checking, theorem proving, and static analysis provide not an "absolute" but rather a context-dependent proof of correctness and system safety. Moreover, they typically model certain core system functionality related to safety, i.e., do not aim at specifying the entire system behaviour. An open challenge is to understand the context surrounding the formal model and systematically represent it. Hence, an important research direction is to develop the mechanisms of safe context-aware incorporation of partial specifications. They should contribute to handling the complexity of autonomous systems.

There are different views on the goal of applying formal methods for engineering AI-enabled systems. It is commonly agreed that the scope of formal methods should be broadened to adapt them to systems engineering. We identified two most likely roles that formal methods will play in designing safe autonomy. On the one hand, formal models can be used to improve robustness of AI components, e.g., assess quality of data sets and facilitate generation of synthetical data. On the other hand, they can treat an AI component as a black box and focus on modeling system-level safety properties under uncertainty. To achieve the latter, the advances in probabilistic reasoning are required to express robustness of an AI component as a part of the entire system specification. We commonly agreed that formal methods do not seem to be a good fit for the high-dimensionality problems, i.e., it is unlikely that formal modeling of neural networks would become feasible. However, we already have had successful experiments with formal verification of decision making and path planning components. Hence, we concluded that formal methods can facilitate improving safety of the overall autonomous system by rigorously modeling monitors and specifying safety wrappers.

An important aspect to be considered is how to validate a formal model, i.e., ensure its appropriateness. In the context of ML-based components, we need to understand how to bridge a reality gap. Typically, a formal model describes the requirements to be fulfilled by a component semantically, while Deep Neural Network (DNN) recognises objects as a collection of pixels. Establishing a mapping between semantics of safety requirements and their actual implementation is an open challenge. Hence, further research is required to understand how to efficiently decompose system-level safety properties into properties to be fulfilled by neural network architecture and data sets.

An important aspect in formal modeling of autonomous systems is representing an open unconstrained operational environment. Since AI algorithms are data-driven and make decisions even in the situations that cannot be foreseen, verifying safety of such decisions currently constitutes an unsolved challenge. We concluded that it could be addressed by generation of a formal model based on the data about system behavior and then verifying its safety, i.e., using formal modeling for re-engineering and verification rather than verification-driven engineering of AI-enabled systems. Overall, we believe that formal methods will play a significant role in safety assurance although their use would be adapted and broadened to incorporate reasoning about system safety that is enabled by AI components.

3.2 Static and dynamic analysis of safety cases

A safety case presents and communicates risk as well as provides analysis. Hence, it plays an important role in the design and certification of safety-critical systems. It is typically used to support risk-informed decision making processes. Safety cases bring a vast amount of work products related to the design and development choices together into a cohesive narrative. Safety management through the lifecycle of an autonomous system is highly dynamic, with changing environments and continuous system improvement occurring through development and operation. In order to properly present this dynamic safety argument, safety cases for autonomous systems must also be managed in a dynamic way.

Dynamic updates to a safety case come in many forms. Safety performance evidence from field data and other ongoing analysis is updated throughout development and during system operation. Additionally, technical analysis and processes may be updated as part of continuous improvement of the system or expanding the operational domain or capabilities.

Dynamic analysis of safety cases is desired to manage efficient evaluation of safety cases in the face of constantly evolving safety evidence and argument. Automated and dynamic evaluation of safety cases can provide many benefits, including:

- Keeping pace with the quick release cycles of modern autonomous systems;
- Provide up-to-date status reporting of the safety assurance argument;
- Provide impact analysis capabilities to manage changes and incremental assurance;
- Minimize the cost and effort required by traditionally slow and highly manual evaluation.

A primary challenge in building methods and tools to support the dynamic evaluation of safety cases is that the structure and usage of safety cases is not standardized. Different domains, different uses (e.g., internal assessment, certification, and messaging to external stakeholders), and relevant stakeholders affect the desired format and needs of a safety case. There is a clear divergence between research focused on using argument structures as models for evaluation and computation and other work which treats the safety case primarily as a human communication tool. Research towards evaluation over the safety argument structure itself is often based on Goal Structuring Notation and other graphical argument notations, adding semantics for evaluation and analysis directly to the argument structure. From the "safety case as communications" perspective, these analyses are better treated as separate analysis concepts, which can be integrated into the safety case as evidence where needed to support communicating the desired properties.

The needs of safety case tooling and analysis methods heavily depend on the usage. We identified two main challenges in dynamic safety case evaluation: defining and aligning the industry with consistent safety-case related terminology and processes to support the value of common methods and tools and navigating the scope overlap between safety case analysis and other related safety and program-management based analysis.

4 Regulatory, assurance and standards for safety-critical autonomous systems

4.1 Safety metrics for ML-based functionality: requirements engineering aspect

The use of ML-based components to implement safety-critical functions differentiates autonomous safety-critical systems from "nominal" safety-critical systems. However, measuring the safety of ML components is notoriously difficult. First and foremost, safety is a system property, not an ML component property. Hence, defining "safety metric for ML" is a contradiction per se. Thus, it is more appropriate to develop performance metrics for ML components. Correspondingly, there are two challenges associated with this problem: decomposition/composition of safety metrics and their interpretation. To address the former, a systematic methodology is needed to decompose and compose system safety metrics into ML component performance metrics and vice versa. This involves deriving a set of necessary and sufficient performance level requirements and metrics from the system level requirements and metrics. It also means that individual performance-level metrics must be composed into system-level metrics. It requires developing techniques enabling inference of system's safety from lower level performance metrics.

To address the latter, a comprehensive framework of safety metrics should be established. The metrics must be both measurable and interpretable, especially given the complex and sometimes unpredictable nature of ML systems. The metrics should be capable of measuring the completeness and appropriateness of requirements decomposition as well as assessing the satisfaction of specific performance requirements.

An open research challenge is developing mechanisms enabling a combination of black-box metrics, which reflect a component's or system's behavior without knowledge of its internal realization, and white-box metrics, which reflect the specific internal realization of a component or system.

We also underscored the need to address the problem of measuring residual uncertainties and how they can be handled by runtime monitoring. This allows for real-time adjustments and improvements, enhancing the safety and performance of the ML system.

A key point is to create a common understanding among different stakeholders that metrics should be assigned to requirements rather than components. We believe that this shift in focus from general components' properties to specific performance requirements is crucial.

Overall, while there are many challenges in defining and measuring safety metrics for ML and no universal solution. It is clear that these obstacles can be overcome with a systematic, engineering-focused approach and an emphasis on continuous monitoring and improvement.

4.2 Safety metrics for ML-based functionality: from AI-component to system-level metrics

The problem of measuring safety of ML-based components is currently considered to be a central one for assuring safety of autonomous systems. Hence, it was discussed in two parallel sessions. Though the sessions focused on different aspects, they had a common starting point – an acknowledgement that safety is a system-level property, which is not straightforward to decompose into component-level metrics. As a result it poses a question: what are the differentiating factors between leading and lagging safety metrics while defining safety at the component versus the system level. This highlighted the industry's inclination to treat AI training data as a competitive edge, complicating the evaluation of AI model performance and necessitating traceability in metrics.

Currently, there is no consolidated view on the problem of safety metrics of system architecture, which is correlated to the problem of modeling probabilistic nature of AI. The automotive industry tends to make emphasis on threshold values for widely agreed functional safety indicators (FuSa – ISO 26262) and safety of the intended functionality (SOTIF – ISO 21448). However, any attempt to define safety metrics for AI-based systems is cursed with dimensionality problems. SOTIF is notoriously ambiguous in the ML safety assessment. The discussion consensus was that explainability, repeatability and testability should be prioritized in safety metrics, despite the inherent challenge of establishing causality in current ML approaches. Furthermore, there is a clear need for real-time metrics post vehicle deployment to monitor the internal AI-based systems performance and the overall vehicle safety (positive risk balance).

We also discussed the use of lagging metrics in the development of Autonomous Systems and the concept of positive risk balance, particularly within operational design domains (ODDs). The experts identified the lack of specification and consistency in ODDs in industry as a critical issue. Moreover, there is also the need to develop a common understanding of risk acceptance during ODD exits.

The most pressing problems in the field today are the definition of metrics for ML-based components of safe autonomous systems that are suitable for legal inclusion. Such metrics should enable and support regulation of the mass deployment of automated driving systems. This requires further immediate work on the relationship between ML performance metrics and overall vehicle safety, the explainability of ML metrics, and the need for cross-domain consensus on metric definitions. In-depth deliberations underscored the importance of focusing on measurable behaviors, the challenges of probing design defects in ML systems, and the societal implications of accepting AI-based solutions as potentially defective systems. The main take-away was a call to develop robust metrics that genuinely imply safety and address the complexities of translating AI-based component-level metrics to vehicle-level safety.

4.3 Explainability and undertandability

The explainability of a complex system is recognised as a critical element in establishing its trustworthiness (for a variety of audiences). It is also essential for its efficient development. The concept of explainability has different aspects including error tracing or liability assignment. It was recognised that the approach may look and feel very different if the audience for explainability is an engineer, a safety investigator, senior management, the media or the public. Similarly, responsibility for communication of system explainability may reside with different roles within an organization depending on the audience.

The approach to explainability may also differ depending on its time sensitivity and purpose. Data collection and analysis for explainability following a safety critical incident is likely to be very different to that applied for focused engineering development activities or for achieving runtime explainability. In all cases, the resources (personnel, computing etc.) devoted to explainability have to be proportionate relative to the requirements. Further work is required to define what proportionate resource allocation means for achieving explainability of an autonomous system.

The concept of "Explainability by design" was referenced as an important technique for satisfying these requirements. This implies using a system architecture that facilitates access to critical information to support the required explainability with modular or layered architectures likely to enable explainability more readily than "black box" systems.

Explainability is a rather new concept that requires further deliberation and analysis. Among the key challenges to be addressed is the question of how to balance explainability-by-design (i.e., deliberate architectural design to maximize/optimize/set explainability) with performance of the end-to-end system (faster, more efficient). We also recognised that it is worth investigating different "layers" of explainability, i.e., would it be sufficient to merely consider, e.g., "reasonable reproducibility" or an ability to identify a root cause. Each "layer" has an impact on the practical resources and time that it would be required to provide an explanation.

We also concluded that currently there is no consensus on minimum requirements for explainability in the context of its audience, purpose and time sensitivity. This work would require creating a consensus on a vocabulary for explainability that should be grounded in the common understanding of the possible deviations from the intended functionality and/or functional limitations

Finally, there is a clear need for work on integrating explainability of AI/ML into a broader category of explainability for autonomy/complex systems as well as the broader topic of safety governance and decision-making (including the safety assessment).

4.4 Bootstrapped safety approaches

The main concept behind bootstrapping is to gain increased confidence and range in the metric "number of miles without a crash". This is achieved through predicting/extrapolating future performance through live testing as follows:

- Run X miles with safety driver (without crashes)
- Assume X miles with confidence interval
- Add Y miles without safety driver
- Extend/extrapolate number of miles without a crash

We observed this approach being used in industrial settings and hence, it is worth to analyze its ethical and technical issues. On the ethical side, we had a debate about whether it was appropriate or responsible to deploy the car without a safety driver when effectively performing live-testing. Some participants argued that the car could not be sold without demonstrating that it can be used without a driver, whilst others felt this was for marketing and a safety driver should always be used when building confidence/assurance. Additionally, the participants discussed how the risk could be communicated to the public or senior management effectively.

From a technical perspective it was noted that there would be many challenges to the validity of statistical data gathered this way. To analyse the problem, we assumed that the software version was identical across all the tests (although this may not be the case in reality). Obviously, a single crash upsets the statistics and to counter this the manufacturer may argue that particular crash does not count (e.g., it's an extremely rare event or not the fault of the autonomous vehicle) or has been fixed in the next software build. It would also be difficult to aggregate different individual drives due to environmental conditions, level of traffic, confidence of the safety driver etc.

A proposed solution was to use the claims from a strong safety case as a "Bayesian prior". This would be updated through monitoring of the deployed vehicles (in all phases). In a "black box" approach, just unwanted safety-related events would be monitored (from vehicle collisions to any safety-related failure at the system level, depending on the monitoring regime) to perform Bayesian updating on the probability of the claim being correct, and hence on risk, e.g. via the conservative process. In a more "white box" approach, the updating (and/or falsification of safety claims) could affect individual subclaims (e.g. elevated rates of functional failures at the component level or violations of assumptions). It was assumed that the safety case had a static structure and that appropriate/useful Safety Performance

Indicators (SPIs) can be quantified, measured and monitored. At a sufficiently detailed level of monitoring, the developers can gather evidence that supports or counters different claims in the case more concretely. This leads to other challenges though, such as how to take into proper account, in inference from the monitoring, correlations between monitored variables, and how to aggregate confidence throughout a safety case.

A relevant topic in this respect is supplementing evidence gathered from public road operations with simulations, which has its own challenges such as performance, repeatability and fidelity to the real world. Also relevant is engineering in safety mitigations and redundancies to reduce risk further.

4.5 Managing Uncertainty

Uncertainty has complex and multi-dimensional nature in the design, operation, and assurance of software-intensive safety-critical products and systems. Several taxonomies have been proposed already. However, there is a need for the new one that would capture the aspects introduced by ML.

Complexity of managing uncertainty has increased as a result of introducing ML-based solutions for managing autonomous vehicles in open environments (aka public roads). Many projects adopt a technical perspective, but the problem is much broader, e.g., it includes assurance/organisational uncertainty. Useful existing taxonomies on uncertainty needs updating to capture this.

Autonomous systems broaden the causes of uncertainty in the design of safety-critical systems. The primary source of uncertainty is the system's operating environment. We agreed that "unknown unknowns" are always going to be unavoidable for autonomous systems deployed in the real world and a lack of knowledge may lead to an unjustified sense of confidence in assurance. Different types of uncertainties can interact in unexpected, complex ways. Managing uncertainty involves finding trade-offs between safety, performance, cost, etc.

An open issue is to find appropriate mechanisms for managing uncertainty. For example, in controlled environments, such as factories, higher level (non-AI) safety-functions could be used to reduce the safety risks of underlying AI-based solutions. However, such solutions are not likely to be easily adaptable to a more open environment. The problem of managing uncertainty should also be considered from an ethical point of view.

4.6 Assurance and standards

Standards play an important role in engineering and assurance of safety-critical systems. They are based on the best practices and operational history of safety-critical systems. There are general, cross domain standards, like, e.g., IEC 61508 – a generic standard applicable to all kinds of electrical, electronic, and programmable electronic safety-related systems. There is also a large number of industry-specific standards, e.g., ISO 26262 addressing functional safety for road vehicles in automotive domain or DO-178C applicable to airborne systems and equipment and in particular, provides guidelines for the development of aviation software. A newer UL 4600 standard focuses on the safety of Autonomous Systems (AS). It aims at providing a framework for evaluating the safety of autonomous systems, including both hardware and software.

Developing comprehensive standards for autonomous systems is complex due to a number of factors. On the one hand, since autonomous systems rely on AI for its functioning, there should be some kind of general alignment between the standards specifically addressing AI and broader standards for autonomous systems. Since AI is a rapidly developing technology, balancing maturity versus freedom to innovate is one of the open issues. On the other hand, defining the appropriate scope and guidelines for application of standards for safety in the area of AS is also challenging because it should provide sufficient technical guidance and ease of checking at the same time.

Currently, the development of AS is driven by industry. Enabling cross-industry learning and sharing best engineering practices is desirable but not an easily attainable task. Ideally, it would be beneficial to distill the principles governing both product and process-related aspects of engineering and assuring safety as well as create technical guidelines addressing all stages of engineering of autonomous systems.

An important aspect is also training and education in safety standards. Standards should provide a technical guidance that can be implemented using various technologies and processes. Hence, compliance to the standards should go beyond merely checking-off safety requirements and focus on creating a safety management system. In particular, in the context of autonomous systems there should be significant advances in creating guidelines addressing technologies and processes associated with recording and analysing data sets and connecting with testing and deployment data to enable continuous safety monitoring and improvement.

Ensuring the safety of autonomous systems is an evolving challenge, given the complexity and unpredictability of real-world environments. Advances in artificial intelligence and machine learning introduce new variables that traditional safety frameworks must adapt to address. Continuous development and refinement of standards, alongside collaboration between regulatory bodies, industry stakeholders, and research institutions, are essential for the safe integration of autonomous systems into society.

5 Safety of AI-based system versus conventional technical system safety

5.1 How AI safety differs from conventional technical system safety? Towards building a comprehensive fault model

For the conventional systems, safety and development assurance processes operate on the system-level functions. They support implementation choices within a system hierarchy. All these processes are based on a requirements tree. The process rigor and demonstration burdens levied on implementations are based on whether these implementations can cause or contribute to functional hazards, and upon the severity of those hazards. The choice to use AI/ML in an implementation creates as-yet open assurance challenges relative to the use of conventional hardware or software.

For conventional hardware and software, fault models have been developed that provide the basis for compelling validation and verification approaches. These are validated theories of operation that indicate what must be controlled and demonstrated. The assurance process seeks predictability of system behavior, and hardware and software submit to physical and logical laws respectively that allow assessment of their contribution to this prediction. Further, the physical and logical properties of hardware and software guide what to evaluate and test in order to identify and repair faults.

Learned AI/ML models resist both the physical and logical rationales for correct performance, and we as yet lack a theory of their operation sufficient to provide a basis for compelling validation and verification methods. Without such a knowledge and experience basis to indicate what to test and control, we lack a clear path to establishing confidence in associated system behavior. Our current inability to fully understand and predict the performance of these implementations therefore limits their application to functions for which we will accept the associated failure conditions. Such low-criticality applications should, however, be used as learning opportunities.

In order to achieve defensible process controls for the development of AI/ML implementations, we require a validated theory of what to control and why, based on an understanding of the contribution of the controlled factors to the performance of the implementation. This theory must also address the sources of incorrect performance, and the assessment of performance itself will need grounding, definition, and validated methods. It will be necessary, but not sufficient, to address properties of both the data and the inferences. A body of experience in low-criticality deployments should complement desktop analysis in evolving and validating such theory.

5.2 How AI safety differs from conventional technical system safety? Closing traceability gap

The use of AI technologies, in particular ML, for the realization of safety-critical functionality is challenging previous approaches to the safety assurance of software-based systems. In some respects, as safety is considered a system-level property, then the use of AI/ML for the implementation of individual components should not radically change systems safety engineering practices. On the other hand, the complex nature of the tasks implemented by AI/ML, as well as the strong reliance on training and verification data, requires adjustments to safety-critical development and assurance processes.

Requirements engineering for complex systems and environments was identified as a particularly relevant area because it covers a number of aspects related to the tasks for which AI is used as well as the specific impact of data-driven ML techniques. There was a feeling for the need for a well-defined scientific approach to the safety assurance of ML-based functions, including a foundational understanding of the objectives, challenges and demonstrable effectiveness of various approaches to ML safety. New approaches to requirements elicitation and analysis of AI-based, complex autonomous systems are required. These approaches should bridge the "semantic gap" between the engineer's understanding of the task and environment (including the relevance of certain semantic features) and the actual syntactic features of the feature space that are used by the AI/ML components to make decisions. For many complex systems, this gap might only be closed through an iterative process of system validation and requirements refinement.

A general theme through all of the discussions related to the impact of AI on systems safety engineering was that in general more rigor must be applied to existing processes to manage the increased complexity of the environment, task and technology. However, specific additional methodologies are also needed to fill in technology-specific gaps. An example of which is filling traceability gaps between functional requirements specifications and data. We strongly believe that an organizational change management approach is required. Such an approach should improve existing systems engineering capabilities and introduce new capabilities needed for a new generation of systems and engineers.

We also discussed the perspectives of using AI in safety engineering processes. This requires addressing a number of fundamental issues related to the role of human judgment and reliance on automation in the safety analysis and engineering of complex systems. AI can be seen as an additional tool that should be used wherever helpful, but caution is needed to avoid automation complacency (blind reliance on the results of the AI-based analysis).

5.3 Safety architectures incorporating low-integrity ML components: quality assurance perspective

Integrity is an underlying concept of many standards governing safety. It comprises engineering systems that exhibit acceptable hazardous hardware failure rates, that have a sufficiently low probability of design flaws leading to hazardous failures, and are suitable for the application, i.e. safety of the intended functionality. The automotive domain highlighted the latter aspect in the SOTIF- ISO 21448 standard but still has no guidance as to how safe is safe enough (amount of residual risk).

The problem of ML integrity was discussed by experts from different domains – automotive, public transport and aviation – and different technologies – perception, testing, architecture and safety engineering. We focused on identifying the ML-specific challenges including algorithms implementing the models, parameters, data, training and testing.

The pace of innovation in ML field is increasing and many ML technologies do not have safety-related integrity arguments. We discussed the question: How can we build systems relying on AI or ML yet achieve a higher Safety Integrity Level (SIL). A related question is how to integrate no/low-integrity ML components into systems with high SIL?

The promising approaches focusing on quality-assurance include demonstrating safety of ML-based systems by thorough testing, applying rigor in training and testing with regard to data quality.

From the architectural perspective, safety envelopes and diversity were identified as the most promising approaches. As a basis for the discussion, we used a simplified generic architecture that should detect an obstacle with certain true positive and negative rates, and have some medium integrity level (e.g., ASIL B, SIL 2, . . .). This functionality together with the safety targets is determined by the application requirements.

The architecture included two diverse sensors, path detectors and a fusion component. We identified the following necessary conditions for achieving the required SIL. First of all, error characteristics for sensing components should be well-understood and the fusion component must be designed to meet the perception level targets. Hence, application-specific ODD should be an important parameter. Moreover, there might as well be a checker which checks the output of the fusion against the data from the sensors.

Secondly, the system should be able to detect an "out of bounds" condition and have a strategy for encountering such conditions. We assume that an additional monitor (model scope monitoring) might be required. This may be implemented by using ML-based models.

Moreover, we identified the need for a metric for architectures with ML-based components that should be similar to the one, which the classical functional safety provides for hardware architectures. An assessment scheme is needed to gain confidence.

In all domains, in particular aviation, an architecture that safeguards ML components with classical components seems to be a way to gain confidence. However, this significantly reduces the advantages gained by the generalization capabilities of trained ML-based systems. Some experts believe that, in the foreseeable future, the integrity of such systems will be established experimentally rather than by a state-of-the-art for process rigor in training and testing. We also underscored that care should be taken that there is no claim of diversity with no proof of sufficient dissimilarity.

5.4 Safety architectures integrating low-integrity ML components: employing redundancy and safety monitoring

Another session on the same topic focused on discussing the problem of designing appropriate safety monitors for ML components. We discussed that currently, the engineers rely only on test evidences to argue about safety of ML components and there is no established framework for the process-based arguments. The experts agreed that the Potential architectural solution will be functionality-specific, i.e., depending on whether they are used for monitoring perception or planning. It is unlikely that it would be possible to design rule-based monitors for some ML components. Moreover, safety monitors might over-constrain the system.

An open challenge in designing monitors for ML components is arbitration, i.e., defining the principles for resolving disagreements between the monitor and the ML component. One potential solution could be based on a comparison to digital twins/HD-map in run-time. Another interesting solution is to rely on plausibility checks, e.g. by using dynamical models and relying on temporal consistency. Developing approaches supporting co-design of ML components and monitors also constitutes an interesting direction for future research. A blue-sky research direction is to design AI-based safety monitors, though, in this case, it would be even more challenging to argue about system safety.

Finally, we discussed that in classical safety-critical systems, redundant architectures that combine components with low reliability result in creating reliable fault tolerant systems. An interesting research direction is to investigate whether redundant architectures combining multiple low-integrity ML components could also increase system integrity. Further research is needed to understand how to identify common cause failures and argue about integrity of redundant ML-based architectures

5.5 Test planning, validation and coverage

Since autonomous systems operate in an open dynamically changing environment, the question of verification, i.e., has the system been sufficiently tested becomes especially challenging. An overall goal of developing a system that would be "testable by design" has several challenges. Since autonomous systems continuously evolve, the development should be automated, i.e., test cases should be automatically generated from system specification. However, specifications themselves might be incomplete or inconsistent and hence, testing would inherit all the problems of them. One approach to address this issue is the use of AI to assist in test case generation. AI can analyze specifications to identify gaps and inconsistencies, suggesting potential test cases. However, this introduces challenges related to explainability and traceability.

An important aspect to be considered is also creating oracles that are appropriate for testing safety-related behaviour. An oracle in testing is the mechanism used to determine whether a system's behavior is correct. Key questions include when to stop testing and how to define pass/fail criteria. These criteria should go beyond simple failure rates to include quantitative properties that comprehensively assess the system's performance. For example, instead of just counting failures, the oracle could evaluate the severity of failures and their impact on overall system safety and functionality.

Some underlying metrics might be orthogonal to each other. For example, some safe behaviour might result in a significant performance degradation and hence, the question is how to define a trade-off between them. Another aspect is addressing uncertainty: how to appropriately approximate it and align with high-level system safety properties?

Since the system continuously changes and operates in an open environment, it is hard to define coverage criteria. There are different alternatives to focus on: requirements coverage, code coverage or structural coverage. The open challenge is how to ensure that verification coverage criteria are aligned with the validation criteria which focus on scenario and real-world coverage.

The discussion concluded that development and testing should be an iterative process. Based on test results as well as field data, specifications and test cases should be continuously updated, i.e., evolve in response to new information. An iterative process is essential for addressing emerging issues, adapting to new requirements and continual safety improvement.

6 Safety interface with security

Autonomous systems extensively rely on networking in their operation, which motivated the need to analyze the relationships between safety and security. We focused on discussing the critical aspects of automotive cybersecurity, the role of existing standards, and the interactions between safety and security functions within organizations

The discussion emphasized that safety and security are often siloed within companies, yet their integration is crucial, especially in safety-critical systems. The non-compositional nature of security versus the compositional nature of safety presents unique challenges in creating systems that are both secure and safe. There is a strong need to create methodologies that integrate safety and security from the design phase. This includes establishing practices that consider the dynamic nature of security threats while maintaining the integrity and reliability required for safety.

We also identified the key challenge for co-engineering safety and security – balancing the need for frequent security updates with the safety requirement for system. Hence, it is crucial to create management systems that can swiftly respond to new security threats without compromising system safety. We emphasized the need for systems that can adapt to evolving security threats. This includes designing architectures that can handle transient attacks. It is important to develop understanding of the long-term implications of such vulnerabilities on safety.

Automated driving is the most actively developed domain of autonomous systems and, hence, the discussion naturally addressed the issues in automotive cybersecurity. We highlighted vulnerabilities in automotive systems such as the CAN-bus and OBD-port. The necessity for robust cybersecurity measures to protect against and mitigate these threats was underscored.

There was a consensus on the inadequacy of current standards (ISO 26262, ISO 21434, UNECE R155) in providing clear methodologies for integrating security risks into safety assessments. The absence of precise application methods in standards was identified as a significant gap. As a result, the organizations have developed their own protocols, creating inconsistencies and potential vulnerabilities. Research should focus on creating unified, actionable guidelines that can be universally applied to assess the impact of the cyberattacks on safety and mitigate cybersecurity induced safety risks.

Since the use of ML is essential for the design of autonomous systems, it is also important to understand the impact of data poisoning and adversarial attacks on system safety. A key challenge in this respect is to reconcile the need for ongoing system updates to address new security threats with the traditional safety perspective of minimal changes for ensuring stability.

We emphasized the necessity for creating an unified approach to safety and security in autonomous systems. The discussions pointed towards the development of integrated safety-security architectures and the importance of continuous updates and management of both safety and security measures There is a need for greater collaboration between safety and security teams within organizations. Ensuring these teams can work together effectively requires organizational changes and a better understanding of the interdependencies between safety and security.

7 Risk perception for autonomous systems

7.1 Ethical Issues in safety-critical autonomous systems

There is a broad range of ethical issues associated with engineering, safety assurance and deployment of autonomous systems. They include the personal ethics of engineers (conduct, due-diligence), ethics of corporate decisions (to deploy when things may not be fully assured), ethics built into the actual system (to reflect societal norms), and ethical consensus (via standards bodies).

We discussed personal and professional ethics and engineering consensus via standards. It was noted that there are many autonomous vehicles standards, but some of these are of poor quality. We observed that there were conflicting interests in standardisation committees between engineers and corporate voices. Therefore, it is important to be involved with standardisation groups, and thus secure some improvement of the quality of the standards.

Possible standards misuse was also discussed, whether that was doing activities simply to comply with requirements (using a checklist, rather than following processes to add value), or using them to influence policy makers and legislation. For example, SAE J3016 standard (Taxonomy And Definitions For Terms Related To Driving Automation Systems For On-Road Motor Vehicles) had been used to influence policy based on the erroneous assumption that safety risk increased in step with levels of autonomy (rather than highest risk in the middle). This significantly diminishes the benefits of the standard.

An integration of different aspects, e.g., such as ethics and security, which had previously been in separate silos was also discussed. We observed that the use of the term "non-functional" properties implies that they are less important than "functional" properties. It was felt that siloing such issues was not going to be effective for autonomous systems. However, engineering mechanisms to deal with them are not well addressed in safety-specific standards (e.g., bias, fairness, transparency). Additionally, standards may not address the increased level of risk that comes with autonomy and AI. Therefore, potentially there might be a need for greater integrity levels.

There is also a controversial issue of whether ethics in the industry was in alignment with societal norms, and what level of risk was considered acceptable. The experts noted that, in the past, much higher levels of risk were tolerated (e.g., in the early days of steam railways) and that different industries/domains have different expectations. This can also vary in different cultures. We discussed that humans tend to think about short term effects rather than long term, which can make discussions about reductions of accidents over several years difficult. "Goal zero", i.e., no accidents was discussed as to whether it should be aspirational even if not possible in reality. In practice, there will be cost trade-offs, and addressing liability may be prioritised over maximising safety. For example, compliance to standards or regulations may be more important than improved design. We emphasized that there is a clear need for the well-grounded safety culture that would allow for avoiding blame and improving safety reporting.

7.2 How Can We Define "Safe Enough" for an Autonomous Vehicle?

A broadly accepted quantitative metric for assessing the safety of machine learning-based systems for autonomous driving has yet to be established. An adequate definition of safe enough for such systems must address the limitations of the evaluation methods. It must also encompass known, unknown, and uncertain factors present for pre-deployment assurance, post-deployment monitoring and incident evaluation. There is also a clear need to establish the metrics that enable public understanding and potential acceptance of the technology.

Using a reference human driver as the model for Automated Driving System (ADS) performance is intuitive, and might be effective for post-incident system evaluation. However, the feasibility of developing a complete model of a reference human driver for pre-deployment assurance is questionable.

A pure statistical average safety metric is likely to be insufficient. Additional considerations required will likely need to address at least the potential for risk transfer across different population demographics, the potential for negligent driving incidents, and mitigation of specific hazards identified both before and after deployment. Pre-deployment prediction of ADS safety will likely suffer from significant uncertainty. An important question to ask when making a deployment decision is not only whether expected safety levels will be acceptable, but also whether the uncertainty of that expectation has such a large range that there is too high a risk of unacceptable safety outcomes, which might require additional data collection to reduce uncertainty pre-deployment.

The inherent uncertainty in determining ADS risk, along with the difficulty of creating explainable ML decision-making, might make it impractical to take a quantitative-only evidence approach for predicting ADS safety before deployment.

Exhaustive scenario testing demonstrating the capacity of an ADS to make appropriate decisions offers one approach to pre-deployment safety evaluation. However, the lack of a standardized set of testing scenarios makes evaluating scenario completeness difficult, with the potential for missing scenarios contributing to deployment risk uncertainty.

Comprehending ML decision-making is challenging, but the recording of tangible ML outputs and the inferences of "why" they might have occurred can still have substantial value. Understanding where in the ML process an error occurs (e.g., perception, decision-making) helps OEMs identify areas of improvement, and is critical in post-deployment monitoring and incident evaluation.

Post-deployment safety should include an absence of credible unacceptable risk, and in particular OEM should mitigate any identified specific risks. However, there should be an expectation that some level of uncertainty will remain as well as the potential for operational environment changes. As such, the assessment of safe enough must continue throughout the ADS lifecycle. This continuing assessment should rely upon incident information as well as continued ADS data collection.

Access to recorded data will be essential for crash investigators and regulators, as well as for continued assurance by OEM. The benefits of recorded data extend beyond observable incidents. A minor property damage crash has obvious and observable safety implications, but a misclassification of a detected object or a poor path plan that does not lead to an observable incident can also have substantive safety implications. Regulators might assess system capabilities beyond any incidents, and OEMs might identify problematic system areas for improvement.

Open issues and topics worth exploring further in this area include: determining concrete acceptance criteria for deployment and continued operation; the types of data and data collection mechanisms needed; comparative benefits and issues with obtaining data before vs. after deployment; defining a maturity framework for Automated Driving System development and deployment (e.g., akin to a Technology Readiness Level scale); explaining black-box decision-making functions within an ADS; defining and characterizing acceptable risk uncertainties; developing a publicly understandable but accurate safety metric for communicating with general audiences; and withstanding political and/or business-motivated decisions that are counter to acceptable safety.

Participants

- Magnus Albert
SICK AG – Waldkirch, DE
- Ignacio J. Alvarez
Intel – Hillsboro, US
- Claus Bahlmann
Siemens Mobility GmbH – Berlin, DE
- Ensar Becic
National Transportation Savety Board – Washington D.C., US
- Nicolas Becker
Stellantis France – Poissy, FR
- Simon Burton
Gerlingen, DE
- Radu Calinescu
University of York, GB
- Betty H. C. Cheng
Michigan State University – East Lansing, US
- Krzysztof Czarnecki
University of Waterloo, CA
- Niels De Boer
Nanyang TU – Singapore, SG
- Francesca Favaro
Waymo LLC – Mountain View, US
- Lydia Gauerhof
Bosch Center for AI – Renningen, DE
- Mallory Graydon
NASA – Hampton, US
- Jérémie Guiochet
LAAS – Toulouse, FR
- Hans Hansson
Mälardalen University – Västerås, SE
- Fuyuki Ishikawa
National Institute of Informatics – Tokyo, JP
- Aaron Kane
Edge Case Research – Pittsburgh, US
- Lennart Kilian
Siemens – München, DE
- Jörg Koch
Renesas Electronics Europe – Düsseldorf, DE
- Philip Koopman
Carnegie Mellon University – Pittsburgh, US
- Lars Kunze
University of Oxford, GB
- Jonas Nilsson
NVIDIA Corp. – Santa Clara, US
- Ganesh J. Pai
KBR, Inc. & NASA Ames – Moffett Field, US
- Nick Reed
Reed Mobility – Wokingham, GB
- Jan Reich
Fraunhofer IESE – Kaiserslautern, DE
- Martin Rothfelder
Siemens – München, DE
- Philippa Ryan
University of York, GB
- Fredrik Sandblom
Zenseact AB – Gothenburg, SE
- Tiziano Santilli
Gran Sasso Science Institute – L'Aquila, IT
- Jan Stellet
Robert Bosch GmbH – Stuttgart, DE
- Reinhard Stolle
Fraunhofer IKS – München, DE
- Stefano Tonetta
Bruno Kessler Foundation – Trento, IT
- Mario Trapp
TU München, DE
- Elena Troubitsyna
KTH Royal Institute of Technology – Stockholm, SE
- Kim Wasson
Joby Aviation – Santa Cruz, US
- Alan Wassyng
McMaster University – Hamilton, CA
- William H. Widen
University of Miami – Coral Gables, US
- Rafael Zalman
Infineon Technologies AG – Neubiberg, DE

Report from Dagstuhl Seminar 24161

Research Software Engineering: Bridging Knowledge Gaps

Stephan Druskat[*1], **Lars Grunske**[*2], **Caroline Jay**[*3], **and Daniel S. Katz**[*4]

1 **German Aerospace Center (DLR), Berlin, DE.** stephan.druskat@dlr.de
2 **HU Berlin, DE.** grunske@informatik.hu-berlin.de
3 **University of Manchester, GB.** caroline.jay@manchester.ac.uk
4 **University of Illinois Urbana-Champaign, US.** d.katz@ieee.org

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar "Research Software Engineering: Bridging Knowledge Gaps" (24161). The seminar brought together participants from the research software engineering and software engineering research communities, as well as experts in research software education and community building to identify knowledge gaps between the two communities, and start collaborations to overcome these gaps. Over the course of five days, participants engaged in learning about each others' work and collaborated in breakout groups on specific topics at the intersection between the two communities. Outputs from the working groups will be collected in a journal special issue and distributed via a dedicated website.

Seminar April 14–19, 2024 – https://www.dagstuhl.de/24161
2012 ACM Subject Classification Applied computing → Computers in other domains; Applied computing → Education; Software and its engineering
Keywords and phrases community building, Dagstuhl Seminar, knowledge transfer, research software engineering, RSE, software engineering research
Digital Object Identifier 10.4230/DagRep.14.4.42

1 Executive Summary

Stephan Druskat (German Aerospace Center (DLR), Berlin, DE)
Lars Grunske (HU Berlin, DE)
Caroline Jay (University of Manchester, GB)
Daniel S. Katz (University of Illinois Urbana-Champaign, US)

Research Software Engineering (*RSEng*) is the practice of applying knowledge, methods and tools from software engineering in research. Software Engineering Research (*SER*) develops methods to support software engineering work in different domains. The practitioners of research software engineering working in academia – Research Software Engineers (*RSEs*) – are often not trained software engineers. Nevertheless, RSEs are the software experts in academic research. They translate research to software, enable new and improved research, and create software as an important output of research [1].

[*] Editor / Organizer

Hypothetically, the RSEng community and the SER community could benefit from each other. RSEs could leverage software engineering research knowledge to adopt state-of-the-art methods and tools, thereby improving RSEng practice towards better research software. Vice versa, software engineering research could adopt RSEng more comprehensively as a research object, to investigate the methods and tools required for the application of state-of-the-art software engineering in research contexts [2].

There are currently both unknown and known unknowns that make it hard for either community to attain the benefits mentioned above. We call these unknowns *gaps*, and we call methods to discover the unknown unknowns and to clarify or resolve the (subsequently) known unknowns *bridges*.

To find the gaps between research software engineering and software engineering research, and start building bridges between the two communities with the aim to create mutual benefit through reciprocal collaboration, we organized and held a five-day seminar in April 2024 at Schloss Dagstuhl – Leibniz Center for Informatics, as Dagstuhl Seminar 24161 "Research Software Engineering: Bridging Knowledge Gaps". Here, we report and document the seminar's program, outputs, and potential outcomes.

Seminar participants

In the past, there has been little focused direct communication between the RSEng and SER communities. Anecdotally, while the German RSEng and SER conferences co-located in 2023 in Paderborn and shared a break room, there was little, and only informal, exchange in session attendance by participants of either conference, and hence little knowledge exchange.

We specifically organized our Dagstuhl Seminar to serve as a bridge across this type of **communication gap** between the communities, by inviting international experts from both communities as well as individuals who have a track record of working at the intersection between the communities. We also invited experts in adjacent fields such as education and training, and research software funding.

In a pre-seminar survey, we asked participants if they could contribute experience in any of the following areas:

- *Research Software Engineering*: practicing software engineering in a research context
- *Software Engineering*: practicing software engineering
- *Research on Software Engineering*: conducting SER
- *Research on Research Software Engineering*: conducting research *on* RSEng
- *Teaching, training, instruction*
- *Research software funding*
- *Community building*

Answers were provided on a 5-point Likert scale: "None" – "A little" – "Some" – "Much" – "A lot". The results are shown in Figure 1. They suggest that we were mostly successful in bringing together participants with at least some relevant expertise, with the possible exception of research software funding, where more than half of participants claimed little or no expertise. This was partly due to invitees with a known background in research software funding being unavailable to attend the seminar.

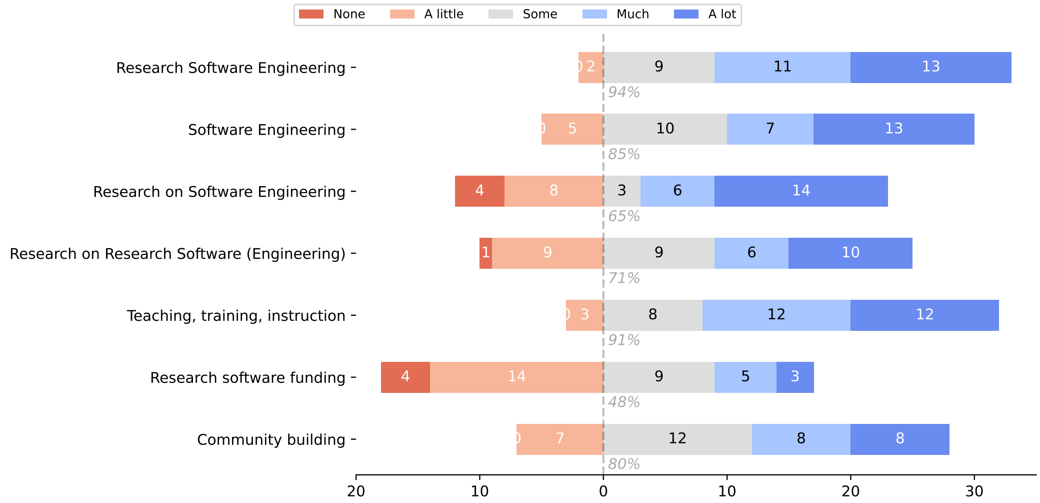


Figure 1 Survey responses to a question on experience in a given area. A total of 35 participants replied to the survey. Each person answered one of five levels of experience in the respective area.

Seminar program

The seminar program was prepared to mainly enable direct collaboration of participants from both the SER and the RSEng communities. Sessions were run either in the plenary, or in breakout groups.

After an introductory session where participants introduced themselves and their work, short presentations were given on ideas for collaborative groups to form at the seminar. From the original set of ideas, participants eventually formed a total of seven breakout groups to work on specific topics. Their work is summarized in the respective working group abstracts in this report. In addition, three additional ideas were brought forward in the course of the seminar in a more informal manner outside of sessions. For these additional ideas, work would either be done in parallel to the sessions, or was planned to be done after the seminar. An overview of all topics is given in Table 1.

Table 1 Topics of working groups and breakout initiatives worked on at the seminar.

Collaboration	**Topic**
Working group	Research Software Engineering Training & Education
Working group	Better Architecture, Better Software, Better Research
Working group	Bridging Communities
Working group	Developing a Common Language
Working group	Research Software: Towards Categories and Lifecycles
Working group	Security and usability of research software
Working group	Demystifying research software engineering for research group leaders
Breakout discussion	Software Engineering Equity, Diversity, Inclusion or Accessibility Research in Research Software/RSEng
Breakout project	Short software engineering social videos

The introductory session was rounded off by a primer presentation of research software engineering, and a preliminary discussion of options for the dissemination of seminar outcomes. Based on early feedback from participants, we decided to give up on our original idea of collecting seminar outcomes as chapters of an edited volume – a field manual for research software engineering. Instead, we discussed options for collecting some of the outputs in a journal special issue, while leaving it generally up to groups to determine the best way for their outputs to be disseminated.

For the remainder of the seminar, work was originally planned to take place in breakout groups, with regular reporting and feedback sessions in the plenary. The third day was planned as an exception, with two fishbowl sessions scheduled for the morning sessions, and a group excursion in the afternoon.

Based on feedback from participants on the first day of the seminar, we realized that there was a wider **perception gap** between RSEs and software engineering researchers than originally anticipated: software engineering researchers specifically did not feel entirely confident in their understanding of the scope and practice of RSEng, and of what research software projects looked like. Vice versa, RSEs did not feel entirely confident in their understanding of the aims and scope of software engineering research. We aimed to address this gap on the second day of the seminar with an ad-hoc "Ask us anything" plenary session where software engineering researchers asked members of the RSEng community questions about their work and experience, and with a subsequent ad-hoc session where RSEs presented the contexts they work in, setup of RSE groups, and particular research software projects.

In light of the need to address this kind of feedback quickly, we moved to adapt the original schedule for each day the night before, incorporating feedback we have had during the day.

In the same spirit as the "Ask us anything" sessions, we ran two sessions of "mythbusting fishbowls," where members of the SER, and then the RSEng, community formed a dynamic panel who discussed topics suggested by the respective other community: preconceptions that one community had about the other's work were fielded via `sli.do`, voted for by the audience, and the most popular ones then discussed by the panel. Panel members were replaced whenever another community member wanted to contribute to the discussion.

Outputs of the seminar activities are presented in brief in the next section.

Outputs

A central *outcome* of the seminar is that community members from the SER and RSEng communities started collaborations to identify and bridge gaps between them.

The outputs of the working groups will be invited to be submitted as articles to a special issue "Research Software Engineering: Discovering and Bridging Knowledge Gaps" in *IEEE Computing in Science & Engineering*, to be edited by the seminar organizers and published in 2025.

Beyond collecting outputs in this way, one of the working groups developed and published a website that aims to collect outputs from the seminar in a way accessible to a wider public and invites contributions from the community: `ser-rse-bridge.github.io`. The website also includes a mapping of terms between SER and RSEng, based on the Guide to the Software Engineering Body of Knowledge [3], which is currently under development (see Section 3.1).

Additionally, a series of short videos was produced during the seminar. In these videos, participants introduce central SER and RSEng knowledge concepts in under a minute. These can be used on social media platforms to create interest in these topics with, e.g., students looking to choose their courses.

Conclusion and future work

In conclusion, Dagstuhl Seminar "Research Software Engineering: Bridging Knowledge Gaps" (24161) was successful in bringing together members of two communities that have a vested interest in research software engineering: research software practitioners and software engineering researchers. Together with software education and community experts, we learned about each others' work and started conversations and collaborations.

Creating conversations between separate communities and their cultures and codes proved to be challenging at times, e.g., where incentives differed. Where we observed antagonism, or where it was brought to our attention by participants, we tried to defuse it and steer conversations into a constructive direction. We are confident that by and large, this worked well due to flexibility on the side of the participants and a general will to collaborate and progress.

We found that adapting the program to the needs of participants where possible while maintaining the general direction, and arguably intensive workload, helped make the seminar very productive and engaging. Participants have continued to engage after the seminar to identify gaps and potential bridges between the communities.

Future work should focus on the continuation of the efforts started at the seminar, and continued communication and collaboration between the communities. We believe that the seminar marked a starting point for collaboration that can realize future reciprocal benefit for research software engineering and software engineering research in equal measure. Interested parties can refer to the seminar website at https://ser-rse-bridge.github.io/ for related resources and activities.

References

1 J. Cohen, D. S. Katz, M. Barker, N. Chue Hong, R. Haines, and C. Jay, "The Four Pillars of Research Software Engineering," IEEE Software, vol. 38, no. 1, pp. 97–105, Jan. 2021, doi: 10.1109/MS.2020.2973362.

2 M. Felderer, M. Goedicke, L. Grunske, W. Hasselbring, A.-L. Lamprecht, and B. Rumpe, "Toward Research Software Engineering Research," Zenodo, Jun. 2023. doi: 10.5281/zenodo.8020525.

3 P. Bourque and R. E. Fairley, Eds., Guide to the Software Engineering Body of Knowledge, Version 3.0. IEEE, 2014.

2 Table of Contents

Executive Summary

Working groups

3 Working groups

3.1 Developing a Common Language: Mapping Between Software Engineering Fundamentals and Research Software Terminology

David E. Bernholdt (Oak Ridge National Laboratory, US), Robert Haines (University of Manchester, GB), Guido Juckeland (Helmholtz-Zentrum Dresden-Rossendorf, DE), Timo Kehrer (Universität Bern, CH), and Shurui Zhou (University of Toronto, CA)

URL https://ser-rse-bridge.github.io/mapping-of-terms/

When trying to build bridges between communities, it is critical that they are able to understand each other – that they "speak the same language". Often in science and technology, different communities tend to communicate more with each other than outside of the community. Over time, this leads a community to develop terminology that is distinctive. When people grounded in different communities meet, they may find that they don't speak the same language – the same word or term may mean different things in each community, and the participants in the conversation may not even realize it.

The term "software engineering" dates back to 1965 and software engineering research (SER) is a well-established academic discipline. Research Software Engineering (RSE), on the other hand, is a relatively young field (the term was coined in 2012) that, in this context, is primarily about applying the concepts, tools, and practices of software engineering to the development of research software (RS). The majority of research software engineers have come from the research software community, and despite the name, rarely have formal training in software engineering – rather than learned on demand as they've pursued their careers, often learning from others in their own community rather than seeking out resources produced directly by the SER community (classes, trainings, papers, etc.). As such the awareness and adoption of what the SER community would recognize as core concepts, practices, and tools, is variable, and often filtered through the experience of others in the RSE community – the two communities don't necessarily speak the same language.

To help build bridges between the two communities, we are developing a map between the terminologies of the SER and RSE communities, along with a rough assessment of the extent to which the RSE community is aware of the concept, the extent to which it is actually used, and the potential for research by the SER community to improve the use of the concept. We are using the Software Engineering Body of Knowledge (SWEBOK) as the jumping-off point for the SE fundamentals that we want to map. SWEBOK represents a systematic distillation of the field of software engineering by that community, though we expect that there will be additional terms arising in both communities that will need to be included in the mapping. The primary output of this work will be a living document, presented via a website, though we plan to summarize the results in papers, presentations, and other venues. We plan to engage first the participants of the Dagstuhl Seminar, and then reach out further in both the RSE and SER communities to flesh out the mapping. We will use the GitHub platform to carry out the discussion and track the revisions as we develop the map.

3.2 Security and usability of research software

Jeffrey Carver (University of Alabama, US), Stuart Allen (Cardiff University, GB), Hannah Cohoon (University of Utah – Salt Lake City, US), Anna-Lena Lamprecht (Universität Potsdam, DE), Christopher Klaus Lazik (HU Berlin, DE), Michael Meinel (DLR – Berlin, DE), and Lata Nautiyal (University of Bristol, GB)

This working group addressed the importance of security and usability in Research Software Engineering (RSE). The benefits of prioritising these quality attributes become apparent relatively late in the software development lifecycle, typically when developers are looking to expand their user base. However, this may be too late for efficient and effective implementation. There is clearly value in ensuring that software is designed from the outset to be both secure and user-friendly. Our work aims to improve awareness and skills related to security and usability in research software development. We find that the appropriate research methodologies for investigating security and usability in the context of RSE are quite similar. At the Dagstuhl Seminar, we conducted a pilot study to understand RSE/SER perspectives on these issues and to assess the level of awareness. These initial results show that despite the critical nature of usability, research software is often perceived as not usable and research software tools are often abandoned due to lack of usability. Furthermore, security is not necessarily perceived by seminar participants as an important quality of research software. We have begun a systematic review of the literature on security and quality in RSE and are planning further work to build on this initial effort, in particular conducting a larger survey and gathering testimonials through interviews.

3.3 Research Software: Towards Categories and Lifecycles

Mikaela Cashman McDevitt (Lawrence Berkeley National Laboratory, US), Michael Felderer (DLR – Köln, DE), Michael Goedicke (Universität Duisburg – Essen, DE), Wilhelm Hasselbring (Universität Kiel, DE), Daniel S. Katz (University of Illinois Urbana-Champaign, US), Frank Löffler (Friedrich-Schiller-Universität Jena, DE), Sebastian Müller (HU Berlin, DE), and Yo Yehudi (Open Life Science – London, GB)

URL https://github.com/ser-rse-bridge/RSE-lifecycle

There is a huge variety of types of research software, at different stages of evolution. This often confuses potential software users, developers, funders, and other stakeholders who need to understand a particular software project, such as when deciding to use them, contribute to them, or fund them. We present work performed by a group who met at a Dagstuhl Seminar consisting of both software engineering researchers (SERs) and research software engineers (RSEs). It includes an initial categorization of research software types, and an initial presentation of an abstract research software lifecycle that can be applied and customized to suit a wide variety of research software types, which then can be used to make decisions and guide development standards that may vary per stage. We also seek community input on improvements of these two artifacts for future iterations.

In addition, because terminologies and definitions often vary, e.g., one person may consider a software project to be early-stage or in "maintenance mode", whilst another project might consider the same software to be inactive or failed. Because of this, we explore and explain concepts such as software maturity, intended audience, and intended future use.

3.4 Better Architecture, Better Software, Better Research

Myra B. Cohen (Iowa State University – Ames, US), Neil Chue Hong (University of Edinburgh, GB), Stephan Druskat (German Aerospace Center (DLR), Berlin, DE), Nasir Eisty (Boise State University, US), Michael Felderer (DLR – Köln, DE), Samuel Grayson (University of Illinois – Urbana-Champaign, US), Lars Grunske (HU Berlin, DE), Wilhelm Hasselbring (Universität Kiel, DE), Jan Linxweiler (TU Braunschweig, DE), and Colin Venters (University of Huddersfield, GB)

In this breakout, we discussed the notion that better architecture leads to better research. Research software engineering requires flexible and modular architectures to accommodate rapid evolution and interdisciplinary collaboration. Hence, we argue that research software engineering should focus on architectural metrics to evaluate and improve their code. Architectural metrics in research software are essential for ensuring the software's scalability, performance, maintainability, and overall quality, facilitating reproducible and reliable research outcomes. We already have many metrics and tools to measure and improve the quality of software architecture. However, we hypothesized that research software is often built using limited resources, and without a long-term vision for maintainability, which may lead to architectural decay. In this breakout, the group (consisting of software engineers, software engineering researchers, and research software engineers) discussed key architectural metrics such as code smells, duplication, test coverage, cyclometric complexity, etc., and how these can be used to improve research software. We explored our hypothesis by applying existing architectural analysis tools to a few open-source research software repositories during our breakouts. We discovered high cyclomatic complexity, large god classes that need refactoring, and low test coverage. We concluded that we should explore this idea further and that there may be an opportunity to build better tools and techniques to help research software engineers improve the architecture of their software, which in turn can improve its quality.

3.5 Bridging Communities: Bringing the Research Software Engineering and Software Engineering Researcher Communities Together for Mutual Benefit

Ian Cosden (Princeton University, US), Jeffrey Carver (University of Alabama, US), Hannah Cohoon (University of Utah – Salt Lake City, US), Stephan Druskat (German Aerospace Center (DLR), Berlin, DE), Nasir Eisty (Boise State University, US), Carole Goble (University of Manchester, GB), Samuel Grayson (University of Illinois – Urbana-Champaign, US), and Samantha Wittke (CSC Ltd. – Espoo, FI)

As the other work from this Dagstuhl Seminar illustrates, there is a chasm between the community of Software Engineering Researchers (SERs) who cater mostly to industry applications and Research Software Engineers who may or may not have formal training in software engineering but develop code for research applications. However, we have identified a number of potential opportunities if we can bridge that chasm: SERs may find novel research questions from RSE experiences, and RSEs could improve their productivity by applying approaches and tools developed by SERs.

Change does not happen on its own. Rather, it must be encouraged and catalyzed by an initial vanguard group and eventually the whole community or communities involved. Therefore, it is incumbent upon those desiring such change to apply concepts from the theory of change, push motivational incentive levers love (open development), power (influence), money (funding), fame (recognition) and create the facilitating conditions of building trust, providing thrust through resources and support, and operating with transparency).

One community observing the other from a distance is a start but not sufficient for lasting change because it treats others as a means to a selfish end. Once community addressing the other is better, but still not enough because there is no feedback from the other to the one. Only true collaboration with cyclic communication, mutual benefit, and shared experiences will be enough for lasting change.

Recognizing the SER and RSE communities have developed and evolved independently, creating bridges between the two represents a cross-disciplinary and cross-cultural endeavor just as significant as between life science and computer science [1]. With this recognition we are developing a guide, as a separate publication, for parties from both sides to better understand how to foster new collaborations between the two communities. This guide, "10 Simple Rules for catalyzing collaborations and building bridges between RSEs and SERs" will outline some of the potential benefits while giving a simple set of rules to follow for both communities to thrive in a new, mutually beneficial collaboration.

References

1 Knapp B, Bardenet R, Bernabeu MO, Bordas R, Bruna M, Calderhead B, et al. (2015) "Ten Simple Rules for a Successful Cross-Disciplinary Collaboration". PLoS Comput Biol 11(4): e1004214. `https://doi.org/10.1371/journal.pcbi.1004214`

3.6 Demystifying research software engineering for research group leaders

Toby Hodges (The Carpentries – Oakland, US), Stuart Allen (Cardiff University, GB), Neil Chue Hong (University of Edinburgh, GB), Stephan Druskat (German Aerospace Center (DLR), Berlin, DE), Lars Grunske (HU Berlin, DE), Daniel S. Katz (University of Illinois Urbana-Champaign, US), Jan Linxweiler (TU Braunschweig, DE), Frank Löffler (Friedrich-Schiller-Universität Jena, DE), Jan Philipp Thiele (Weierstraß Institut – Berlin, DE), and Samantha Wittke (CSC Ltd. – Espoo, FI)

Unfamiliarity among principal investigators with some of the most important principles of research software engineering remains one obstacle to successful integration of software engineering practices into research. Research group leaders unfamiliar with essential concepts and practices in (research) software engineering may find it difficult to provide guidance to RSEs in their projects/groups, or to leverage their expertise effectively as part of the research process. While a growing body of literature, training materials, and other resources exists to help novice research software engineers learn key principles and develop good practices, one reason for the enduring knowledge gap among research group leaders may be that they are not the target audience of such literature, lacking first-hand experience of computational research methods. Often, these resources do not frame RSEng skills within the context of the research process as a whole. Inspired by the popular "Ten Simple Rules" series of articles in PLOS CompBio, this group aims to inform PIs and other researchers about good practices in RSEng – e.g. software testing, documentation, version control, modelling software architecture – and explain the value of these when applied by an RSE to enrich research.

Participants

- Stuart Allen
Cardiff University, GB
- David E. Bernholdt
Oak Ridge National Laboratory, US
- Jeffrey Carver
University of Alabama, US
- Mikaela Cashman McDevitt
Lawrence Berkeley National Laboratory, US
- Neil Chue Hong
University of Edinburgh, GB
- Myra B. Cohen
Iowa State University – Ames, US
- Hannah Cohoon
University of Utah – Salt Lake City, US
- Ian Cosden
Princeton University, US
- Stephan Druskat
German Aerospace Center (DLR), Berlin, DE
- Nasir Eisty
Boise State University, US
- Michael Felderer
DLR – Köln, DE
- Carole Goble
University of Manchester, GB
- Michael Goedicke
Universität Duisburg – Essen, DE
- Samuel Grayson
University of Illinois – Urbana-Champaign, US
- Lars Grunske
HU Berlin, DE
- Robert Haines
University of Manchester, GB
- Wilhelm Hasselbring
Universität Kiel, DE
- Toby Hodges
The Carpentries – Oakland, US
- Caroline Jay
University of Manchester, GB
- Guido Juckeland
Helmholtz-Zentrum Dresden-Rossendorf, DE
- Daniel S. Katz
University of Illinois Urbana-Champaign, US
- Timo Kehrer
Universität Bern, CH
- Anna-Lena Lamprecht
Universität Potsdam, DE
- Christopher Klaus Lazik
HU Berlin, DE
- Jan Linxweiler
TU Braunschweig, DE
- Frank Löffler
Friedrich-Schiller-Universität Jena, DE
- Michael Meinel
DLR – Berlin, DE
- Sebastian Müller
HU Berlin, DE
- Lata Nautiyal
University of Bristol, GB
- Bernhard Rumpe
RWTH Aachen, DE
- Heidi Seibold
München, DE
- Jan Philipp Thiele
Weierstraß Institut – Berlin, DE
- Colin Venters
University of Huddersfield, GB
- Samantha Wittke
CSC Ltd. – Espoo, FI
- Yo Yehudi
Open Life Science – London, GB
- Shurui Zhou
University of Toronto, CA

Report from Dagstuhl Seminar 24162

Hardware Support for Cloud Database Systems in the Post-Moore's Law Era

David F. Bacon[*1], **Carsten Binnig**[*2], **David Patterson**[*3], **and Margo Seltzer**[*4]

1 **Google – New York, US.** dfb@google.com
2 **TU Darmstadt, DE.** carsten.binnig@cs.tu-darmstadt.de
3 **University of California – Berkeley, US.** pattrsn@cs.berkeley.edu
4 **University of British Columbia – Vancouver, CA.** mseltzer@cs.ubc.ca

Abstract

The end of scaling from Moore's and Dennard's laws has greatly slowed improvements in CPU speed, RAM capacity, and disk/flash capacity. Meanwhile, cloud database systems, which are the backbone for many large-scale services and applications in the cloud, are continuing to grow exponentially. For example, most of Google's products that run on the Spanner database have more than a billion users and are continuously growing. Moreover, the growth in data also shows no signs of slowing down, with further orders-of-magnitude increases likely, due to autonomous vehicles, the internet-of-things, and human-driven data creation. Meanwhile, machine learning creates an appetite for data that also needs to be preprocessed using scalable cloud database systems. As a result, cloud database systems are facing a fundamental scalability wall on how to further support this exponential growth given the stagnation in hardware.

While database research has a long tradition of investigating how modern hardware can be leveraged to improve overall system performance – which is also shown by the series of past Dagstuhl Seminars – a more holistic view is required to address the imminent exponential scalability challenge that databases will be facing. However, applying hardware accelerators in a database needs a careful design. In fact, so far, no commercial system has applied hardware accelerators at scale. Unlike other hyper-scale applications such as machine learning training and video processing where accelerators such as GPUs and TPUs circumvent this problem, workloads in cloud database systems are typically not compute-bound and thus benefit less or not at all from such existing accelerators. This Dagstuhl Seminar thus aimed to bring together leading researchers and practitioners from database systems, hardware architecture, and storage systems to rethink, from the ground up, how to co-design database systems and compute/storage hardware. By uniting experts across these disciplines, the seminar sought to identify the architectural changes and system designs that could enable the order-of-magnitude improvements required for the next generation of applications.

Seminar April 14–19, 2024 – https://www.dagstuhl.de/24162
2012 ACM Subject Classification Information systems → Data management systems; Hardware
Keywords and phrases Databases, Modern Hardware, Cloud
Digital Object Identifier 10.4230/DagRep.14.4.54

[*] Editor / Organizer

Hardware Support for Cloud Database Systems in the Post-Moore's Law Era, *Dagstuhl Reports*, Vol. 14, Issue 4, pp. 54–84
Editors: David F. Bacon, Carsten Binnig, David Patterson, and Margo Seltzer
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

David F. Bacon (Google – Seattle, US, dfb@google.com)
Carsten Binnig (TU Darmstadt, DE, carsten.binnig@cs.tu-darmstadt.de)
David Patterson (University of California – Berkeley, US, pattrsn@cs.berkeley.edu)
Margo Seltzer (University of British Columbia – Vancouver, CA, mseltzer@cs.ubc.ca)

This Dagstuhl Seminar on the Future of Cloud Database Systems was convened to address the pressing challenges arising from the stagnation in hardware performance gains, historically driven by Moore's and Dennard's laws. As data continues to grow exponentially – propelled by the expansion of autonomous systems, the Internet of Things (IoT), and machine learning – there is an urgent need to rethink the co-design of database systems and hardware. This seminar brought together experts from database systems, hardware architecture, and storage systems to explore innovative approaches to overcoming these scalability bottlenecks and envisioning the future of cloud database systems.

A central theme of the seminar was the growing disconnect between the exponential increase in data and the slowing pace of hardware improvements, leading to what participants referred to as a "scalability wall." Addressing this challenge requires groundbreaking architectural changes in cloud database systems to support the next generation of applications. One significant area of focus was the potential role of AI-driven hardware and software in reshaping database management systems (DBMS). Participants explored whether AI hardware, such as GPUs and TPUs, could be adapted for database workloads, which traditionally are not compute-bound. Additionally, the concept of leveraging large language models (LLMs) as a new paradigm for databases was discussed, prompting further considerations of the future interplay between AI and DBMS.

To kickstart these discussions, several invited impulse talks were presented, each designed to set the stage for the working groups by exploring possible future scenarios for cloud database systems:

1. **AI Rules:** This talk examined a future where AI hardware and software dominate data centers, fundamentally altering the design and function of DBMS. The discussion centered on how DBMSs might need to evolve in a world where AI is integral to data processing and whether an LLM could serve as a database.
2. **A Disaggregated Future:** This presentations offered a perspective on a future where heterogeneous devices (compute, memory, storage) are connected via ultra-fast networks, creating a fully disaggregated cloud infrastructure. The talk prompted discussions on how DBMS could adapt to and thrive in such an environment.
3. **A Fully Reprogrammable Future:** The talk on this future envisioned a future where all hardware is reprogrammable and customizable at runtime, drastically changing how data processing and storage are handled. The implications for DBMS in such a highly flexible hardware environment were critically examined.
4. **The Pipe Dream:** This session explored the idea of "dreaming up" new DBMS hardware, revisiting the concept of a dedicated database machine. The discussion focused on whether this approach, which has failed in the past, could succeed in the context of modern cloud environments.

Following these impulse talks, the seminar divided into working groups to delve deeper into specific challenges:

1. **Working Group 1:** The Next Order of Magnitude focused on how database technologies can evolve to achieve order-of-magnitude improvements in performance, despite the slowdown in hardware advancements. This group was particularly concerned with managing the exponential growth of unstructured data feeding machine learning models.
2. **Working Group 2:** Memory-Centric DBMS Design advocated for a shift from processor-centric to memory-centric designs, emphasizing the optimization of data access in cloud environments as a solution to the performance bottlenecks caused by traditional architectural models.
3. **Working Group 3:** AI Hardware for Databases investigated how emerging AI hardware, like GPUs and TPUs, could be leveraged for cloud DBMS, even though database workloads typically do not benefit as much from compute-bound acceleration as other applications do.
4. **Working Group 4:** The last working group explored taking disaggregation to the extreme and considering its impact on systems for cloud DBMSs.

As the seminar progressed, participants emphasized the importance of cross-disciplinary collaboration and knowledge sharing. They worked together to draft a comprehensive paper for publication, summarizing the insights and innovations discussed. The seminar concluded with a focus on the need for continued innovation in both hardware and software to meet the demands of future cloud database systems.

In summary, the Dagstuhl Seminar provided a crucial platform for reimagining the future of cloud database systems in light of hardware stagnation. By bringing together leading experts from multiple disciplines and sparking deep discussions through targeted impulse talks, the seminar laid the groundwork for the architectural and system-level innovations necessary to overcome the scalability challenges posed by exponential data growth. The insights and collaborative efforts from this seminar will be instrumental in guiding the development of next-generation database systems.

2 Table of Contents

3 Overview of Impulse Talks

In the following, we provide the information about the (stage setting) impulse talks. While the initial talks had the goal to connect communities by providing the state-of-the-art regarding hardware and cloud databases, the other impulse talks were motivating possible futures how hardware and databases might evolve.

3.1 Computer Architecture 101

David A. Patterson (University of California – Berkeley, US)

The slowing of Moore's Law and the lack of new big ideas to improve CPUs mean slow improvement in general purpose computing in data centers. Innovations in packaging such as chiplets and high bandwidth memories help, but they do not restore the doubling of performance every 18 months that we enjoyed previously. Today the path to much faster general purpose computing that lifts all boats requires scaling out to more computers and more data centers. Similarly, the end of Dennard Scaling means faster computation uses more power. The maximum power of data centers CPUs and GPUs has risen from 300W, which could be air cooled, to 700W and 1200W, respectively, which requires liquid cooling. Domain Specific Architectures (DSAs) are the only path left for big gains in performance. Nevertheless, DSAs must follow Amdahl's Law: performance improvement to be gained from using a faster mode of execution is limited by the fraction of the time the faster mode can be used. Given the high expense to develop new hardware, which domains merit DSAs?

Deep Neural Networks (DNNs) easily justify their own chips and supercomputers. Through boldness and good fortune, NVIDIA dominates commercial DSAs for DNNs with an unconventional architecture and new programming language (GPU/SIMT and CUDA), although most hyperscalers also have their internal inhouse alternatives. DNN DSAs have leveraged on fast matrix multiply, high memory bandwidth, and new narrow data types. We have likely reached a historic tipping point in data center hardware. From the 1960s to the 2010s, conventional software was king and Moore's law still held, so CPUs dominated the hardware investment and deployment. Looking forward, Moore's Law is slowing and AI software now holds the throne, so we expect DNN DSAs will be the majority of investment and deployment.

3.2 Some Hardware Impacts on Cloud Databases

Mark D. Hill (University of Wisconsin-Madison, US)

This talk addresses three hardware impacts on cloud database systems. First, using accelerators will be necessary to get more rapid performance improvement than will be possible with slowly improving CPUs. To this end, I recommend that innovation first target existing accelerators like GPUs, TPUs, SmartNICs, data movers, encryption, and compression. This avoids the substantial investment that new accelerators require. Second, memory is and will likely be the bottleneck for many computations. With general purpose CPUs, database work should consider exploiting Compute eXpress Link (CXL) that enables more more memory to be attached to CPUs than possible with directly-attached DDR memory alone. For GPUs, high-bandwidth memory (HBM) will provide high bandwidth, but low capacity, leaving the challenge for how the next memory tier should evolve and be exploited. Third, the public cloud is evolving toward confidential compute wherein tenant data is protected cryptographically end-to-end. New database ideas and accelerators must be compatible with confidential compute to impact the public cloud.

3.3 Cloud Databases (OLTP) – Where are we and where are we going?

David F. Bacon (Google – New York, US)

There are two fundamental types of databases: operational (OLTP) and analytic (OLAP). My presentation focuses on Spanner, Google's largest OLTP database; Justin Levandoski will cover BigQuery, which is our analytic database, in the next session. Since this is an "opinionated overview" and I haven't worked on other Cloud database systems, I'll invite others in the audience to contribute details where other systems are different. Spanner stores 15 EiB of data and runs at over 4 billion QPS. Its data under management has been doubling roughly every year. Given the hardware trends described earlier in the talks by Dave Patterson and Mark Hill, we are at a crossroads where it will either cost dramatically more to store data each year, or we will have to dramatically slow the growth in data storage, or we will have to find new ways to storing and operating on data efficiently. As a co-organizer of this event, it was solving this problem that motivated me to help bring us together.

Strong Consistency Wins. Spanner is now used to store the data for virtually all of Google's largest consumer products, including Search, YouTube, Gmail, Drive, Photos, Maps, Meet, and so on. It also stores much of the data for the Google Cloud control plane and for Google's internal infrastructure. Finally, it is offered as a product as part of Google Cloud. Cloud Spanner is used by Uber, Walmart, Ford, and others.

Google was instrumental in launching the "NoSQL" movement for building its hyper-scale applications. A fundamental part of NoSQL (although the term has become somewhat fuzzy over time) was using relaxed consistency as a way of achieving scale. While this may still be true in parts of the industry, at Google we have come full circle: Spanner has solved the problems of strong consistency at scale. Meanwhile, we have repeatedly had the experience that building products without strong consistency may work well in the beginning, but as products evolve, add more features, and become more heavily relied upon by the public, that weak consistency becomes an Achilles heel. Each application wound up trying to hide weak consistency from users, and building custom protocols which are often ad-hoc and error-prone.

In 1976, when System R was introduced with the relational model, SQL was running at perhaps 20 QPS on 60 MB of data. Since then SQL has scaled by 11 decimal orders of magnitude in storage and 8 orders of magnitude in QPS. This is a testament to the remarkable power of its declarative programming model and its ability to express massive parallelism in an architecture-independent way.

The Scaling Wall? However, with the end of Moore and Dennard scaling, exponential data growth would cause almost exponential increase in cost, which is untenable. However, data growth shows no sign of slowing down. In fact, LLMs and related technologies seem to be accelerating data growth. At its current growth rate, Spanner could reach a zettabyte (1000 exabytes) by 2030, with an aggregate of 1 trillion QPS. Meanwhile, data is getting colder, on a per-byte basis. Spanner's storage is growing substantially faster than its QPS and CPU consumption. While more and more IOPs are moving to SSD, HDD IOPs are an increasingly large part of the total system cost.

No Silver Bullet. Unfortunately, there is no clear answer to these problems. Database systems are notoriously diverse in their computational load. Spanner's hottest function is roughly 2% of total CPU time, so there are no "kernels" amenable to hardware optimization. The biggest opportunity lies in data compression. Compression is a modest amount of total CPU time, but that is due to the fact that better compression is too expensive in software. If we had cheap compression available, we could compress more aggressively. The biggest savings would come not from the CPU time, but from the savings in HDD, SSD, and IOPs. Total system optimizations to make more effective use of the hardware, either by improving CPI or by reducing power consumption, are also fruitful areas for exploration. The best hope for hardware acceleration actually lies in creating new database paradigms and workloads, which have more computational density. While it isn't yet clear how, AI will clearly play a large role here.

Reliability Challenges at Exascale. Google has previously reported that a small but significant number of cores occasionally performs incorrect computation. Unlike memory or disk corruption, where there are long-standing mechanisms to detect and prevent corruption, we do not have experience with corruptions in the computational path. These corrupted computations can and do lead to corrupted data, and are therefore of enormous concern for Spanner. As we move towards exascale databases, we will need to tame this problem if we are to maintain the data integrity guarantees that our users expect.

3.4 Cloud Databases (OLAP) – Where are we and where are we going?

Justin Levandoski (Google – Seattle, US)

This talk provided an overview of current cloud-native analytics system architectures (e.g., Amazon Redshift [2], Snowflake [10], Google BigQuery [22]) that separate compute and storage. It then focused on the high-level architecture of BigQuery that also disaggregates memory for intermediate shuffle. This serverless and disaggregated design has several benefits, as it (1) allows for on-demand *scaling* of each resource, (2) allows for on-demand *sharing* of resources, and (3) adapts well to *multi-tenant* USge at *lower cost* – all of which is important to running a cloud data service at scale.

While the benefits and flexibility of compute/storage separate are well known, this talk reiterated the benefit of disaggregated memory for shuffle (also covered in [22]), which (1) reduced shuffle latency by and *order-of-magnitude*, (2) enabled *order-of-magnitude* larger shuffles, (3) reduced resource cost by 20% by allowing the system to avoid resource fragmentation, stranding, poor isolation of memory resources. This talk ended by introducing new workloads on the horizon for cloud data warehousing, focusing on unstructured data becoming a first-class citizen in these systems and the overall trend of traditional data warehouses becoming general purpose cloud data platforms for all data types [19].

3.5 The AI Future: ML for Systems

Tim Kraska (MIT – Cambridge, US)

Machine learning (ML) and Generative AI (GAI) is changing the way we build, operate, and use data systems. For example, ML-enhanced algorithms, such as learned scheduling algorithms and indexes/storage layouts are being deployed in commercial data services, GAI-code assistant help to more quickly develop features, ML-based techniques simplify operations by automatically tuning system knobs, and GenAI-based assistants help to debug operational issues. Most importantly though, Generative AI is reshaping the way users interact with data systems. Even today, all leading cloud providers already offer natural language to SQL (NL2SQL) features as part of their Python Notebook or SQL editors to increase the productivity of analysts. Business-line users are starting to use natural language as part of their visualization platforms or enterprise search, whereas application developers are exploring new ways to expose (structured) data as part of their GAI-based experiences using RAG and other techniques. Some even go so far and say that "English will become the new SQL" despite the obvious challenges that English is often more ambiguous. Arguably, industry is leading many of these efforts and they are happening at unprecedented speed – almost every week there is a new product announcement. Yet, a lot of the work feels ad-hoc and many challenges remain to make ML/GAI for systems in all these areas really practical despite all the product announcements. In this talk, I provide an overview of some of these recent developments and outline how the academic solution often differs from the ones deployed in industry. Finally, I list several opportunities for academia to not only contribute but also build a better, more grounded foundation.

3.6 The AI Future: Do we need Databases at all? Or Model = DB?

Carsten Binnig (TU Darmstadt, DE)

In recent years, the DBMS community has outlined a new direction of so-called learned DBMS components, where core components such as indexes or query optimizers are replaced by machine learning (ML) models. In this talk, I conducted a (somewhat extreme) thought experiment and asked the question: "Can we replace the entire DBMS with an ML model" or in short "DBMS = ML model". This direction is particularly interesting as "DBMS = ML model" would allow us to run DBMS natively on AI hardware and leverage advances of AI hardware in which massive investments are being made today. In addition, recent results on learned DBMS components have shown that they can significantly improve DBMS performance. However, it is still far from clear whether a complete DBMS can be replaced by a single ML model or whether this is simply impossible. Wait, is it really unclear whether "DBMS = ML model" can be true?

In my talk I implied that there is (at least) hope that "DBMS = ML model" can be true. For example, if we look at LLMs today, we can see that LLMs are already being used as a type of database because they support question-answering on external data sources, including tables, when used in combination with retrieval in what is called retrieval-augmented generation. Beyond the fact that we can thus use AI hardware as discussed above, using LLMs as DBMSs offers many other possibilities for modern DBMS workloads which need to deal with multimodal data (images and text). However, LLMs have many known weaknesses such as hallucinations and other issues that need to be addressed in order to utilize them for query answering and act as a replacement for DBMSs.

3.7 The AI Future: Where is AI HW going?

Holger Fröning (Universität Heidelberg – Mannheim, DE)

Graphics Processing Units (GPUs) and Deep Neural Networks (DNNs) synergize to advance computational capabilities. GPUs, originally for graphics rendering, accelerate DNN training and inference with parallel processing. DNNs' demanding computations benefit from GPUs' parallel architecture, driving efficiency and speed. As DNN complexity grows, so does the demand for more powerful GPUs, spurring GPU advancements. In turn, GPU evolution fuels DNN innovation, enabling breakthroughs in computer vision, natural language processing, and autonomous systems. This reciprocal relationship propels technological progress, shaping the future of AI and computational sciences.

In this light, this talk will review fundamentals of CMOS technology scaling, in particular considering the end of Dennard scaling. As a result, it is anticipated that overall performance in terms of operations per second is rather governed by power consumption budget and energy efficiency in operations per Joule. Furthermore, technology analysis suggests that data movements are more expensive than computations.

The talk will conclude with a couple of research directions in the light of these observations.

3.8 The AI Future: AI rules (NOT)? Real-Time Intelligent Systems

Anastasia Ailamaki (EPFL – LaUSnne, CH)

Database systems face new hurdles with the rise of diverse hardware infrastructures and varying workloads. Evolving hardware, with its mix of different components and dynamic configurations, complicates how data moves within systems, and affects performance and robustness. At the same time, the surge in data-centric analytics and powerful AI models brings a wide range of workloads that systems must handle. The old way of designing systems based on predicted performance no longer works, leading to inefficiencies.

This talk analyzes the complexity of heterogeneous hardware and diverse workloads in database systems. It highlights the limitations of standard approaches and stresses the need for systems that can adapt without relying on fixed assumptions about hardware or workloads. The future is **real-time intelligent systems**, i.e., systems that adapt to changes in their execution environment in intelligent ways using ML, GenAI, abstraction and just-in-time code generation. The goal is modular, composable infrastructures which enable cross-optimizations dynamically, while preserving separation of concerns in system design.

3.9 A Disaggregated Heterogeneous Future: An Overview

Gustavo Alonso (ETH Zürich, CH)

The trend towards hardware specialization and disaggregation raises the question of how data processing engines will be take advantage of these developments or, at the least, adapt to them. In the talk, I point out that the biggest bottleneck in data processing is the data movement and new hardware can be used to turn the data path from storage to processing units into a series of active components that filter, reduce, transform, and pre-process the data. I give an example of how this can be implemented using smart storage (e.g., to project data out), a smart NIC on the storage system (e.g., to filter the data further), a controller in disaggregated CXL memory (e.g., performing an initial hashing of the data), a smart NIC on the computing node (e.g., decompression and decrypting the data), and an accelerator between memory and caches (e.g., hashing the data to partition the data across different processors). Such a pipeline of processing elements is feasible today and can be used as an experimental platform to inform hardware evolution and better understand how near-data-processing can be orchestrated to achieve the biggest possible gains.

3.10 A Disaggregated Heterogeneous Future: Building Cloud-native Data Systems for the Post-Moore Era

Jana Giceva (TU München – Garching, DE)

Considering the disaggregated cloud environment, there are many open questions on how to build accelerators for data intensive applications and the impact resource disaggregation may have on the whole system stack. In this talk I touch upon a few directions that are worthwhile discussing in this context and exploring further in the spirit of Dagstuhl. In the first part I introduce the idea of using operator primitives as a building block both for expressing various types of dataflows (beyond relational) and for enabling hardware acceleration across the data-path in the era of resource disaggregation and active hardware components. In the second part I discuss the systems challenges and opportunities for adopting such primitives in practice. For example, one idea is to start treating databases as domain specific compilers, so we can transform optimizations of the logical plan into compiler passes, before generating a physical plan (DAG) of tasks with binaries suitable to the target environment. Nevertheless, for the primitives to be fully adopted we need to find a way to address the heterogeneity of the underlying memory- and compute-models, the need for data transformations, and virtualizing the resources in the cloud context. And finally, in the third part I propose using a memory-centric view of the system as a programming model for fully disaggregated system.

3.11 A fully (Re-)Programmable Future: Cloud Databases and Hardware

Zsolt István (TU Darmstadt, DE)

In modern clouds, Resource Disaggregation has been adopted as a way of offering scalability and efficient resource utilization for large-scale applications. Provisioning CPU, memory, and storage resources independently for distributed data-intensive applications is a great enabler and cloud databases are already designed with disaggregation in mind. In this talk, we focus on an exciting opportunity that emerges in this context, namely, to dramatically increase the efficiency of cloud databases through the use the specialized and programmable hardware devices that already underpin resource disaggregation. To take advantage of this opportunity, however, we need to overcome several challenges. First, we need to find practical ways of co-designing databases and the offloaded hardware functionality. Second, programming such devices must become easier, to be able to achieve good performance without having to entirely re-design operators. In this talk I sampled from relevant related work and from our early results on overcoming these two challenges, getting us closer to a fully programmable future for cloud databases.

3.12 The Pipe Dream: Database Systems Chasing Hardware

Jignesh M. Patel (Carnegie Mellon University – Pittsburgh, US)

New hardware is typically designed to support new essential applications that are poorly served by existing commodity hardware or to address fundamental architectural constraints. Database systems are modular in their internal organization, allowing them to adapt to new hardware in ways other software applications generally cannot. Thus, database applications haven't become critical motivators for new hardware, and this trend will continue. To get high performance, database systems must adapt in two primary ways. First, methods that reduce data movement will be even more critical. This was the primary reason analytic database systems went from row to column stores. We can further slice columns by bits and develop bit stores to reduce data movement even more. The second observation is that AI will drive the development of new hardware and that database systems have no choice but to adapt to run efficiently on this new hardware, which includes GPUs today. Fortunately, this seems possible, given the modular abstractions in database systems. Thus, the future of high-performance database systems is in developing methods that intrinsically reduce data movement and enable them to run efficiently on available diverse commodity hardware.

3.13 The Pipe Dream: Hardware Acceleration For Databases

Lisa Wu Wills (Duke University – Durham, US)

In this talk, we ask the question of whether it is possible to accelerate databases by having a true hardware-software co-design. Surveying the hardware development landscape, hardware specialization is motivated by more efficient processing where efficiency is defined as higher performance, lower power, lower energy, and therefore lower total cost of ownership. We introduce the concept of using datatype acceleration to raise the level of abstraction when designing hardware and leveraging already-defined software method calls and containers to provide more efficiency. We showed a classic example of Q100 accelerating databases achieving one to two orders of magnitude of performance and energy efficiency using heterogeneous functional tiles and exploiting pipeline and data parallelism. We then pose three possible futures for exploration: 1) replacing some worthwhile software operations with hardware primitives, 2) using an FPGA to provide fused specialized units for common database operations, and 3) providing hardware support for primitives used in a data-centric computing world.

4 Working Group 1: The Next Order of Magnitude

Gustavo Alonso (ETH Zürich, CH, alonso@inf.ethz.ch)
Carsten Binnig (TU Darmstadt, DE, carsten.binnig@cs.tu-darmstadt.de)
Mark Hill (University of Wisconsin-Madison, US, markhill@cs.wisc.edu)
Ihab F. Ilyas (University of Waterloo, CA, ilyas@uwaterloo.ca)
Justin Levandoski (Google – Settle, US, levandoski@google.com)
Jignesh M. Patel (Carnegie Mellon University – Pittsburgh, US, jignesh@cmu.edu)
Holger Pirk (Imperial College, London, UK, pirk@imperial.ac.uk)
Tobias Ziegler (TU Darmstadt, DE, tobias.ziegler@cs.tu-darmstadt.de)

Data growth continues being exponential[1], especially regarding the unstructured data feeding machine learning and large language models. In the past, hardware advancements enabled keeping pace with this trend. However, with the slowing of Moore's Law [28], managing large data volumes solely through hardware improvements has become increasingly challenging. This raises a critical question: if data continues to grow exponentially, how can we evolve database technologies to achieve order-of-magnitude improvements in performance?

Data workload growth and addressing *Hill's law*

At a high level, database workloads can be divided into (1) *Online transaction processing (OLTP)* that handles transactional/operational workloads for a system-of-record and (2) *Online analytics processing (OLAP)* for large-scale data analytics and enterprise business

[1] https://www.statista.com/statistics/871513/worldwide-data-created/

reporting. We argue that OLTP workloads are unlikely to experience exponential growth. Several of the authors have spent decades in industry building and operating cloud-based transaction processing systems. For the vast majority of these workloads (the 99.999%), a single large machine typically suffices. Furthermore, transaction processing workloads easily partition (e.g., by use case) and can thus horizontally scale without needing to coordinate cross-partition transactions.

Conversely, analytics workloads are experiencing a renaissance in both data volume growth and workload types. Traditionally OLAP workloads dealt with precise structured tabular data that is aggregated and fed into business intelligence/reporting applications. While valuable, these "traditional" analytics/BI workloads are expected to grow modestly at 20-30% year-over-year in line with the current cloud data warehousing market. A key exponential growth area for OLAP will be unstructured data and the new workload types it will bring about at the intersection of AI/ML and analytics. There are three key trends driving this growth:

1. **Unstructured data growth and collection in the enterprise.** With the advent of cheap cloud object storage, it has become economical for enterprises to store unstructured data (e.g., pdfs, video, audio, images) in raw form. As one data point, the IDC expects 80% data to be unstructured by 2025[2], and is driving new use cases for analytics on images, audio, speech and text.
2. **Cloud data warehouse architectural shifts and customer expectations.** The data analytics industry is shifting toward a so-called "lakehouse" approach to data management, whereby data warehouses evolve into general-purpose data orchestration platforms for all data types. This architectural shift separates storage and compute. Data is now stored in a scalable storage layer and the data warehouse software provides the compute layer orchestrating IO and computing on that data as needed. This architecture can be generalized to handle all kinds of data including non-tabular unstructured data. This architecture meets customer expections of a single system (or the illusion of a single system) to seamlessly handle both traditional data warehousing and advanced analytics use cases [2, 19].
3. **Democratization of in AI/ML inference/extraction.** Powerful LLM capabilities and advances in AI/ML inference techniques have democratized the ability to extract valuable enterprise data from unstructured data (e.g., audio, video, images, pdfs). Hence, more structured, but less accurate, tabular data will be available to OLAP workloads based on these techniques.

This new reality will affect OLAP workloads in three ways. First, data curation and cleaning workloads will incur a significant jump in both compute and storage budget. Second, the volume of structured data generated from inference over unstructured data will grow exponentially compared to that traditionally ingested from structured/tabular sources. This structured extraction will take place directly within the analytics platform, as unstructured data has become a first-class citizen in these systems, e.g., through BigQuery object tables [19] or Snowflake directory tables[3]. This functionality gives rise to a new workload that is expected to dominate the storage and compute cost of modern OLAP stacks. Last but not least, we will soon embed this multi-modal data as large vector stores to enable features such as dense-retrieval, clustering and serving downstream models. It is probably too early to tell

[2] https://solutionsreview.com/data-management/80-percent-of-your-data-will-be-unstructured-in-five-years

[3] https://docs.snowflake.com/en/user-guide/data-load-dirtables

how these vector stores will change the overall data growth, but it definitely calls for a new scaling paradigm since data growth cannot simply be handled by hardware advancements anymore. We will call this observation *Hill's law.*

Hill's Law: The exponentially widening disparity between accelerating data growth and modest hardware improvements must be covered by exponentially better data reduction techniques.

Will hardware improvements be sufficient to handle data workload growth?

The answer is a qualified "no:" Hardware improvements may be able to cover the growth of OLTP and structured OLAP workloads, but will be woefully insufficient for straightforward handling of the exploding growth of exploratory OLAP processing of unstructured data.

A key hardware impact on data and other workloads occurs because some hardware parameters scale at different rates that others. One must pay particular attention to this now as 2D transistor scaling slows (Moore's Law).

- 2D logic scaling of general purpose cores on SOC package is slowing but will still proceed faster than scaling of DDR memory bandwidth off package and memory capacity per chip.
- Compute eXpress Link (CXL) will allow more memory bandwidth and capacity but at substantial cost. CXL also enables the hardware system designer to flexibility allocate a package's "lanes" to memory, I/O (PCIe), or accelerators.
- High-bandwidth memory (HBM) will continue to provide GPUs (and others) high bandwidth at high cost, but with limited capacity.
- SSD capacity will grow well due to its monolithic 3D implementation while its bandwidth growth will follow PCIe growth.
- Hard disk drive capacity and bandwidth will likely grow very slowly.
- Optical interconnect use will move closer to computing systems and will eventually terminate on SOC packages ("co-packaged" optics). This will unlock more bandwidth that can reach further, e.g., for disaggregation.

These disparate scaling trends will encourage hardware systems that carefully husband bandwidth (memory and I/O) and capacity (memory and hard drive) more than other resources. In the extreme, one can model computation as free.

With careful design, we expect hardware improvements may be more or less sufficient to cover the relatively slow growth of OLTP and Structured OLAP. However, substantial innovation in algorithms, software, and (co-designed) hardware will be needed to support exploding OLAP processing on exploding unstructured data.

First order principles to address Hill's Law

For cloud data platforms, the hardware developments outlined before and in particular the stagnation regarding bandwidth (both SSD and memory bandwidth) as well as memory capacity will have a significant impact on analytical data platform where data is growing at fast rates. In the following, we discuss first order principles that will help future cloud DBMS to get around these hardware limitations.

(1) Increase information density. A first principle that will help us to support the future growth in data without sacrificing performance is the principle that we need to increase information density per bit (IDB). An important aspect is that the information density needs to be increased along the full data access path from storage over memory until data hits the compute. This will have two important consequences: First, when we need to move data along the hierarchy from storage over memory to compute, we can significantly reduce the footprint of the data movement, which means we can "move more for less". Moreover, at the same time when keeping the information density also the footprint of intermediate data stored in memory will also be reduced which will help us to "store more for less".

(2) Consider computation free. Computation, unlike bandwidth, is projected to scale with the increasing number of processors that can be integrated into a server. So, we advocate for the following second core principle: *exploit the "free" computation capacity.* This principle is in particularly interesting since it closely aligns with our first principle, as it allows for a trade-off between computation and bandwidth and effectively use the information density per bit. For instance, by integrating more aggressive compression schemes such as heavy-weight compression along the full data access path until, we can keep data footprint low until it reaches the compute and use (free) computation to inflate data once it hits compute. Additional techniques to leverage this principle will be discussed in the following.

(3) Use better what we already have. As an alternative to reduce data footprint, as a third principle we can use resources better that we already have. Current infrastructure suffers from stranded memory resources, i.e., memory that is not fully utilized. Microsoft, for example, reported that up to 25% [20] of memory capacity is stranded. This under-utilization has long been an issue but has become even more critical as memory capacity now represents a constrained resource. As such, we should optimize the use of memory to avoid wastefulness and manage costs effectively. In particular, there are two ways forward: (1) We should adopt the principle of resource-constraint systems that memory is precious instead of trading memory for computation, e.g., by large pre-computed lookup tables. (2) more flexible database architectures, e.g., by using CXL to pool memory [20].

What can DBMS do to deal with the growth in structured data?

It is clear from the discussion above that storage devices will be more bandwidth-constrained than capacity-constrained in the future, and this discrepancy will only worsen over time. Further, the other hardware trend is that compute cycles are plentiful and nearly free. These driving factors open up new opportunities for efficiently scaling data management techniques by leveraging the first principles above.

Thus, methods like compression and encoding that intrinsically increase the IDB and thus reduce data sent across communication channels will be essential. Leveraging the nearly "free" compute cycles and "cheap" storage capacity, one can be far more aggressive in pre-computation, replication, and summarization of data and query results. Memory capacity, however, is likely to be an increasingly constrained resource, and one will have to throw away data far more aggressively in that layer. However, since analytic data systems are often bandwidth-constrained, designing methods that trade storage bandwidth for lower memory capacity will be critical.

What can DBMS do to deal with the growth in unstructured data?

Unstructured data use cases are relatively new and emerging quickly driven by Generative AI technologies. Coupled with the rapid changes in hardware, we can only speculate on the approaches needed to deal with this class of data applications.

At the data scales we have today and due to the influence of machine learning, there is the opportunity to take advantage of lossy compression and approximated computing as way to reduce the amount of data moving from storage to processing units. There might also be an opportunity to apply learning to data so that there is no need to process the data to find the answer to a query since the answer can be obtained through inference over a model (similar to the way some queries can be answered by looking at an index rather than at the data). These approaches would be a direct application of Hill's Law.

Similarly, data selection and model behavior attribution techniques [16, 24] operating over, for example, training data can help in reducing data movement, especially if the sampling and filtering can be directly done on computational storage, thereby providing hardware support for the improvements needed in Hill's Law. This will also require to revisit conventional data processing algorithms so that they can operate on approximated subsets of the data and low precision vector representations. Finally, given the size of vectors employed in ML and LLMs, a more efficient used of memory though different representations and data organizations will help in reducing the impact of limited I/O and memory bandwidth on our ability to process large amounts of data.

5 Working Group 2: A Case for Memory-Centric Design of Cloud Servers and DBMS

Anastasia Ailamaki (EPFL – Lausanne, CH, anastasia.ailamaki@epfl.ch)
Lawrence Benson (TU München, DE, lawrence.benson@tum.de)
Helena Caminal (Google – Sunnyvale, US, hcaminal@google.com)
Yannis Chronis (Google – Sunnyvale, US, chronis@google.com)
Jana Gieceva (TU München, DE, jana.giceva@in.tum.de)
David A. Patterson (University of California – Berkeley, US, pattrsn@cs.berkeley.edu)
Eric Sedlar (Oracle Labs – Redwood Shores, US, eric.sedlar@oracle.com)
Lisa Wu Wills (Duke University – Durham, US, lisa@cs.duke.edu)

The exponential growth of data in cloud-based systems, coupled with the plateauing of traditional processor performance gains, has created a fundamental bottleneck for database management systems (DBMS). The processor-centric architectural model, dominant for decades, is now hindered by slowing improvements dictated by Moore's Law and Dennard Scaling. This shift in the performance landscape requires a rethinking of DBMS design principles, moving the focus from pure computation to optimizing data access via memory and storage.

Conventional processor-centric architectures place the CPU as the central element, surrounded by statically allocated DRAM and connected to storage and networking via I/O buses. While this design served well when all components improved at similar rates, it is increasingly unsuited to modern workloads. Innovations like chiplets help mitigate limitations on the compute side, but they cannot compensate for the DRAM and data management challenges posed by modern datasets. Furthermore, the focus on computation in traditional design conflicts with the fundamentally data-oriented nature of DBMS operations.

In response to these challenges, we advocate for a memory-centric design approach. This model dissociates processors and data, allowing processors to access a shared pool of DRAM as needed. This improves DRAM utilization and reduces the cost impact of DRAM, which is increasingly expensive compared to other components. Additionally, memory-centric systems strategically leverage low-cost processors near storage devices. These "smart" storage nodes pre-process data on-site, performing operations like filtering, aggregation, and compression, thereby minimizing the need for costly data movement to higher system levels. In short, the emphasis is on moving the computation to the data, not the other way around.

The recent CXL (Compute Express Link) industry standard enables realistic implementations of the memory-centric concept. CXL creates a cacheable shared memory space across multiple sockets, where at least some of the space is hardware coherent. While CXL introduces some latency and bandwidth trade-offs compared to local DRAM, the optimization of data locality and utilization more than compensate for these factors in DBMS workloads. For long-term scalability, CXL switches and optical interconnects could dramatically expand the size of the shared memory pool.

A memory-centric approach fundamentally transforms the design and optimization of DBMS systems. Here are key areas of change:

Query Execution

Traditional query optimization generates physical query execution plans from relational algebra expressions using an estimate of the available compute and memory resources as well as a cost model. Query processing relies on the optimizer's decisions to maximize efficiency through selecting specific operator implementations and implicit data movement (e.g., repartitioning, data shuffling with the exchange operator in a distributed setting). Both modules focus on minimizing processor cycles while mindfully using memory resources. Memory-centric computing makes data a first-class citizen. Query processing and optimization in memory-resident query execution engines will focus on pushing processing to and operating directly on data stored in local storage. Query optimization will produce query plans which make decisions about distribution of execution. The physical query execution plans will be produced locally on each node and may vary depending on node capabilities and local memory technologies (just-in-time mappings to local micro-architecture and code generation will produce the final query executions plan on each participating node during execution time, leveraging cache-conscious algorithms). When data from remote storage is needed, engines must push operator code to the processors attached to the remote storage. The integration of low-cost general-purpose cores (ARM/RISC-V) into those devices is a reality and recent innovation of tightly-integrated implementations of processing-in-storage will further increase the benefits of the memory-centric computing scheme. The more computation is executed close to storage the more we can reduce the pressure on the capacity and bandwidth of pooled DRAM. When the execution is placed near the data, instead of copying data between compute units, where possible it should be passed by reference. Memory-centric indexing needs to facilitate access to local data and use references to remote indexes to facilitate code transfer. Real-time adaptive query processing algorithms in combination with code-generated operators can bind the query processing logic with the specifics of memory microarchitecture, thereby optimizing for in-situ hardware characteristics. This makes query processing a natural fit to the memory pooling design, and enables more powerful operations to be offloaded to where the data actually sits (e.g., closer to storage).

Transaction Processing (OLTP)

Cloud-native transaction processing can significantly benefit from a memory-centric design as it allows for more efficient data access and processing by bringing computation closer to where the data resides in memory, thereby eliminating or reducing the need for frequent data movement between different memory locations and simplifies handling consistency. In addition, memory-centric designs enable better data partitioning strategies that minimize the need for cross-partition transactions.

Workload Infrastructure

Moving to a memory-centric system design does not have to make things complicated when reasoning about core infrastructure logic and management of resources. Whereas in a typical CPU-centric system, we have a coordinator and multiple compute nodes, in a memory-centric system the control plane can be placed on the host's root complex or a set of CPU nodes. The control plane's coordinator threads operate on shared system state and metadata, which can be placed on the coherent memory pool for ease of coordination, and to avoid leading to host-congestions [cite, Meta].

Data Pipelines

Databases are the first step in many data intensive tasks (e.g., Recommendation systems, Retrieval Augmented Generation, ML inference, sensor data monitoring) where the database output is the input to one (or more) "consumer" systems (e.g., Tensorflow model inference). The end to end task execution is organized in a data pipeline where each part of the pipeline uses the hardware and software platform suited to the operation it executes.

A memory centric design can enable a shift where CPUs and the accelerators that are increasingly becoming part of such data pipelines are "equidistant" from the data enabling simpler communication and minimizing the performance cliffs observed by data movement or by CPU and accelerators operating with vastly different memory budgets.

Conclusion

The conventional processor-oriented computer design, centered around a CPU, is becoming less attractive due to the slowing of Moore's Law and the increasing costs and constraints of DRAM. As a result, we should shift towards memory-oriented computer designs, where data plays a central role.

Memory-oriented computer designs dissociate processors from data, enabling computation wherever the data resides and minimizing data movement. By placing a greater emphasis on memory and storage components, memory-oriented designs optimize resource utilization and enable more efficient processing of data-intensive workloads. This approach also aligns with the rise of domain-specific architectures (DSAs), particularly for tasks such as Deep Neural Networks (DNNs), which are driving significant advancements in data center computing. For database systems specifically, a memory-centric design offers several advantages, including improved query processing, transaction management, and hybrid transactional-analytical processing. By partitioning query processing operations based on data location, pushing processing closer to the data, and optimizing memory capacity, memory-centric DBMSs can achieve higher performance and efficiency. Additionally, memory-centric designs facilitate the integration of data pipelines, enabling simpler communication and minimizing performance cliffs associated with data movement.

The adoption of memory-centric design principles represents a fundamental shift in how we approach computing architecture, particularly in cloud environments and database systems. By prioritizing memory and data access over traditional compute-centric approaches, memory-oriented designs offer the promise of greater performance, scalability, and efficiency in handling data-intensive workloads in the modern computing landscape.

6 Working Group 3: AI Hardware. What is in it for Cloud DBMSs?

David F. Bacon (Google – New York, US, dfb@google.com)
Holger Fröning (Universität Heidelberg – Mannheim, DE, holger.froening@ziti.uni-heidelberg.de)
Mark D. Hill (University of Wisconsin-Madison, US, markhill@cs.wisc.edu)
Holger Pirk (Imperial College London, GB, pirk@imperial.ac.uk)
Pinar Tözün (IT University of Copenhagen, DK, pito@itu.dk)
Tianzheng Wang (Simon Fraser University – Burnaby, CA, tzwang@sfu.ca)

AI hardware and deep neural networks (DNNs) synergize to advance computational capabilities. Graphics Processing Units (GPUs), originally for graphics rendering, accelerate DNN training and inference with parallel processing. DNNs' demanding computations benefit from GPUs' parallel architecture, driving efficiency and speed.

Similarly, DNN demands drive the design of specialized AI hardware such as Google's TPU [18], Intel's Gaudi [17] processor, and various other examples from industry and academia [5, 7, 3, 8, 6, 1, 11, 13, 30]. As DNN complexity grows, so does the demand for more powerful processors, spurring advancements in processor design. In turn, processor evolution fuels DNN innovation, enabling breakthroughs in computer vision, natural language processing, and autonomous systems. This reciprocal relationship propels technological progress, shaping the future of AI and computational sciences.

AI Hardware and Workloads

Given the synergy between AI hardware and DNNs, we make two main observations that mutually fuel each other. Firstly, the computational rule behind DNNs – including but not limited to convolutional layers, attention modules, and linear layers – is to a large extent dominated by the dot product operation. Seen as a set of operations that shares input operands, dot product operations exhibit high computational/arithmetic intensity i, calculated as the number of computations executed per byte fetched from memory. Secondly, analyzing the scaling trends reveals that compute performance scales much better than memory performance. In more detail, the ratio r of compute performance in operations per second and memory performance in bytes per second continues to grow. Notably, these two observations complement each other, as achieving peak performance on CMOS-based processors requires an increasing computational intensity i.

Furthermore, a growing ratio implies that overall execution costs are increasingly dominated by data movements such as fetching data from memory to the processor. In contrast, the contribution of the number of computations following such a data fetch to overall costs is becoming insignificant. Fundamentally, this suggests that it is unpromising to design algorithms based on the number of operations as it happens in classical Big O analysis. Instead, it is highly promising to revisit algorithmic alternatives that have no longer been pursued due to high computational costs.

Last, the energy (non-)proportionality of processors suggests that it is unpromising not to run a processor at peak utilization, as only then the best power efficiency can be achieved. As peak utilization requires utilizing all components, including compute and memory, such a situation is only achievable for a high ratio r. This in combination with other effects, such as large electrical surges when power variations occur, also indicates that computationally intensive workloads are desirable.

From Data-Intensive to Compute-Intensive

World-wide data production is increasing very quickly, and the amount of data stored by databases is increasing rapidly as well. For instance, Google's Spanner has been roughly doubling in size every year, expecting to reach a zettabyte by 2030 as mentioned earlier. While compute and I/O costs for data continue to rise rapidly, they are not rising nearly as fast as the demand for storage. This means that on a per-byte basis, data is becoming colder.

Databases have historically been I/O-dominated. They retrieve large amounts of data and do relatively small amounts of computation per byte. Increases in both storage volume and I/O cost have motivated more and more use of data compression, as one way of trading space for time.

These database trends are on a collision course with the hardware trends described in the previous section. The one resource that is becoming cheaper is specialized computation (in the form of GPUs and TPUs). Thus far we have not found a way to exploit this form of computation. Some work has explored how to process data on TPUs [14], however, as the amount of data grows exponentially we still lack approaches to managing data storage cost, and bringing substantially more data from storage to the computing resources still presents a major bottleneck.

Our fundamental thesis is therefore *we must switch the balance in database systems from being data-intensive to being compute-intensive*. If we can trade space for time, and move the compute time into specialized hardware (and in particular, specialized hardware being deployed for AI), then we can bring database growth back in line with technology growth.

In the following, we explore two basic approaches to solving this problem:

- Developing new forms of data compression that takes advantage of various properties of large language models (LLMs) [27, 29, 9, 4] and their use in emerging applications.
- Using more computationally expensive algorithms for database operations, and moving them into TPUs and GPUs.

Learned Compression

Compression is typically divided into lossless and lossy compression. We propose a third category, *learned compression*, which uses LLMs to compress data.

The fundamental observation is that much of the data growth is being driven by the storage of generated data, using prompts to LLMs. This presents an opportunity for databases to reduce their storage footprint by storing prompts rather than storing the "expansion" and re-generating the expansion on demand. This can lead to order-of-magnitude reduction in data storage and data transfer costs, at the expense of much more compute-intensive data retrieval. But that retrieval has now been moved to the GPU/TPU, where we can ride the commodity curve of capability and capacity.

Assuming the source data used to generate an LLM-based response is already in the database, and that we store a pointer to the input data, we describe three forms of learned compression:

- Use the LLM as a data source.
- Use the LLM to generate a summary of the information that is stored.
- Use the LLM to produce an approximation of the original input when it is retrieved.

Regenerating Auto-generated Information. A use of LLMs that has very rapidly entered widespread commercial use is automatic generation of text inside of productivity applications. For example, Gmail plugins can generate responses using GPT-4. In many cases, users simply accept the generated response and send it. The generated email is then stored in the "Sent" folder. However, we observe that the response can be encoded far more compactly as a triple consisting of (1) a "pointer" to the message to which the reply was generated (e.g. a message ID), (2) an identifier for the model that was used to generate it (e.g. "GPT-4"), and (3) the random seed used for the generative operation. Upon retrieval, this triple is combined with the original message and fed into the model, which should re-generate the identical response text.

The same approach can be applied for applications like auto-generated document summaries, translations and so on.

Delta Compression. It is quite common for the auto-generated email text to be modified by the user, and frequently those changes are small relative to the total size of the text. We can augment the technique above by storing a delta against the generated text. If the delta gets too large, we simply revert to storing the text itself.

Other Modalities. Images, audio, and video represent even larger opportunities for compression. In some cases this will be many orders of magnitude. One example would be *Generate an MP3 audio file saying "I'll Be Back" using Donald Duck's voice.* The database then only needs to store such prompt along with the random seed and model information, instead of full MP3 audio file copies.

Generating Column Data from Models. Another way of turning data-intensive operations into more compute-intensive ones is rather than storing all the data in a database to answer queries, we let models to either directly answer certain questions or generate materialized views of data for further data processing.

Even though it may be imprecise, LLMs have been good at answering common knowledge questions. For example, the answers to which or how many books are written by an author or the movies an actor stars are such questions. If we are to use a SQL query over a database to answer these questions, the data has to be loaded into a database with a predefined schema, such as:

```
-- A traditional SQL table:
CREATE TABLE Actors (
Actor STRING NOT NULL
Movie STRING NOT NULL
) PRIMARY KEY (Actor);
```

This results in higher space consumption either in storage or memory compared to keeping raw/unstructured data because of the explicit schema.

Rather than explicitly storing the data in a database with a schema, today we can effectively pose these questions as prompts to a LLM. Depending on the questions, an LLM deployed on a GPU or TPU can either answer it directly or generate a column or a table that can later be used by more traditional database operators. For example, we may turn the above predefined schema into the following form:

```
-- A SQL table generated from a model:
CREATE TABLE Movies (
Actor STRING NOT NULL,
Movie STRING NOT NULL AS (GENERATE FROM Gemini3.0
WITH "What movies featured the actor [Actor]?")
) PRIMARY KEY (Actor);
```

Also, a COUNT * query can be answered by the model without generating a table. If we expect follow-up questions after the initial prompt, it could be better to generate a table and materialize it. For example, a search of the movies an actor starred in may trigger a follow up question on keywords on the movie title. The materialized tables can be cached in accelerator memory if there is space or stored in a fast persistent storage medium (e.g., NVMe SSDs) for later reuse by other queries.

Re-inflation of Summarized Data. LLMs are also heavily used for both summarizing documents and generating longer documents given a summary. One can use these features to store the summarized versions of documents closer to the compute nodes with accelerators. Storing the summaries instead of the whole documents reduces the overall data size and makes caching of this data more feasible. When someone asks for the contents of the whole document, an LLM deployed on a hardware accelerator such as GPU or TPU can re-inflate this document using its summary.

Revisiting Compute-intensive Problems in Data Processing

Analytical query processors need to be conscious of the bottleneck shifting from compute to data access. Fortunately, there is precedent for such shifts, such as the move from sequential to parallel processing. To take that into account, we believe that algorithms that have been considered too compute-intensive in the past should be reinvestigated to determine if they are more competitive given the new balance in hardware. There are past examples that can serve as inspiration: For aggregation, sequential scans are optimal on sequential processors, while massively parallel prefix scans perform optimally on GPUs. Worst-case optimal join algorithms reduce the need for large intermediate result materialization at the cost of highly CPU-inefficient control paths. Neural networks were considered too expensive to evaluate upon their invention (in the 1960s) efficiently but received a boost in popularity when massively parallel processors became available. Another example from scientific computing is iterative methods used to solve systems of linear equations. While they share a common objective of minimal time, they differ in how they update the solution at each iteration. In the classical Gauss-Seidel method which was designed with sequential processing in mind, the updated value of one element is immediately used in the calculation of the next element. In the Jacobi method, in contrast, all components of the solution vector are updated simultaneously using the values from the previous iteration. In practice, this means that Gauss-Seidel converges faster with regard to the number of iterations. However, due to the algorithm's sequential nature, the execution is not in line with parallel processors. Given that all processors are highly parallel, it is thus much more promising to use the Jacobi method instead, even though it is work-inefficient with regard to the number of iterations.

We believe that more such "newly-relevant" techniques can be discovered for data-intensive operations such as relational analytics, graph processing and even transaction processing.

Non-Traditional Workloads

Finally, the increasing diversity of data management workloads creates challenges beyond traditional analytics and transaction processing: workloads such as spatial data processing, data cleaning or data integration will likely benefit from AI-supporting hardware. Developing creative solutions to map such workloads to the new hardware is an exciting research opportunity. Emerging domains such as vector databases are also a natural fit for AI-focused hardware.

7 Working Group 4: Incrementally Distributed

Nandita Vijaykumar (University of Toronto, CA, nandita@cs.toronto.edu),
Zsolt István (TU Darmstadt, DE, zsolt.istvan@tu-darmstadt.de),
Tilmann Rabl (HPI Potsdam, DE, Tilmann.Rabl@hpi.de),
Alexander Boehm (SAP HANA, DE, alexander.boehm@sap.com),
Margo Seltzer (University of British Columbia – Vancouver, CA, mseltzer@cs.ubc.ca)

Today's database landscape renders as a black and white image in which database systems are either single-node systems or distributed systems. Generally, the distributed systems offer scalability, but pay a penalty in terms of single-node performance. But single-node systems face obvious scalability challenges. The ideal would be a system with the simplicity and ease of operation of a single-node system capable of infinite scaling.

In lieu of being able to achieve our ideal, we currently provide two different solutions. First, we have small-scale distributed systems, such as Oracle RAC [26] and SAP HANA [12], that typically scale only to a double-digit number of geographically co-located machines. Second, cloud storage systems enable single-node systems to host databases larger than the storage available locally on a single node. Both of these architectures are point solutions that do not fully cover the range of possibilities between single-node and cloud-scale databases.

Resource disaggregation provides a logically centralized abstraction for a physically distributed reality, managed by infrastructure providers. We already make this a reality for persistent storage: cloud-scale storage systems provide practically infinite capacity from a single node. Emerging interconnect technology, such as Compute Express Link (CXL) [25], makes it possible for compute nodes to directly access more memory than is possible in a purely local configuration. In other words, in the same way that cloud storage systems allow for disaggregated storage; CXL allows for disaggregated memory. In addition to providing a simpler abstraction on which to build databases, such disaggregation solves problems for cloud providers: it makes use of stranded resources, thereby increasing cloud provider efficiency. This trend towards disaggregation introduces exciting, new software architectures.

We present a taxonomy that provides a framework in which to consider disaggregation more broadly. We ask what it might mean to disaggregate any combination of CPU, memory, network connectivity, storage, and custom accelerators. We consider different combinations of resource disaggregation and how such new configurations influence database management systems.

Considering disaggregation more broadly lets us optimize systems for different metrics. Rather than focusing on maximizing queries-per-second (QPS), we might ask for an architecture that maximizes QPS/core, QPS/byte, QPS/joule, etc. Alternately, these considerations allow us to consider software architectures that are impractical today, e.g., single-node databases with access to petabytes of main memory or systems with new forms of elasticity in memory, network bandwidth, or access to hardware acceleration. In other words, the golden age of computer architecture [15] should be enabling a new golden age in database system design.

All resources become disaggregated

In the past, scalable database architectures have introduced disaggregation for storage systems such as Oracle RAC that use dedicated and separately scalable storage networks. This separation of storage and compute improves adaptability and independent scalability of resources, enables better sizing decisions, and avoids underutilization. Meanwhile, disaggregated storage has become the standard way of persisting data in cloud environments. This allows even systems that were not designed with disaggregated storage in mind to benefit from features such as virtually infinitely scalable, highly available, and dynamically growing storage that looks like a traditional block device.

Today, DRAM contributes a significant fraction of the cost of data centers [21]. This memory is utilized differently and dynamically depending on the applications and operators running in the system. Database engine designers have started addressing this problem by sharing memory across nodes. Today, memory sharing is selectively used for specific operations, such as data shuffling in Google BigQuery [23], or to temporarily "borrow" memory from remote nodes using RDMA to avoid spilling data to disk in situations where not enough local memory is available [21]. However, there is no standard and easy-to-use way for cross-node memory sharing. Technologies such as CXL provide a new opportunity as they enable transparent memory capacity sharing within racks, with limited, but noticeable, overhead over local memory. This enables gracefully scaling memory requirements beyond single node capacity by utilizing either specialized memory instances or neighboring nodes' memory.

As a next step, after memory and storage are disaggregated, one can also think about disaggregating other resources, in particular, network and accelerators. Given increasing network bandwidth and varying communication patterns in applications, it is likely that systems frequently fail to utilize all their local network capacity. The NICs in some nodes will be overutilized while others are idle, similarly to the situation with memory today. If nodes are also interconnected with other technologies such as CXL, we see the potential of disaggregating networking, by sharing NICs across nodes. This enables new heterogeneous setups of network topologies, increasing the total bandwidth available, and improving utilization. Analogously, disaggregating accelerators makes it possible to share them for certain workloads rather than requiring a homogeneous overprovisioning on every node or requiring specialized node configurations. A fully disaggregated system will be able to efficiently use local and remote resources, scale each resource individually based on demand from a single system perspective, and optimize deployments from a multi-tenant perspective.

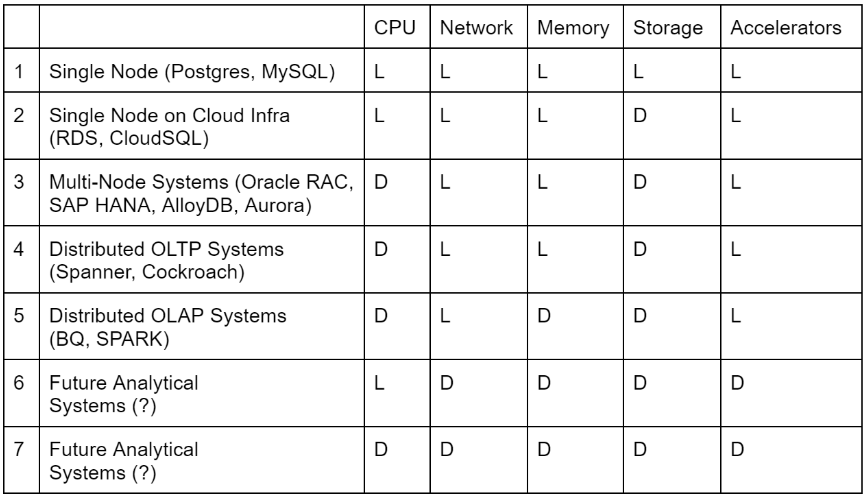
Database Taxonomy of Disaggregation

We claim that disaggregation will enable DBMS that can be incrementally distributed over time, regardless of whether they were initially designed as single-node systems with node-local resources or as distributed systems, where resources are spread out in a cluster.

Table 1 below gives an overview of the resource allocation for various database system designs and deployments. Node-local resources are depicted as an "L", and disaggregated/distributed resources are denoted by "D".

The classical single-node databases are built with the assumption that all resources are node-local (Line 1 in Table 1). By deploying them in a cloud-environment where storage is already disaggregated, but still offered as a block-device abstraction (e.g., AWS Elastic Block Storage (EBS), Google Persistent Disk (PD)), these systems implicitly adopt a distributed storage layer (Line 2). This brings them closer to multi-node DBMS such as Oracle RAC or Google AlloyDB (Line 3), which were initially designed for (small) clusters of machines leveraging a shared storage pool that enables data sharing across nodes. While Google's Spanner system (Line 4) is architected as a globally distributed system that significantly scales beyond the capabilities of multi-node DBMS, its use of disaggregated hardware is still similar to "classical" scale-out database deployments.

Table 1 Overview of the resource allocation for various database system designs and deployments.



		CPU	Network	Memory	Storage	Accelerators
1	Single Node (Postgres, MySQL)	L	L	L	L	L
2	Single Node on Cloud Infra (RDS, CloudSQL)	L	L	L	D	L
3	Multi-Node Systems (Oracle RAC, SAP HANA, AlloyDB, Aurora)	D	L	L	D	L
4	Distributed OLTP Systems (Spanner, Cockroach)	D	L	L	D	L
5	Distributed OLAP Systems (BQ, SPARK)	D	L	D	D	L
6	Future Analytical Systems (?)	L	D	D	D	D
7	Future Analytical Systems (?)	D	D	D	D	D

Today, there are only a few large-scale, distributed systems that are designed to make use of disaggregated memory. A prominent example is Google BigQuery [23] (Line 5), which uses a shared memory pool for data shuffling. With the proliferation of far memory, provided via CXL, existing multi-node systems can easily evolve in this direction, by getting (some of) their memory from remote CXL devices instead of local DRAM.

We envision that other resources such as network and accelerator cards will be disaggregated in the future, leading to potential new analytical systems that use communication channels other than Ethernet or RDMA for inter-node communication, thus allowing for the shared use of network and accelerator cards.

With the ever increasing trend of hardware disaggregation, we claim that all DBMS will eventually run on disaggregated storage, memory, and even make use of other disaggregated resources such as networking or accelerators. This is independent of whether the systems were initially designed for such a hardware configuration or not. Thus, database architects will need to adapt existing system architectures to the different hardware characteristics (e.g., increased latency, more capacity, changes in bandwidth) and leverage potential opportunities that come with disaggregation (e.g., increased elasticity, opportunities for data sharing).

Open Challenges & Opportunities

For database system architecture, disaggregation brings many opportunities. We see three steps toward DBMSs embracing a disaggregated future.

The low hanging fruit is to use disaggregated resources like local resources to expand single node systems, essentially, ignoring disaggregation. The prototypical example is memory expansion over CXL, which enables a system to extend local memory with slightly longer-latency far memory without changing the architecture. To avoid running into performance cliffs when crossing the single node boundary, awareness for local and far memory is beneficial. Otherwise, systems will continue to function as designed, but they may be notably slower. Without coping with disaggregation and optimizing across different workloads and database deployments, this will improve scalability but not utilization.

A next level is a cluster level view on the database and application deployments, which enables improving resource utilization by sharing resources across nodes with different requirements. Besides static, but disaggregated, mapping of resources, dynamic assignment can further improve utilization and reduce cost and energy USge.

When fully embracing disaggregation, it is possible to fluidly scale across resources, also scaling down to fractions of nodes. This would enable more fine-grain control in resource offerings from cloud providers, carbon efficiency from being able to consolidate and turn off unused pools of resources, and more flexibility in constructing hardware architectures for any database application. Being aware of hardware heterogeneity in disaggregated setups also enables efficient compute and data placement, new data structure designs, and innovative communication-less data exchange.

Carbon and Economic Efficiency

Data centers incur costs from provisioning sufficient compute resources (e.g., memory, CPU, storage) separately from provisioning power and cooling for these resources. Replacing and upgrading components further contribute to cost. In addition to direct economic costs, data centers create a significant carbon footprint from both power consumption and embodied carbon due to chip manufacturing for acquiring new components. As hardware technologies and applications evolve at a rapid pace, there is an increasing requirement for data centers to refresh older components.

Disaggregation offers the opportunity to deliver cost savings to both cloud providers and users, while enabling a reduction in carbon footprint. The reduction in carbon footprint via disaggregation can be accomplished in several ways. First, disaggregation can potentially reduce the overall requirements for hardware components by reducing resource stranding. The major benefits of disaggregation are in enabling better utilization of resources in the data center by addressing the bin packing problem, enabling flexible resource sharing, and more efficient resource utilization, potentially necessitating fewer resources overall. Second, disaggregation enables the use of older, lower performance components opportunistically in addition to newer higher performance components. This can potentially be done without sacrificing performance with novel system-level techniques to address performance differences. Providing different pricing points for the use of data center hardware can also motivate the use of older components. Third, disaggregation can enable flexibly turning off power to large pools of hardware resources in data centers during periods of low utilization. During these periods, all utilization of compute resources can be redirected to fewer pools of disaggregated resources and has the potential to significantly reduce the overall power consumption in data centers.

Economic Model: Elastic and Fine-grained Allocation

The initial promise of the cloud was elastic and fine-grained resource allocation. In today's cloud resource model, clients typically pick between pre-defined "bundles" of resources: even though there are plenty of different instance types in the cloud offering, these all link together CPUs, memory, network, and storage. In a disaggregated future, with resources "unbundled", clients can express their needs more specifically, in terms of CPU cores, memory, storage capacity, network bandwidth, etc., in smaller increments than what instance types allow for today. This latter way of requesting, allocating, and billing for resources is the fulfillment of the initial promise of clouds on granularity.

In terms of elasticity of resources, even though today we can scale out at VM increments. Within a single VM, storage capacity and bandwidth can be elastic, but memory and CPU allocations are static. In a disaggregated future we will have elasticity of CPUs, memory, and network bandwidth even within a single VM. This is an important improvement for single node databases, which will be able to allocate an expected amount of memory but have the ability to temporarily use more main memory, in case they face workload spikes.

A beneficial side-effect of allocating resources for applications in a more fine-grained and elastic manner is that it will bring to light the hidden "divisor" of performance metrics. Today, we reason in Queries Per Second (QPS), hiding the fact that in most workloads we should consider QPS/core, QPS/GB of memory, or QPS/aggregate power of the resources we run on. By being able to expose the divisor, scalability bottlenecks can be easier to identify and requirements can be more transparently communicated between users and service providers.

In terms of the pricing model of disaggregated resources, we believe that those cloud providers (existing or new) that will find an economic model that passes on the savings resulting from the more efficient use of hardware to their customers will be at a significant advantage over their competitors who stick to rigid instance types. For customers, once they can allocate and pay only for the resources they need, there will be little incentive to stick to the old model.

Call to Action

Disaggregation is happening. As database designers, we have three choices: we can ignore it, cope with it, or embrace it. In reality, we have no choice, we must adapt systems to work in a disaggregated world. Coping with disaggregation means redesigning our systems to effectively use multiple memory tiers. Embracing disaggregation means designing next generation architectures that seamlessly grow from the fastest single-node system to a fully, cloud-scale system. In designing these systems, we should demand that hardware designers and cloud providers give us the mechanisms we need. If a small amount of cache coherent memory is game-changing for databases, then we need to convince the hyperscalers to make it available; we should demand elasticity in all resources.

We should embrace the entire spectrum from single-node to cloud-scale growth. Cloud providers should provide seamless elasticity in every dimension: CPU, storage, memory, network capacity, and hardware acceleration (including GPUs). Database providers should design systems that take full advantage of this elasticity. Together we can enable customers to optimize for metrics other than pure latency and bandwidth – customers might optimize for queries per second (QPS), QPS/core, QPS/byte, or QPS/watt.

References

1 Aayush Ankit, Izzat El Hajj, Sai Rahul Chalamalasetti, Geoffrey Ndu, Martin Foltin, R Stanley Williams, Paolo Faraboschi, Wen-mei W Hwu, John Paul Strachan, Kaushik Roy, et al. Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference. In *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*, pages 715–731, 2019.

2 Nikos Armenatzoglou, Sanuj Basu, Naga Bhanoori, Mengchu Cai, Naresh Chainani, Kiran Chinta, Venkatraman Govindaraju, Todd J. Green, Monish Gupta, Sebastian Hillig, Eric Hotinger, Yan Leshinksy, Jintian Liang, Michael McCreedy, Fabian Nagel, Ippokratis Pandis, Panos Parchas, Rahul Pathak, Orestis Polychroniou, Foyzur Rahman, Gaurav Saxena, Gokul Soundararajan, Sriram Subramanian, and Doug Terry. Amazon redshift re-invented. In *SIGMOD*, pages 2205–2217. ACM, 2022.

3 Eunjin Baek, Dongup Kwon, and Jangwoo Kim. A multi-neural network acceleration architecture. In *Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture*, ISCA '20, page 940 – 953. IEEE Press, 2020.

4 Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, US, 2020. Curran Associates Inc.

5 Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam. Diannao: a small-footprint high-throughput accelerator for ubiquitous machine-learning. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '14, page 269 – 284, New York, NY, US, 2014. Association for Computing Machinery.

6 Yu-Hsin Chen, Joel Emer, and Vivienne Sze. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. *ACM SIGARCH computer architecture news*, 44(3):367–379, 2016.

7 Yunji Chen, Tao Luo, Shaoli Liu, Shijin Zhang, Liqiang He, Jia Wang, Ling Li, Tianshi Chen, Zhiwei Xu, Ninghui Sun, and Olivier Temam. Dadiannao: A machine-learning supercomputer. In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 609–622, 2014.

8 Ping Chi, Shuangchen Li, Cong Xu, Tao Zhang, Jishen Zhao, Yongpan Liu, Yu Wang, and Yuan Xie. Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory. *ACM SIGARCH Computer Architecture News*, 44(3):27–39, 2016.

9 Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sashank Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24(1), mar 2024.

10 Benoît Dageville, Thierry Cruanes, Marcin Zukowski, Vadim Antonov, Artin Avanes, Jon Bock, Jonathan Claybaugh, Daniel Engovatov, Martin Hentschel, Jiansheng Huang, Allison W. Lee, Ashish Motivala, Abdul Q. Munir, Steven Pelley, Peter Povinec, Greg Rahn, Spyridon Triantafyllis, and Philipp Unterbrunner. The snowflake elastic data warehouse. In *SIGMOD*, pages 215–226, 2016.

11 Renhao Fan, Yikai Cui, Qilin Chen, Mingyu Wang, Youhui Zhang, Weimin Zheng, and Zhaolin Li. Maicc: A lightweight many-core architecture with in-cache computing for multi-dnn parallel inference. In *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '23, page 411 – 423, New York, NY, US, 2023. Association for Computing Machinery.

12 Franz Färber, Norman May, Wolfgang Lehner, Philipp Große, Ingo Müller, Hannes Rauhe, and Jonathan Dees. The sap hana database–an architecture overview. *IEEE Data Eng. Bull.*, 35(1):28–33, 2012.

13 Soroush Ghodrati, Sean Kinzer, Hanyang Xu, Rohan Mahapatra, Yoonsung Kim, Byung Hoon Ahn, Dong Kai Wang, Lavanya Karthikeyan, Amir Yazdanbakhsh, Jongse Park, Nam Sung Kim, and Hadi Esmaeilzadeh. Tandem processor: Grappling with emerging operators in neural networks. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, page 1165 – 1182, New York, NY, US, 2024. Association for Computing Machinery.

14 Dong He, Supun C Nakandala, Dalitso Banda, Rathijit Sen, Karla Saur, Kwanghyun Park, Carlo Curino, Jesús Camacho-Rodríguez, Konstantinos Karanasos, and Matteo Interlandi. Query processing on tensor computation runtimes. *Proc. VLDB Endow.*, 15(11):2811 – 2825, jul 2022.

15 John L. Hennessy and David A. Patterson. A new golden age for computer architecture. *Commun. ACM*, 62(2):48 – 60, jan 2019.

16 Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Understanding predictions with data and data with predictions. In *ICML*, volume 162, pages 9525–9587, 2022.

17 Intel Corporation. Intel gaudi ai accelerators, 2024.

18 Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Clifford Young, Xiang Zhou, Zongwei Zhou, and David A Patterson. Tpu v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ISCA '23, New York, NY, US, 2023. Association for Computing Machinery.

19 Justin Levandoski, Garrett Casto, Mingge Deng, Rushabh Desai, Pavan Edara, Thibaud Hottelier, Amir Hormati, Anoop Johnson, Jeff Johnson, Dawid Kurzyniec, Sam McVeety, Prem Ramanathan, Gaurav Saxena, Vidya Shanmugam, and Yuri Volobuev. Biglake: Bigquery's evolution toward a multi-cloud lakehouse. In *SIGMOD*, 2024.

20 Huaicheng Li, Daniel S. Berger, Lisa Hsu, Daniel Ernst, Pantea Zardoshti, Stanko Novakovic, Monish Shah, Samir Rajadnya, Scott Lee, Ishwar Agarwal, Mark D. Hill, Marcus Fontoura, and Ricardo Bianchini. Pond: Cxl-based memory pooling systems for cloud platforms. In Tor M. Aamodt, Natalie D. Enright Jerger, and Michael M. Swift, editors, *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, ASPLOS 2023, Vancouver, BC, CA, March 25-29, 2023*, pages 574–587. ACM, 2023.

21 Huaicheng Li, Daniel S. Berger, Lisa Hsu, Daniel Ernst, Pantea Zardoshti, Stanko Novakovic, Monish Shah, Samir Rajadnya, Scott Lee, Ishwar Agarwal, Mark D. Hill, Marcus Fontoura, and Ricardo Bianchini. Pond: Cxl-based memory pooling systems for cloud platforms. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS 2023, page 574 – 587, New York, NY, US, 2023. Association for Computing Machinery.

22 Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, Theo Vassilakis, Hossein Ahmadi, Dan Delorey, Slava Min, Mosha Pasumansky, and Jeff Shute. Dremel: A decade of interactive SQL analysis at web scale. *PVLDB*, 13(12):3461–3472, 2020.

23 Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, Theo Vassilakis, Hossein Ahmadi, Dan Delorey, Slava Min, Mosha Pasumansky, and Jeff Shute. Dremel: a decade of interactive sql analysis at web scale. *Proc. VLDB Endow.*, 13(12):3461 – 3472, aug 2020.

24 Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: attributing model behavior at scale. In *ICML*, volume 202, pages 27074–27113, 2023.

25 Debendra Das Sharma, Robert Blankenship, and Daniel S. Berger. An introduction to the compute express link (cxl) interconnect, 2024.

26 Steve Shaw and Martin Bach. *RAC Architecture*, pages 63–95. Apress, Berkeley, CA, 2010.

27 Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

28 Thomas N. Theis and H.-S. Philip Wong. The end of moore's law: A new beginning for information technology. *Comput. Sci. Eng.*, 19(2):41–50, 2017.

29 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

30 Swagath Venkataramani, Vijayalakshmi Srinivasan, Wei Wang, Sanchari Sen, Jintao Zhang, Ankur Agrawal, Monodeep Kar, Shubham Jain, Alberto Mannari, Hoang Tran, Yulong Li, Eri Ogawa, Kazuaki Ishizaki, Hiroshi Inoue, Marcel Schaal, Mauricio Serrano, Jungwook Choi, Xiao Sun, Naigang Wang, Chia-Yu Chen, Allison Allain, James Bonano, Nianzheng Cao, Robert Casatuta, Matthew Cohen, Bruce Fleischer, Michael Guillorn, Howard Haynie, Jinwook Jung, Mingu Kang, Kyu-hyoun Kim, Siyu Koswatta, Saekyu Lee, Martin Lutz, Silvia Mueller, Jinwook Oh, Ashish Ranjan, Zhibin Ren, Scot Rider, Kerstin Schelm, Michael Scheuermann, Joel Silberman, Jie Yang, Vidhi Zalani, Xin Zhang, Ching Zhou, Matt Ziegler, Vinay Shah, Moriyoshi Ohara, Pong-Fei Lu, Brian Curran, Sunil Shukla, Leland Chang, and Kailash Gopalakrishnan. Rapid: Ai accelerator for ultra-low precision training and inference. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 153–166, 2021.

Participants

- Anastasia Ailamaki
EPFL – Lausanne, CH
- Gustavo Alonso
ETH Zürich, CH
- David F. Bacon
Google – New York, US
- Lawrence Benson
TU München, DE
- Carsten Binnig
TU Darmstadt, DE
- Alexander Böhm
SAP SE – Walldorf, DE
- Helena Caminal
Google – Sunnyvale, US
- Yannis Chronis
Google – Sunnyvale, US
- Holger Fröning
Universität Heidelberg –
Mannheim, DE
- Jana Giceva
TU München – Garching, DE
- Mark D. Hill
University of Wisconsin-
Madison, US
- Ihab Francis Ilyas
University of Waterloo, CA
- Zsolt Istvan
TU Darmstadt, DE
- Lana Josipovic
ETH Zürich, CH
- Tim Kraska
MIT – Cambridge, US
- Justin Levandoski
Google – Seattle, US
- Jignesh M. Patel
Carnegie Mellon University –
Pittsburgh, US
- David A. Patterson
University of California –
Berkeley, US
- Holger Pirk
Imperial College London, GB
- Tilmann Rabl
Hasso-Plattner-Institut,
Universität Potsdam, DE
- Eric Sedlar
Oracle Labs –
Redwood Shores, US
- Margo Seltzer
University of British Columbia –
Vancouver, CA
- Pinar Tözün
IT University of
Copenhagen, DK
- Nandita Vijaykumar
University of Toronto, CA
- Tianzheng Wang
Simon Fraser University –
Burnaby, CA
- Lisa Wu Wills
Duke University – Durham, US
- Tobias Ziegler
TU Darmstadt, DE

Report from Dagstuhl Seminar 24171

Automated Synthesis: Functional, Reactive and Beyond

S. Akshay*[1], Bernd Finkbeiner*[2], Kuldeep S. Meel*[3], Ruzica Piskac*[4], and Arijit Shaw†[5]

1 **Indian Institute of Technology Bombay – Mumbai, IN.** `akshayss@cse.iitb.ac.in`
2 **CISPA – Saarbrücken, DE.** `finkbeiner@cispa.de`
3 **University of Toronto, CA.** `meel@comp.nus.edu.sg`
4 **Yale University – New Haven, US.** `ruzica.piskac@yale.edu`
5 **Chennai Mathematical Institute, IN & University of Toronto, CA.** `if.arijit@gmail.com`

Abstract

This report summarizes the program of Dagstuhl Seminar 24171 on “Automated Synthesis: Functional, Reactive and Beyond”. The seminar brought together researchers working on different aspects of functional synthesis and investigated its relationship with reactive synthesis. Through multiple expository tutorials, diverse technical talks, and multiple open discussion sessions, the seminar crystallized the current challenges for theory and tools in this area and opened fresh directions towards new applications.

Seminar April 21–26, 2024 – https://www.dagstuhl.de/24171
2012 ACM Subject Classification Computing methodologies; Theory of computation → Logic
Keywords and phrases automated synthesis, boolean functions, knowledge representations, reactive synthesis, SAT/SMT solvers
Digital Object Identifier 10.4230/DagRep.14.4.85

1 Executive Summary

S. Akshay (Indian Institute of Technology Bombay – Mumbai, IN)
Bernd Finkbeiner (CISPA – Saarbrücken, DE)
Kuldeep S. Meel (University of Toronto, CA)
Ruzica Piskac (Yale University – New Haven, US)

In Dagstuhl Seminar 24171, we brought together researchers working in various aspects of automated functional synthesis. This diverse topic encompasses areas ranging from Boolean variants to quantified variants, automated reasoning for general theories, program synthesis, and more. One particular focus was on finding synergies between functional and reactive synthesis communities and investigating the deep connections between these two areas.

On the first day, we started with two introductory tutorials: one on Boolean functional synthesis and another on reactive synthesis, setting the agenda for the entire seminar. This was succeeded by technical presentations on definability and dependency in quantified Boolean formulas. The second day included a tutorial on automated reasoning and synthesis, with an emphasis on theories extending beyond Boolean (e.g., SMT), followed by discussions

* Editor / Organizer
† Editorial Assistant / Collector

on quantitative properties. On the third day, we organized a special session with other tool competition organizers to assess the feasibility of a competition or track dedicated to functional synthesis.

The remaining days were filled with diverse technical talks that fell into two categories. The first category included talks that delved deeper into specific aspects of functional synthesis, reactive/LTL synthesis, and specific problems within these fields. The second category introduced new applications or connections, such as quantum applications and functional programming. Discussions during and beyond these talks were further explored in different open and problem sessions. Some of the identified and discussed problems were:

1. How to formalize the Boolean functional synthesis problem at the heart of reactive synthesis? Various problem formulations were discussed, and some benchmarks were created.
2. Can we go beyond Boolean theories and synthesize programs and functions for general SMT? What bottlenecks do we face?
3. How can we find synergy between automated functional synthesis and synthesis using transformers? Specifically, what is the meeting ground between machine learning and inductive program synthesis techniques, functional synthesis, and automated reasoning?
4. Can the successful lens of knowledge representations and compilations for model counting and Boolean functional synthesis be extended to other settings?
5. Can we synthesize quantum circuits from specifications, thus leading to a theory of automated reasoning for quantum systems?
6. Can reactive synthesis over finite traces utilize techniques developed in automated functional synthesis?

These were among the prominent topics discussed, but the list is by no means exhaustive. Several bottlenecks were identified, such as the need for growth within the community developing these tools before establishing a proper competition. Additionally, there was a recognized necessity for broader and more extensive discussions on benchmarks.

Overall, the seminar fostered a collaborative spirit among theoreticians, tool developers, and experts across different aspects of automated functional synthesis. The seminar was also attended by a large number of early career researchers, postdoctoral fellows, and graduate students who also participated enthusiastically throughout the seminar. The shared optimism generated during this seminar has laid a strong foundation for future advancements. We advocate for the continuation of these valuable discussions and propose organizing further meetings of a similar nature to build on the momentum gained and to explore new frontiers in automated functional synthesis.

In the remainder of this report, we provide the abstracts of all the talks, as well as discussion sessions held during the seminar. We thank all the speakers and attendees for their active participation and look forward to attending and organizing more such events in the future.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

Panel discussions

3 Overview of Talks

3.1 To Assume, Or Not To Assume

Ashwani Anand (MPI-SWS – Kaiserslautern, DE)

Joint work of Ashwani Anand, Kaushik Mallik, Satya Prakash Nayak, Anne-Kathrin Schmuck
Main reference Ashwani Anand, Anne-Kathrin Schmuck, Satya Prakash Nayak: "Contract-Based Distributed Logical Controller Synthesis", in Proc. of the 27th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2024, Hong Kong SAR, China, May 14-16, 2024, pp. 11:1–11:11, ACM, 2024.
URL https://doi.org/10.1145/3641513.3650123

Reactive synthesis techniques assume that the environment acts adversarially. However, in many real-life scenarios, the environment might not work antagonistically. We solve the problem of automatically computing a new class of environment assumptions in two-player turn-based finite graph games which characterize an "adequate cooperation" needed from the environment to allow the system player to win [1]. Given an ω-regular winning condition Φ for the system player, we compute an ω-regular assumption Ψ for the environment player, such that (i) every environment strategy compliant with Ψ allows the system to fulfill Φ (sufficiency), (ii) Ψ can be fulfilled by the environment for every strategy of the system (implementability), and (iii) Ψ does not prevent any cooperative strategy choice (permissiveness).

For parity games, which are canonical representations of ω-regular games, we present a polynomial-time algorithm for the symbolic computation of adequately permissive assumptions and show that our algorithm runs faster and produces better assumptions than existing approaches – both theoretically and empirically. To the best of our knowledge, for ω-regular games, we provide the first algorithm to compute sufficient and implementable environment assumptions that are also permissive.

In the second part of the talk, we apply the lessons learned to strategies computation [2], and negotiations between multiple agents [3].

References

1 Ashwani Anand, Kaushik Mallik, Satya Prakash Nayak, and Anne-Kathrin Schmuck. "Computing Adequately Permissive Assumptions for Synthesis." In *Tools and Algorithms for the Construction and Analysis of Systems*, edited by Sriram Sankaranarayanan and Natasha Sharygina, 211–228. Cham: Springer Nature Switzerland, 2023.

2 Ashwani Anand, Satya Prakash Nayak, and Anne-Kathrin Schmuck. "Synthesizing Permissive Winning Strategy Templates for Parity Games." In *Computer Aided Verification – 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part I*, edited by Constantin Enea and Akash Lal, 13964:436–458. Lecture Notes in Computer Science. Springer, 2023. https://doi.org/10.1007/978-3-031-37706-8_22.

3 Ashwani Anand, Anne-Kathrin Schmuck, and Satya Prakash Nayak. "Contract-Based Distributed Logical Controller Synthesis." In *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*, 1–11. HSCC '24. New York, NY, USA: Association for Computing Machinery, 2024. https://doi.org/10.1145/3641513.3650123.

3.2 LTLf Model Checking

Suguman Bansal (Georgia Institute of Technology – Atlanta, US)

Joint work of Suguman Bansal, Yong Li, Lucas M. Tabajara, Moshe Y. Vardi, Andrew Wells
Main reference Suguman Bansal, Yong Li, Lucas M. Tabajara, Moshe Y. Vardi, Andrew M. Wells: "Model Checking Strategies from Synthesis over Finite Traces", in Proc. of the Automated Technology for Verification and Analysis – 21st International Symposium, ATVA 2023, Singapore, October 24-27, 2023, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 14215, pp. 227–247, Springer, 2023.
URL https://doi.org/10.1007/978-3-031-45329-8_11

The innovations in reactive synthesis from Linear Temporal Logics over finte traces (LTLf) will be amplified by the ability to verify the correctness of the strategies generated by LTLf synthesis tools. This motivates our work on LTLf model checking. LTLf model checking, however, is not straightforward. The strategies generated by LTLf synthesis may be represented using terminating transducers or non-terminating transducers where executions are of finite-but-unbounded length or infinite length, respectively. For synthesis, there is no evidence that one type of transducer is better than the other since they both demonstrate the same complexity and similar algorithms.

In this work, we show that for model checking, the two types of transducers are fundamentally different. Our central result is that LTLf model checking of non-terminating transducers is exponentially harder than that of terminating transducers. We show that the problems are EXPSPACE-complete and PSPACE-complete, respectively. Hence, considering the feasibility of verification, LTLf synthesis tools should synthesize terminating transducers. This is, to the best of our knowledge, the first evidence to use one transducer over the other in LTLf synthesis.

3.3 Formal XAI via Syntax-Guided Synthesis

Katrine Bjørner (New York University, US)

Joint work of Katrine Bjørner, Samuel Judson, Filip Cano Córdoba, Drew Goldman, Nicholas Shoemaker, Ruzica Piskac, Bettina Könighofer
Main reference Katrine Bjørner, Samuel Judson, Filip Cano Córdoba, Drew Goldman, Nicholas Shoemaker, Ruzica Piskac, Bettina Könighofer: "Formal XAI via Syntax-Guided Synthesis", in Proc. of the Bridging the Gap Between AI and Reality – First International Conference, AISoLA 2023, Crete, Greece, October 23-28, 2023, Proceedings, Lecture Notes in Computer Science, Vol. 14380, pp. 119–137, Springer, 2023.
URL https://doi.org/10.1007/978-3-031-46002-9_7

We propose a novel application of syntax-guided synthesis to find symbolic representations of a model's decision-making process, designed for easy comprehension and validation by humans. Our approach takes input-output samples from complex machine learning models, such as deep neural networks, and automatically derives interpretable mimic programs. A mimic program precisely imitates the behavior of an opaque model over the provided data. We discuss various types of grammars that are well-suited for computing mimic programs for tabular and image input data.

Our experiments demonstrate the potential of the proposed method: we successfully synthesized mimic programs for neural networks trained on the MNIST and the Pima Indians diabetes data sets. All experiments were performed using the SMT-based cvc5 synthesis tool.

3.4 Programming by example for end user tasks and the use of LLMs

José Cambronero (Microsoft – Redmond, US)

Joint work of José Cambronero, Mukul Singh, Gust Verbruggen, Sumit Gulwani, Vu Le

Programming by example (PBE) allows users with little to no formal computation experience to carry out tasks by providing simple demonstrations (e.g. input/output-based examples). In practice, PBE has found considerable industrial uptake, particularly in end-user environments like spreadsheet software (e.g. Microsoft Excel, Google Sheets). In this talk, I'll present a recent project on learning data-dependent formatting rules in Excel from examples. We'll then discuss how PBE in this domain can be extended to also incorporate multimodal specifications, by supporting use of natural language. Using this as a segue into combining symbolic and neural methods, I'll discuss recent work from the field that uses LLMs and may provide ideas for nice collaborations between formal reasoning and popular LLM-based approaches.

3.5 Boolean Functional Synthesis: A Quick Tour

Supratik Chakraborty (Indian Institute of Technology Bombay – Mumbai, IN)

Joint work of S. Akshay, Aman Bansal, Supratik Chakraborty, Priyanka Golia, Kuldeep Meel, Subhajit Roy, Preey Shah, Shetal Shah, Friedrich Slivovsky

Given a Boolean relational specification $\varphi(X, Y)$ over input variables X and output variables Y, Boolean functional synthesis concerns finding Skolem functions F(X) for Y such that $\exists Y \varphi(X, Y)$ is semantically equivalent to $\varphi(X, F(X))$. In this talk, we introduce the problem, survey some earlier results and then take a deeper dive into two solution approaches that have shown promise in recent years. Specifically, we discuss the guess-check-repair paradigm for synthesizing Skolem functions, and also present a knowledge compilation based approach for Boolean functional synthesis. Finally, we conclude with some perspectives on future research in this area. The talk is based on work reported in [1, 2, 3, 4, 5].

References

1 S. Akshay, Supratik Chakraborty, Shubham Goel, Sumith Kulal, Shetal Shah: Boolean Functional Synthesis: Hardness and Practical Algorithms, Formal Methods Syst. Des. 57(1): 53-86 (2021).
2 S. Akshay, S. Chakraborty, S. Shah: Tractable Representations for Boolean Functional Synthesis, Annals of Mathematics and Artificial Intelligence, 1-46, 2023.
3 Preey Shah, Aman Bansal, S. Akshay, Supratik Chakraborty: A Normal Form Characterization for Efficient Boolean Skolem Function Synthesis, LICS 2021: 1-13.
4 Priyanka Golia, Friedrich Slivovsky, Subhajit Roy, Kuldeep S. Meel: Engineering an Efficient Boolean Functional Synthesis Engine, ICCAD 2021: 1-9.
5 Priyanka Golia, Subhajit Roy, Kuldeep S. Meel: Manthan: A Data-Driven Approach for Boolean Function Synthesis, CAV (2) 2020: 611-633.

3.6 Symbolic Fixpoint Techniques for Logical LTL Games

Deepak D'Souza (Indian Institute of Science – Bangalore, IN)

Joint work of Deepak D'Souza, Stanly John Samuel, Komondoor V. Raghavan
Main reference Stanly Samuel, Deepak D'Souza, Raghavan Komondoor: "Symbolic Fixpoint Algorithms for Logical LTL Games", in Proc. of the 38th IEEE/ACM International Conference on Automated Software Engineering, ASE 2023, Luxembourg, September 11-15, 2023, pp. 698–709, IEEE, 2023.
URL https://doi.org/10.1109/ASE56229.2023.00212

We consider the problem of synthesizing strategies in logically-specified infinite-state two-player games with LTL winning conditions. We lift classical fixpoint algorithms for synthesizing strategies in finite-states games, to our setting. Our evaluation of these algorithms show that they compare well with earlier techniques based on template-based logical synthesis and abstraction-refinement, on benchmarks from the literature.

This is joint work with Stanly Samuel and K V Raghavan.

3.7 On the compilation of non-CNF systems of constraints (or, your weekly dose of knowledge compilation)

Alexis de Colnet (TU Wien, AT)

Knowledge compilers often take as inputs a CNF formula and construct an equivalent Boolean circuit with specific properties. Generally, the size of the output circuit increases exponentially. However, for some families of CNF formulas, one can exploit the structure of the formulas to compile them efficiently. In this talk, I first give a general overview of knowledge compilation and of the circuits that knowledge compilers construct. Then, I present results on the compilation of non-CNF inputs. Seeing CNF as systems of constraints, where every constraint is a clause, I explain how positive results on the compilation of CNF with a certain structure can be extended to more general systems of constraints.

3.8 Synthesis of Infinite-State Reactive Systems (and why it needs functional synthesis for theories beyond Boolean)

Rayna Dimitrova (CISPA – Saarbrücken, DE)

Joint work of Philippe Heim, Rayna Dimitrova
Main reference Philippe Heim, Rayna Dimitrova: "Solving Infinite-State Games via Acceleration", Proc. ACM Program. Lang., Vol. 8(POPL), pp. 1696–1726, 2024.
URL https://doi.org/10.1145/3632899

Infinite-state games are a commonly used model for the synthesis of reactive systems with unbounded data domains. Symbolic methods for solving such games need to be able to construct intricate arguments to establish the existence of winning strategies. Furthermore, the synthesis of the resulting reactive system implementations necessitates the use of functional synthesis for theories beyond Boolean. In this talk, I will present a recent symbolic approach for the synthesis of infinite-state reactive systems, called attractor acceleration, which employs ranking arguments to improve the convergence of symbolic game-solving algorithms. I will then discuss the application and the challenges for functional synthesis in this context.

3.9 A Semi-Gentle Introduction to Reactive Synthesis

Rüdiger Ehlers (TU Clausthal, DE)

Reactive synthesis is the process of computing correct-by-construction finite-state controllers from temporal logic specifications. In this tutorial, we have a look at the basic concepts that underlie current reactive synthesis approaches. We discuss the topic on a fairly technical level in order to highlight the connections to functional and Boolean synthesis.

3.10 On Dependent Variables in Reactive Synthesis

Dror Fried (The Open University of Israel – Ra'anana, IL)

Given a Linear Temporal Logic (LTL) formula over input and output variables, reactive synthesis requires us to design a deterministic Mealy machine that gives the values of outputs at every time step for every sequence of inputs, such that the LTL formula is satisfied. In this paper, we investigate the notion of dependent variables in the context of reactive synthesis. Inspired by successful pre-processing techniques in Boolean functional synthesis, we define dependent variables in reactive synthesis as output variables that are uniquely assigned, given an assignment to all other variables and the history so far. We describe an automata-based approach for finding a set of dependent variables. Using this, we show that dependent variables are surprisingly common in reactive synthesis benchmarks. Next, we develop a novel synthesis framework that exploits dependent variables to construct an overall synthesis solution. By implementing this framework using the widely used library Spot, we show that reactive synthesis that exploits dependent variables can solve some problems beyond the reach of existing techniques. Furthermore, we observe that among benchmarks with dependent variables, if the count of non-dependent variables is low (≤ 3 in our experiments), our method outperforms state-of-the-art tools for synthesis.

3.11 Exploring Connections between Automated Reasoning and Synthesis

Mikoláš Janota (Czech Technical University – Prague, CZ)

In this talk we explore the connections between synthesis and automated reasoning techniques. Generally, synthesis, from logic perspective, is formalized as solving a formula of the following form.

$$\exists f\, \forall \mathbf{x}. P[f, \mathbf{x}]$$

where P is a predicate parametrized by a vector of variables $\mathbf{x}$ and an unknown function f. Typically, the (second order) quantifier $\exists f$ it is omitted; in particular in Satisfiability Modulo Theories (SMT), where f functions are implicitly quantified existentially.

Many approaches use solvers in a black-box fashion by assuming a certain *template* for f, such as linear, quadratic etc. [9]. Then, the synthesis problem is formulated as an SMT problem that search as for the parameters (coefficients) of the template. Interestingly, such approach can also be used to search for *all* the possible f [1].

Some approaches integrate more deeply with the solver. A powerful technique is *deskolemization* of f (as inverse of skolemization), which is possible, if f is always applied to the same tuple of arguments everywhere in P. In the literature, such specifications are referred to as *single-invocation properties* [5, 8]. An example of such property would be $\forall x_1 x_2 .\, f(x_1, x_2) > x_1 \wedge f(x_1, x_2) > x_2$, which would be deskolemized as $\forall x_1 x_2 \exists z.\, z > x_1 \wedge z > x_2$.

For deskolemized version of the specification, f can be that synthesized by *quantifier elimination* (QE) if the formula is in a theory that admits QE, such as linear real/integer arithmetic [7, 2], cf. [5, 6]. Alternatively, Reynolds et al. [8] synthesize f by inspecting the SMT refutation (proof). Hozzová et al. [3] synthesize f in the setting of first or logic (FOL), again from the proof, which relies on explicit axiomatization of any theory that may be used.

Specifications going beyond the single-invocation property fragment maybe tackled by embedding the language of possible solutions into the solver as than algebraic datatype [8]. More recent research shows that refutations containing mathematical induction enable synthesizing recursive functions [4].

References

1 Chad E. Brown, Mikoláš Janota, and Mirek Olšák. Symbolic computation for all the fun. In *Satisfiability Checking and Symbolic Computation*, 2024. https://ceur-ws.org/Vol-3717/paper6.pdf.

2 David C Cooper. Theorem proving in arithmetic without multiplication. *Machine intelligence*, 7(91-99):300, 1972.

3 Petra Hozzová, Laura Kovács, Chase Norman, and Andrei Voronkov. Program synthesis in saturation. In *CADE*, 2023.

4 Petra Hozzová, Daneshvar Amrollahi, Márton Hajdu, Laura Kovács, Andrei Voronkov, and Eva Maria Wagner. Synthesis of recursive programs in saturation. In *International Joint Conference on Automated Reasoning IJCAR*, 2024.

5 Swen Jacobs and Viktor Kuncak. Towards complete reasoning about axiomatic specifications. In *Verification, Model Checking, and Abstract Interpretation*, 2011.

6 Viktor Kuncak, Mikaël Mayer, Ruzica Piskac, and Philippe Suter. Software synthesis procedures. *Communications of the ACM*, 55(2):103–111, February 2012.

7 Rüdiger Loos and Volker Weispfenning. Applying linear quantifier elimination. *Comput. J.*, 36(5):450–462, 1993.

8 Andrew Reynolds, Viktor Kuncak, Cesare Tinelli, Clark W. Barrett, and Morgan Deters. Refutation-based synthesis in SMT. *Formal Methods Syst. Des.*, 55(2):73–102, 2019.

9 Saurabh Srivastava, Sumit Gulwani, and Jeffrey S. Foster. Template-based program verification and program synthesis. *Int. J. Softw. Tools Technol. Transf.*, 15(5-6):497–518, 2013.

3.12 Stochastic Boolean Satisfiability: Recent Developments and Connection to Functional Synthesis

Jie-Hong Roland Jiang (National Taiwan University – Taipei, TW)

Joint work of Jie-Hong Roland Jiang, Pei-Wei Chen, Che Cheng, Yu-Wei Fan, Cheng-Han Hsieh, Yu-Ching Huang, Nian-Ze Lee, Yun-Rong Luo, Christoph Scholl, Kuan-Hua Tu, Hao-Ren Wang, Yen-Shi Wang

Stochastic Boolean Satisfiability (SSAT) generalizes quantified Boolean formulas (QBFs) by allowing quantification over random variables. It is often referred to as games against nature and has applications in making decisions or optimizing under uncertainty. This talk will introduce SSAT, its recent developments, and its connection to Boolean functional synthesis.

References

1 Yu-Wei Fan, Jie-Hong R. Jiang: Unifying Decision and Function Queries in Stochastic Boolean Satisfiability, AAAI 2024: 7995-8003.

2 Yun-Rong Luo, Che Cheng, Jie-Hong R. Jiang: A Resolution Proof System for Dependency Stochastic Boolean Satisfiability, J. Autom. Reason. 67(3): 26 (2023).

3 Che Cheng, Jie-Hong R. Jiang: Lifting (D)QBF Preprocessing and Solving Techniques to (D)SSAT, AAAI 2023: 3906-3914.

4 Yu-Wei Fan, Jie-Hong R. Jiang: SharpSSAT: A Witness-Generating Stochastic Boolean Satisfiability Solver, AAAI 2023: 3949-3958.

5 Jie-Hong R. Jiang: Second-Order Quantified Boolean Logic, AAAI 2023: 4007-4015.

6 Cheng-Han Hsieh, Jie-Hong R. Jiang: Encoding Probabilistic Graphical Models into Stochastic Boolean Satisfiability, IJCAI 2022: 1834-1842.

7 Hao-Ren Wang, Kuan-Hua Tu, Jie-Hong R. Jiang, Christoph Scholl: Quantifier Elimination in Stochastic Boolean Satisfiability, SAT 2022: 23:1-23:17.

8 Pei-Wei Chen, Yu-Ching Huang, Jie-Hong R. Jiang: A Sharp Leap from Quantified Boolean Formula to Stochastic Boolean Satisfiability Solving, AAAI 2021: 3697-3706.

9 Nian-Ze Lee, Jie-Hong R. Jiang: Dependency Stochastic Boolean Satisfiability: A Logical Formalism for NEXPTIME Decision Problems with Uncertainty, AAAI 2021: 3877-3885.

10 Nian-Ze Lee, Yen-Shi Wang, Jie-Hong R. Jiang: Solving Exist-Random Quantified Stochastic Boolean Satisfiability via Clause Selection, IJCAI 2018: 1339-1345.

11 Nian-Ze Lee, Yen-Shi Wang, Jie-Hong R. Jiang: Solving Stochastic Boolean Satisfiability under Random-Exist Quantification, IJCAI 2017: 688-694.

3.13 The Unreasonable Effectiveness of Classical Automated Reasoning in Quantum Computing

Alfons Laarman (Leiden University, NL) and Jingyi Mei (Leiden University, NL)

Joint work of Jingyi Mei, Arend-Jan Quist, Alejandro Villoria, Sebastiaan Brand, Dimitrios Thanos, Tim Coopmans, Alfons Laarman
Main reference Jingyi Mei, Tim Coopmans, Marcello M. Bonsangue, Alfons Laarman: "Equivalence Checking of Quantum Circuits by Model Counting", in Proc. of the Automated Reasoning – 12th International Joint Conference, IJCAR 2024, Nancy, France, July 3-6, 2024, Proceedings, Part II, Lecture Notes in Computer Science, Vol. 14740, pp. 401–421, Springer, 2024.
URL https://doi.org/10.1007/978-3-031-63501-4_21

In this talk, we will show that existing classical automated reasoning methods perform exceedingly well for computationally hard problems in quantum computing and physics. In particular, we demonstrate a linear-length #SAT encoding of the simulation and equivalence checking of universal quantum circuits. An implementation of this method, called Quokka#, outcompetes other state-of-the-art approaches using an off-the-shelve #SAT solver that supports negative weights (GPMC). While decision diagrams offer a viable alternative, we unveil their inherent limitations stemming from their inability to represent the prevalent stabilizer states. This limitation is particularly noteworthy considering the efficient classical simulatability of circuits generating such states. To address this constraint, we introduce Local Invertible Map Decision Diagrams (LIMDDs), which offer exponential improvements in succinctness compared to the combination of stabilizer formalism and existing decision diagrams. Finally, we illustrate how these findings hold relevance beyond quantum computing by translating them back to the domain of quantum physics. To achieve this, we build upon Darwiche and Marquis' seminal "knowledge compilation map" approach, by pioneering a knowledge compilation map for quantum information. This map juxtaposes various decision diagrams against tensor networks and Boltzmann machines, two formalisms extensively utilized in physics to address quantum-hard problems such as simulating many-body systems and determining their ground energy. Our results underscore the significant potential of existing automated reasoning methods in both quantum computing and physics domains.

References

1 Coecke, B., Duncan, R.: Interacting Quantum Observables: Categorical Algebra and Diagrammatics. New Journal of Physics 13(4), 043016 (Apr 2011), http://arxiv.org/abs/0906.4725, arXiv:0906.4725 [quant-ph]

2 Vrudhula, S.B.K., Pedram, M., Lai, Y.T.: Edge Valued Binary Decision Diagrams, pp. 109–132. Springer US (1996)

3 Burgholzer, L., Wille, R.: Advanced equivalence checking for quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 40(9), 1810–1824 (2020)

4 Zulehner, A., Wille, R.: Advanced simulation of quantum computations. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38(5), 848–859 (2019)

5 Tafertshofer, P., Pedram, M.: Factored edge-valued binary decision diagrams. Formal Methods in System Design 10(2), 243–270 (1997)

6 Miller, D.M., Thornton, M.A.: QMDD: A decision diagram structure for reversible and quantum circuits. 36th International Symposium on Multiple-Valued Logic (ISMVL'06) pp. 30–30 (2006)

7 Viamontes, G.F., Markov, I.L., Hayes, J.P.: Quantum circuit simulation. Springer Science and Business Media (2009)

8 Mei, J., Coopmans, T., Bonsangue, M., Laarman, A. (2024, July). Equivalence checking of quantum circuits by model counting. In International Joint Conference on Automated Reasoning (pp. 401-421). Cham: Springer Nature Switzerland.

9 Mei, J., Bonsangue, M., Laarman, A. (2024). Simulating Quantum Circuits by Model Counting. to appear in CAV 2024, available as arXiv preprint arXiv:2403.07197.

10 Quist, A.J., Laarman, A.: Optimizing quantum space using spooky pebble games. In: International Conference on Reversible Computation. pp. 134–149. Springer (2023)

11 Thanos, D., Coopmans, T., Laarman, A.: Fast equivalence checking of quantum circuits of Clifford gates. In: Andr´e, ´E., Sun, J. (eds.) Automated Technology for Verification and Analysis. pp. 199–216. Springer Nature Switzerland, Cham (2023)

12 Villoria, A., Basold, H., Laarman, A.: Enriching diagrams with algebraic operations. arXiv preprint arXiv:2310.11288 (2023)

13 Vinkhuijzen, L., Coopmans, T., Elkouss, D., Dunjko, V., Laarman, A.: LIMDD: A decision diagram for simulation of quantum computing including stabilizer states. Quantum 7, 1108 (2023), https://doi.org/10.22331/q-2023-09-11-1108

14 Vinkhuijzen, L., Coopmans, T., Laarman, A.: A knowledge compilation map for quantum information. arXiv preprint arXiv:2401.01322 (2024), https://doi.org/10.48550/arXiv.2401.01322

15 Vinkhuijzen, L., Grurl, T., Hillmich, S., Brand, S., Wille, R., Laarman, A.: Efficient implementation of LIMDDs for quantum circuit simulation. In: International Symposium on Model Checking of Software (SPIN) (2023)

16 Brand, S., Coopmans, T., Laarman, A.: Quantum graph-state synthesis with SAT. Proceedings of the 14th International Workshop on Pragmatics of SAT (2023)

17 Rennela, M., Brand, S., Laarman, A., Dunjko, V.: Hybrid divide-and-conquer approach for tree search algorithms. Quantum 7, 959 (2023)

3.14 Reactive Synthesis modulo Theories using Abstraction Refinement

Benedikt Maderbacher (TU Graz, AT)

Joint work of Benedikt Maderbacher, Roderick Bloem
Main reference Benedikt Maderbacher, Roderick Bloem: "Reactive Synthesis Modulo Theories using Abstraction Refinement", in Proc. of the 22nd Formal Methods in Computer-Aided Design, FMCAD 2022, Trento, Italy, October 17-21, 2022, pp. 315–324, IEEE, 2022.
URL https://doi.org/10.34727/2022/ISBN.978-3-85448-053-2_38

Temporal stream logic modulo theories (TSL-T) is used to specify the behavior of infinite state reactive systems. We present a refinement based synthesis method that works using LTL synthesis and SMT solving. First, a LTL underapproximation is computed and given to a LTL synthesis tool. In case this is unrealizable the created counter-strategy is analyzed for inconsistencies with the theory. New assumptions and predicates are added to the specification to rule out the counter-strategy and the LTL synthesis is run again. If the problem becomes realizable a program statisfying the original specification is extracted.

3.15 Boosting Definability Bipartition Computation using SAT Witnesses

Pierre Marquis (University of Artois/CNRS – Lens, FR)

Joint work of Jean-Marie Lagniez, Pierre Marquis
Main reference Jean-Marie Lagniez, Pierre Marquis: "Boosting Definability Bipartition Computation Using SAT Witnesses", in Proc. of the Logics in Artificial Intelligence – 18th European Conference, JELIA 2023, Dresden, Germany, September 20-22, 2023, Proceedings, Lecture Notes in Computer Science, Vol. 14281, pp. 697–711, Springer, 2023.
URL https://doi.org/10.1007/978-3-031-43619-2_47

Bipartitioning the set of variables $\mathrm{Var}(\Sigma)$ of a propositional formula Σ w.r.t. definability consists in pointing out a bipartition $\langle I, O\rangle$ of $\mathrm{Var}(\Sigma)$ such that Σ defines the variables of O (outputs) in terms of the variables in I (inputs), i.e., for every $o \in O$, there exists a formula Φ_o over I such that $o \Leftrightarrow \Phi_o$ is a logical consequence of Σ. The existence of Φ_o given o, I, and Σ is a coNP-complete problem, and as such, it can be addressed in practice using a SAT solver. From a computational perspective, definability bipartitioning has been shown as a valuable preprocessing technique for model counting, a key task for a number of AI problems involving probabilities. To maximize the benefits offered by such a preprocessing, one is interested in deriving subset-minimal bipartitions in terms of input variables, i.e., definability bipartitions $\langle I, O\rangle$ such that for every $i \in I$, $\langle I \setminus \{i\}, O \cup \{i\}\rangle$ is not a definability bipartition. We show how the computation of subset-minimal bipartitions can be boosted by leveraging not only the decisions furnished by SAT solvers (as done in previous approaches), but also the SAT witnesses (models and cores) justifying those decisions.

3.16 A Flock of Birds: Owl & Strix

Tobias Meggendorfer (Lancaster University Leipzig, DE)

Joint work of Tobias Meggendorfer, Javier Esparza, Jan Křetínský, Michael Luttenberger, Salomon Sickert, Philipp J. Meyer
Main reference Javier Esparza, Jan Kretínský, Salomon Sickert: "A Unified Translation of Linear Temporal Logic to ω-Automata", J. ACM, Vol. 67(6), pp. 33:1–33:61, 2020.
URL https://doi.org/10.1145/3417995

In this talk, I briefly outline the theoretical and practical advances that together form the foundation of Owl and Strix, state-of-the-art tools for LTL to automata translation and LTL synthesis, respectively.

This includes a large body of work, a small selection follows:

- Unified translation (JACM): `https://doi.org/10.1145/3417995`
- One theorem to rule them all (LICS): `https://doi.org/10.1145/3209108.3209161`
- Owl tool paper: `https://doi.org/10.1007/978-3-030-01090-4_34`
- Strix tool paper: `https://doi.org/10.1007/978-3-319-96145-3_31`

The small list above is the culmination of about a dozen papers, see the respective cites within for more details.

3.17 Pre-condition and Program Synthesis for Polynomial Specifications over Integers

Govind Rajanbabu (Indian Institute of Technology Bombay – Mumbai, IN), S. Akshay (Indian Institute of Technology Bombay – Mumbai, IN), Supratik Chakraborty (Indian Institute of Technology Bombay – Mumbai, IN)

Joint work of Govind Rajanbabu, S. Akshay, Amir Kafshdar Goharshady, Supratik Chakraborty, Harshit Jitendra Motwani, Sai Teja Varanasi

Main reference S. Akshay, Supratik Chakraborty, Amir Kafshdar Goharshady, R. Govind, Harshit J. Motwani, Sai Teja Varanasi: "Automated Synthesis of Decision Lists for Polynomial Specifications over Integers", in Proc. of the LPAR 2024: Proceedings of 25th Conference on Logic for Programming, Artificial Intelligence and Reasoning, Port Louis, Mauritius, May 26-31, 2024, EPiC Series in Computing, Vol. 100, pp. 484–502, EasyChair, 2024.

URL https://doi.org/10.29007/NJPH

In this talk, we will look at the problem of synthesizing both the program and pre-condition, when the post-condition is given as Boolean combination of polynomial inequalities and variables take integral values over a bounded region. The problem does not have a sub-exponential time procedure under Exponential Time Hypothesis. We will discuss an approach that is more efficient than naive enumeration by exploiting results from algebraic geometry.

3.18 Synthesis of Semantic Actions in Attribute Grammars

Subhajit Roy (Indian Institute of Technology Kanpur, IN)

Joint work of Subhajit Roy, Pankaj Kumar Kalita, Miriyala Jeevan Kumar

Main reference Pankaj Kumar Kalita, Miriyala Jeevan Kumar, Subhajit Roy: "Synthesis of Semantic Actions in Attribute Grammars", in Proc. of the 22nd Formal Methods in Computer-Aided Design, FMCAD 2022, Trento, Italy, October 17-21, 2022, pp. 304–314, IEEE, 2022.

URL https://doi.org/10.34727/2022/ISBN.978-3-85448-053-2_37

Attribute grammars allow the association of semantic actions to the production rules in context-free grammars, providing a simple yet effective formalism to define the semantics of a language. However, drafting the semantic actions can be tricky and a large drain on developer time. In this work, we propose a synthesis methodology to automatically infer the semantic actions from a set of examples associating strings to their meanings. We also propose a new coverage metric, derivation coverage. We use it to build a sampler to effectively and automatically draw strings to drive the synthesis engine. We build our ideas into our tool, PĀNINI, and empirically evaluate it on twelve benchmarks, including a forward differentiation engine, an interpreter over a subset of Java bytecode, and a mini-compiler for C language to two-address code. Our results show that PĀNINI scales well with the number of actions to be synthesized and the size of the context-free grammar, significantly outperforming simple baselines.

3.19 Reactive Program Synthesis Modulo LLM Code Generation

Mark Santolucito (Barnard College, Columbia University – New York, US)

Main reference Raven Rothkopf, Hannah Tongxin Zeng, Mark Santolucito: "Enforcing Temporal Constraints on Generative Agent Behavior with Reactive Synthesis", CoRR, Vol. abs/2402.16905, 2024.
URL https://doi.org/10.48550/ARXIV.2402.16905

Temporal logics are powerful tools that are widely used for the synthesis and verification of reactive systems. The recent progress on Large Language Models (LLMs) has the potential to make the process of writing such specifications more accessible. However, writing specifications in temporal logics remains challenging for all but the most expert users. A key question in using LLMs for temporal logic specification engineering is to understand what kind of guidance is most helpful to the LLM and the users to easily produce specifications. Looking specifically at the problem of reactive program synthesis, we explore the impact of providing an LLM with guidance on the separation of control and data–making explicit for the LLM what functionality is relevant for the specification, and treating the remaining functionality as an implementation detail for a series of pre-defined functions and predicates. We present a benchmark set and find that this separation of concerns improves specification generation. Our benchmark provides a test set against which to verify future work in LLM generation of temporal logic specifications.

3.20 A Short Introduction to Inductive Functional Programming

Ute Schmid (Universität Bamberg, DE)

Main reference Sumit Gulwani, José Hernández-Orallo, Emanuel Kitzelmann, Stephen H. Muggleton, Ute Schmid, Benjamin G. Zorn: "Inductive programming meets the real world", Commun. ACM, Vol. 58(11), pp. 90–99, 2015.
URL https://doi.org/10.1145/2736282

Inductive functional programming, also called inductive program synthesis, addresses the problem of learning (mostly recursive) functional programs from input/output examples. An related area of research is inductive logic programming (ILP). IP is a type of machine learning because programs (models) are synthesized by inductive generalisation. In contrast to statistical and neural approaches to machine learning, IP approaches typically only need a small number of training examples. Since learned models are represented in form of programs, IP belongs to the group of interpretable machine learning approaches. In the talk, I will give an introduction to inductive functional programming and also present basic concepts of ILP. Furthermore, I will point out how IP can be combined with Deep Learning Architectures for explainability.

3.21 Oracle-Guided Inductive Synthesis, Learning Theory, and LLMs

Sanjit A. Seshia (University of California – Berkeley, US)

More than a decade ago, I described how many problems in formal methods are effectively addressed through reduction to synthesis, including the synthesis of proof artifacts during verification, and synthesis within solvers such as for theory lemmas and quantifier instantiation. Additionally, I observed how an inductive, data-driven approach to synthesis is often very effective. In this talk, I review these ideas, which are also summarized in [1]. I also describe how they enable one to develop learning-theoretic foundations for synthesis, leading to the frameworks of formal inductive synthesis and oracle-guided inductive synthesis (OGIS), with initial theoretical results reported in [2]. Finally, I note how synthesis with large language models (LLMs) is but a special case of oracle-guided synthesis where the LLM forms an untrusted but often effective oracle for searching over large expression (program) spaces. I describe how we can use this oracle-guided synthesis view to formulate an approach to verified code transpilation with LLMs, which beats all conventional approaches to verified code transpilation – initial results are presented in [3].

References

1 Sanjit A. Seshia. Combining Induction, Deduction, and Structure for Verification and Synthesis. Proceedings of the IEEE, 103(11):2036–2051, 2015. Conference version in DAC 2012.

2 Susmit Jha and Sanjit A. Seshia. A Theory of Formal Synthesis via Inductive Learning. Acta Informatica, 54(7):693–726, 2017.

3 Sahil Bhatia, Jie Qiu, Sanjit A. Seshia and Alvin Cheung. Can LLMs Perform Verified Lifting of Code? Technical Report No. UCB/EECS-2024-11, EECS Department, UC Berkeley, March 2024.

3.22 An Approximate Skolem Function Counter

Arijit Shaw (Chennai Mathematical Institute, IN & University of Toronto, CA)

Joint work of Arijit Shaw, Brendan Juba, Kuldeep S. Meel
Main reference Arijit Shaw, Brendan Juba, Kuldeep S. Meel: "An Approximate Skolem Function Counter", CoRR, Vol. abs/2312.12026, 2023.
URL https://doi.org/10.48550/ARXIV.2312.12026

Motivated by the recent development of scalable approaches to Boolean function synthesis, we study the problem of counting Boolean functions: given a Boolean specification between a set of inputs and outputs, count the number of functions of inputs such that the specification is met. This stands in relation to our problem analogously to the relationship between Boolean satisfiability and the model counting problem. Yet, counting Skolem functions poses considerable new challenges. From the complexity-theoretic standpoint, counting Skolem functions is not only #P-hard; it is quite unlikely to have an FPRAS (Fully Polynomial Randomized Approximation Scheme) as the problem of even synthesizing one Skolem function remains challenging, even given access to an NP oracle. The primary contribution of this work is the first algorithm, SkolemFC, that computes an estimate of the number of Skolem functions.

SkolemFC relies on technical connections between counting functions and propositional model counting: our algorithm makes a linear number of calls to an approximate model counter and computes an estimate of the number of Skolem functions with theoretical guarantees. Moreover, we show that Skolem function count can be approximated through a polynomial number of calls to a SAT oracle. Our prototype displays impressive scalability, handling benchmarks comparably to state-of-the-art Skolem function synthesis engines, even though counting all such functions ostensibly poses a greater challenge than synthesizing a single function.

References

1 Arijit Shaw, Brendan Juba, and Kuldeep S. Meel. *An Approximate Skolem Function Counter*. Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, number 8, pages 8108–8116, 2024.

3.23 Counterexample-Guided DQBF Solving

Friedrich Slivovsky (University of Liverpool, GB)

Joint work of Franz-Xaver Reichl, Friedrich Slivovsky, Stefan Szeider
Main reference Franz-Xaver Reichl, Friedrich Slivovsky, Stefan Szeider: "Certified DQBF Solving by Definition Extraction", in Proc. of the Theory and Applications of Satisfiability Testing – SAT 2021 – 24th International Conference, Barcelona, Spain, July 5-9, 2021, Proceedings, Lecture Notes in Computer Science, Vol. 12831, pp. 499–517, Springer, 2021.
URL https://doi.org/10.1007/978-3-030-80223-3_34

In a Dependency Quantified Boolean Formula (DQBF), each existentially quantified variable is annotated with a dependency set consisting of universally quantified variables. A model of a DQBF consists of functions that correctly assign values to the existentially quantified variables based on the values of the universally quantified variables they depend on. Determining whether a DQBF has a model is NEXP-complete, and DQBFs can naturally express a range of synthesis problems. This talk presents an algorithm for finding a model of a DQBF that iteratively refines a candidate model based on counterexamples. It covers techniques that are crucial to make this approach work well in practice, such as identifying unique Skolem functions by propositional definition extraction, and finding local repairs of invalid functions.

3.24 A problem with functional synthesis

Mate Soos (University of Toronto, CA), Supratik Chakraborty (Indian Institute of Technology Bombay – Mumbai, IN), and Kuldeep S. Meel (University of Toronto, CA)

There is an issue we should address related to functional synthesis. Its definition is incomplete. It talks about inputs and outputs only – but many of the variables are in fact don't cares. It is easily imaginable that many users don't need the skolem function for a number of internal variables that are not inputs. However, current systems have to create a skolem function for these, too, potentially wasting the end user's resources. In my opinion, this should to be changed, allowing users to give a dontcare set.

So, if e.g. there are 100 variables, and the user sets 1..50 as inputs, they should be able to say that they are only interested in the skolem function of variable 100 – and if that involves variables 51..99 then of course their skolem functions, too. But if, for example, variable 100 is not connected in any way, shape or form, to variable 66, then there is absolutely no point in creating the skolem function for variable 66. It would be noting but a waste of the end user's resources. We should focus on what the end users want – and I'm quite sure they only want specific variable(s)' skolem functions, not everything that isn't an input.

Let's discuss. Obviously this new formulation can gracefully simulate the original problem, simply set dontcare = $\emptyset$. But I am pretty sure that once users start using functional synthesis, their dotcare set will be quite large.

3.25 Reactive synthesis via parity and Rabin games

K. S. Thejaswini (IST Austria – Klosterneuburg, AT)

To solve the reactive synthesis problem from LTL specifications or non-deterministic Buchi automata, there are two common approaches: either reduce it to the solving a parity game or solving a Rabin game. We discuss some algorithms to solve these games.

3.26 On the Power of LTLf in Reactive Synthesis

Shufang Zhu (University of Oxford, GB)

Joint work of Shufang Zhu, Geguang Pu, Moshe Y. Vardi, Jianwen Li, Lucas M. Tabajara, Giuseppe De Giacomo, Antonio Di Stasio, Marta Kwiatkowska, Pian Yu

Reactive synthesis emerges as a trustworthy-by-design technique in developing verifiably correct autonomous AI systems. This talk puts a particular focus on reactive synthesis of Linear Temporal Logic on finite traces (LTLf). LTLf, on the one hand, allows for specifying a rich set of temporally extended specifications, and on the other hand, focuses on finite traces, which makes it particularly suitable for specifying tasks of autonomous AI systems. Note that autonomous AI systems will not get stuck accomplishing a task for all their lifetime, but only for a finite (but unbounded) number of steps. In this talk, I will present an overview of key advancements in LTLf synthesis, highlighting its scalability and potential in complex scenarios. These results base on a so-called DFA-technology, which essentially takes the maximal simplicity of reasoning about efficiently constructed deterministic finite word automaton (DFA) of the LTLf objective. The goal of this talk is to engage researchers in automated synthesis, encouraging further advances on the scalability and applicability of LTLf synthesis.

References

1 Shufang Zhu, Lucas M. Tabajara, Jianwen Li, Geguang Pu, Moshe Y. Vardi: Symbolic LTLf Synthesis. IJCAI 2017: 1362-1369

2 Shufang Zhu, Giuseppe De Giacomo, Geguang Pu, Moshe Y. Vardi: LTLf Synthesis with Fairness and Stability Assumptions. AAAI 2020: 3088-3095

3 Giuseppe De Giacomo, Antonio Di Stasio, Moshe Y. Vardi, Shufang Zhu: Two-Stage Technique for LTLf Synthesis Under LTL Assumptions. KR 2020: 304-314

4 Shufang Zhu, Lucas M. Tabajara, Geguang Pu, Moshe Y. Vardi: On the Power of Automata Minimization in Temporal Synthesis. GandALF 2021: 117-134

5 Giuseppe De Giacomo, Antonio Di Stasio, Lucas M. Tabajara, Moshe Y. Vardi, Shufang Zhu: Finite-Trace and Generalized-Reactivity Specifications in Temporal Synthesis. IJCAI 2021: 1852-1858

6 Pian Yu, Shufang Zhu, Giuseppe De Giacomo, Marta Kwiatkowska, Moshe Y. Vardi: The Trembling-Hand Problem for LTLf Planning. To appear at IJCAI2024

4 Working groups

4.1 Benchmarking (and LLMs)

José Cambronero (Microsoft – Redmond, US), Johannes Klaus Fichte (Linköping University, SE), Benedikt Maderbacher (TU Graz, AT), Tobias Meggendorfer (Lancaster University Leipzig, DE), Ruzica Piskac (Yale University – New Haven, US), and Mark Santolucito (Barnard College, Columbia University – New York, US)

In our discussion, we focused on current challenges to identifying opportunities for impact for the work carried out by the reactive synthesis community. One point discussed was the need to identify (industrial) applications and then use these applications to drive benchmark development. While current benchmarks, such as SYNTCOMP, provide cases that are able to test the limits of current solvers (an important goal, and complementary to what we propose here), there is no evidence that the tasks being solved in these competitions are necessarily realistic for industrial use. Discussion surfaced some natural domains for possible application, such as networking, robotics, smart-home systems, or other applications that require substantial planning and long running environments. However, how to obtain concrete tasks from these domains remains a challenge. One idea discussed was the importance of establishing connections with such communities, potentially through venues like Dagstuhl, to bring together their problems with the solutions from the reactive synthesis community.

Finally, we discussed the potential for using generative AI (specifically LLMs) as a potential tool to improve or extend the current benchmarking done or existing tools. For example, we discussed that an LLM may be able to provide a natural language description of an LTL specification, which may be helpful for explainability (making this more accessible to non-experts). Similarly, an LLM may be potentially useful for generating LTL benchmark tasks, which may be of interest to evaluate performance on problems with different structure or that somehow reflect the bias of problems observed in their training data (which in turn may correlate with potential applications). Alternatively, we also discussed that LLM-based systems (e.g. agents) may themselves be a good application area for LTL.

One challenge identified by participants is that the community may not necessarily reward applications of techniques, since these efforts would be reflected in publications (or other achievements) in the target domain community instead.

4.2 Developing a Computational Theory for Learning Functions from Relations

Kuldeep S. Meel (University of Toronto, CA), Dror Fried (The Open University of Israel – Ra'anana, IL), Alfons Laarman (Leiden University, NL), Sanjit A. Seshia (University of California – Berkeley, US), and Mate Soos (University of Toronto, CA)

The working group participants included Dror Fried, Mate Soos, Sanjit A. Seshia, and Alfons Laarman. The group's primary objective was to explore the necessity for developing a computational theory for learning functions from relations. The discussions encompassed understanding the connections between the work of Jha and Seshia, particularly their paper titled *A Theory of Formal Synthesis via Inductive Learning* (2014).

Moreover, another significant line of discussion was the distinction between the traditional setting of computational learning theory, where the underlying specification guarantees a unique function, and the scenarios we are interested in. Specifically, the focus was on developing a theoretical framework that can characterize situations where it is possible to learn small Skolem functions, assuming such functions exist.

The discussions concluded with the consensus that these issues represent important open problems in the field, warranting further research and investigation.

5 Panel discussions

5.1 Reactive Synthesis and Model Counting Competitions

Guillermo A. Pérez (University of Antwerp, BE) and Johannes Klaus Fichte (Linköping University, SE)

Automated Reasoning (AR) covers different aspects of deductive reasoning as practiced in mathematics and formal logic. Practical and theoretical research enabled ground-breaking success in a variety of application domains. At the core of this success lie incredibly sophisticated and complex pieces of software (so-called solvers), which tackle specific problems of AR. Recurring (typically annual) solver competitions or evaluations play a significant role in this practical success. Competitions aim at advancing applications, identifying challenging benchmarks, fostering new solver development, enhancing existing solvers, bringing together various researchers, identifying challenges, and inspiring numerous new applications. In this talk, we share experience from two competitions. The Model Counting Competition (`https://mccompetition.org/`)[2] and the Reactive Synthesis Competition (`https://www.syntcomp.org/`)[1]. Subsequently, we provide thoughts and tasks for a broad discussion to establish a Boolean Function Synthesis Challenge.

References

1 Swen Jacobs, Guillermo A. Pérez, Remco Abraham, Véronique Bruyère, Michaël Cadilhac, Maximilien Colange, Charly Delfosse, Tom van Dijk, Alexandre Duret-Lutz, Peter Faymonville, Bernd Finkbeiner, Ayrat Khalimov, Felix Klein, Michael Luttenberger, Klara J. Meyer, Thibaud Michaud, Adrien Pommellet, Florian Renkin, Philipp Schlehuber-Caissier, Mouhammad Sakr, Salomon Sickert, Gaëtan Staquet, Clément Tamines, Leander Tentrup, Adam Walker: *The Reactive Synthesis Competition (SYNTCOMP): 2018-2021.* CoRR abs/2206.00251 (2022)

2 Johannes Klaus Fichte, Markus Hecher, Florim Hamiti: *The Model Counting Competition 2020.* ACM J. Exp. Algorithmics 26: 13:1-13:26 (2021)

Participants

- S. Akshay
Indian Institute of Technology Bombay – Mumbai, IN
- Ashwani Anand
MPI-SWS – Kaiserslautern, DE
- A. R. Balasubramanian
MPI-SWS – Kaiserslautern, DE
- Suguman Bansal
Georgia Institute of Technology – Atlanta, US
- Katrine Bjørner
New York University, US
- José Cambronero
Microsoft – Redmond, US
- Supratik Chakraborty
Indian Institute of Technology Bombay – Mumbai, IN
- Deepak D'Souza
Indian Institute of Science – Bangalore, IN
- Alexis de Colnet
TU Wien, AT
- Rayna Dimitrova
CISPA – Saarbrücken, DE
- Rüdiger Ehlers
TU Clausthal, DE
- Johannes Klaus Fichte
Linköping University, SE
- Bernd Finkbeiner
CISPA – Saarbrücken, DE
- Dror Fried
The Open University of Israel – Ra'anana, IL
- Mikoláš Janota
Czech Technical University – Prague, CZ
- Jie-Hong Roland Jiang
National Taiwan University – Taipei, TW
- Ayrat Khalimov
TU Clausthal, DE
- Alfons Laarman
Leiden University, NL
- Benedikt Maderbacher
TU Graz, AT
- Pierre Marquis
University of Artois/CNRS – Lens, FR
- Kuldeep S. Meel
University of Toronto, CA
- Tobias Meggendorfer
Lancaster University Leipzig, DE
- Jingyi Mei
Leiden University, NL
- Guillermo A. Pérez
University of Antwerp, BE
- Ruzica Piskac
Yale University – New Haven, US
- Govind Rajanbabu
Indian Institute of Technology Bombay – Mumbai, IN
- Subhajit Roy
Indian Institute of Technology Kanpur, IN
- Mark Santolucito
Barnard College, Columbia University – New York, US
- Ute Schmid
Universität Bamberg, DE
- Sanjit A. Seshia
University of California – Berkeley, US
- Shetal Shah
Indian Institute of Technology Bombay – Mumbai, IN
- Arijit Shaw
Chennai Mathematical Institute, IN & University of Toronto, CA
- Friedrich Slivovsky
University of Liverpool, GB
- Mate Soos
University of Toronto, CA
- K. S. Thejaswini
IST Austria – Klosterneuburg, AT
- Hazem Torfah
Chalmers University of Technology – Göteborg, SE
- Shufang Zhu
University of Oxford, GB

Report from Dagstuhl Seminar 24172

Code Search

Satish Chandra*[1], Michael Pradel*[2], and Kathryn T. Stolee*[3]

1 **Google – Mountain View, US.** `schandra@acm.org`
2 **Universität Stuttgart, DE.** `michael@binaervarianz.de`
3 **North Carolina State University – Raleigh, US.** `ktstolee@ncsu.edu`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar "Code Search" (24172). The seminar brought together researchers and practitioners working on techniques that enable software developers to find code and artifacts related to code. The participants discussed the state of the art in code search, identified open problems, and discussed future directions for research and practice. The seminar was structured with keynote talks, short talks, and breakout groups. Breakout groups identified how researchers can situate their code search research in terms of the targeted user groups, the access point for the developer, and the stage of software development that is most relevant to the code search tasks. Synergies between generative AI and Code Search were discussed, concluding that for some users and some tasks, generative AI can work with Code Search to enhance the developer experience and effectiveness. For other tasks, code search without generative AI would be more effective because of concerns regarding data provenance, update frequency, privacy, and the need for correctness.

Seminar April 21–24, 2024 – https://www.dagstuhl.de/24172

2012 ACM Subject Classification Software and its engineering → Integrated and visual development environments; Software and its engineering → Software development process management; Software and its engineering → Software evolution; Software and its engineering → Software libraries and repositories

Keywords and phrases code reuse, code search

Digital Object Identifier 10.4230/DagRep.14.4.108

1 Executive Summary

Kathryn T. Stolee (North Carolina State University – Raleigh, US)
Satish Chandra (Google – Mountain View, US)
Michael Pradel (Universität Stuttgart, DE)

The 3-day Dagstuhl Seminar on "Code Search" brought together leading experts from academia and industry to discuss and advance the field of code search. This seminar highlighted the critical role of code search in various software engineering activities, from locating where an error was thrown to learning new APIs or programming languages. It also emphasized the importance of search in automated software engineering tasks like automated program repair, code recommendation, and clone detection. The emergence of generative AI tools, which offer alternative methods for finding and reusing code, was also a significant topic of discussion.

* Editor / Organizer

Code Search, *Dagstuhl Reports*, Vol. 14, Issue 4, pp. 108–123
Editors: Satish Chandra, Michael Pradel, and Kathryn T. Stolee
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Participants explored the implications of code search research on developer productivity, code quality, and software engineering ethics. They examined the diverse tools available for code search, ranging from internal company tools to open-source platforms like GitHub, and generative AI tools like ChatGPT. The seminar addressed various dimensions of code search, such as appropriate scope for search results, indexing methodologies, and combinations of code search and LLMs, e.g., in the form of retrieval-augmented generation.

In addition to talks and informal discussions, there were several break-out sessions during which participants discussed specific topics in smaller groups and eventually reported back to the other participants. Sections 4.1 provides an overview of the breakout sessions.

As a result of the seminar, several participants plan to launch various follow-up activities, such as joint publications and transferring promising ideas from academia to industry.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

3 Overview of Talks

3.1 Trustworthy Code Search: A Data-Centric Perspective

Bowen Xu (North Carolina State University – Raleigh, US)

Data quality plays an important role in LLMs' performance. For code search, there also exist several data quality issues from different aspects that may affect LLMs. For example, security, consistency, label correctness, etc. Regarding this, I presented an open-source library named SEEDGuard (https://seedguard.ai) I am currently developing with my students. SEEDGuard aims to generate higher-quality data for building LLM4Code.

3.2 Representations for (searching) (for? in? with?) spreadsheets

José Cambronero (Microsoft – Redmond, US)

Spreadsheet environments remain one of the most popular platforms for end-users (and non-professional programmers) to carry out computational tasks. In contrast to traditional programming environments, spreadsheets are inherently multimodal: they contain tabular (and non-tabular) data; code in the form of sheet formulas, recorded macros, and small data analysis programs in popular languages like Python; artifacts of analyses such as plots and formatted tables; and natural language in the form of table headers, comments, and values. To expand the applicability of code search to such environments, we must inherently tackle retrieval (and similarity and so on) across these different types of data. In this talk, I argue one possible way to do so is to leverage learned representations. However, for these representations to be effective we must incorporate domain-specific insights into the learning process. To illustrate this, I present an approach to learning spreadsheet formula representations that incorporates data curation, spreadsheet-specific tokenization, and pretraining objectives. Next, I provide an overview of a table representation learning approach that incorporates hierarchical position information and sheet-oriented pre-training objectives that enable these representations to be effective for the heterogeneity of tables in spreadsheets. Finally, I present some results showing that the effectiveness of LLMs at solving basic table tasks (such as value lookups) when using prompting-based approaches are not robust to the table serialization.

3.3 DiffSearch: A Scalable and Precise Search Engine for Code Changes

Luca Di Grazia (Universität Stuttgart, DE), Michael Pradel (Universität Stuttgart, DE)

Joint work of Luca Di Grazia, Paul Bredl, Michael Pradel
Main reference Luca Di Grazia, Paul Bredl, Michael Pradel: "DiffSearch: A Scalable and Precise Search Engine for Code Changes", IEEE Trans. Software Eng., Vol. 49(4), pp. 2366–2380, 2023.
URL https://doi.org/10.1109/TSE.2022.3218859

The source code of successful projects is evolving all the time, resulting in hundreds of thousands of code changes stored in source code repositories. This wealth of data can be useful, e.g., to find changes similar to a planned code change or examples of recurring code improvements. This paper presents DiffSearch, a search engine that, given a query that describes a code change, returns a set of changes that match the query. The approach is enabled by three key contributions. First, we present a query language that extends the underlying programming language with wildcards and placeholders, providing an intuitive way of formulating queries that is easy to adapt to different programming languages. Second, to ensure scalability, the approach indexes code changes in a one-time preprocessing step, mapping them into a feature space, and then performs an efficient search in the feature space for each query. Third, to guarantee precision, i.e., that any returned code change indeed matches the given query, we present a tree-based matching algorithm that checks whether a query can be expanded to a concrete code change. We present implementations for Java, JavaScript, and Python, and show that the approach responds within seconds to queries across one million code changes, has a recall of 80.7% for Java, 89.6% for Python, and 90.4% for JavaScript, enables users to find relevant code changes more effectively than a regular expression-based search and GitHub's search feature, and is helpful for gathering a large-scale dataset of real-world bug fixes.

You can try our online instance here: `http://diffsearch.software-lab.org/diffsearch`.

3.4 AI-Resilient Interfaces and the Value of Variation

Elena Leah Glassman (Harvard University – Allston, US)

Joint work of Elena Leah Glassman, Ziwei Gu, Jonathan Kummerfeld
Main reference Elena L. Glassman, Ziwei Gu, Jonathan K. Kummerfeld: "AI-Resilient Interfaces", CoRR, Vol. abs/2405.08447, 2024.
URL https://doi.org/10.48550/ARXIV.2405.08447

AI is powerful, but it can make both objective errors and contextually inappropriate choices. We need AI-resilient interfaces that help people be resilient to the AI choices that are not right, or not right for them. Existing human-AI interaction guidelines recommend that interfaces include user-facing features for efficient dismissal, modification, or otherwise efficient recovery from AI choices that the user does not like. However, users cannot decide to dismiss or modify AI choices that they have not noticed, and, without sufficient context, users may not realize that some of the noticed AI choices are wrong or inappropriate. In this talk, I discuss the challenges and benefits of designing AI-resilient interfaces for code search, and how two complementary theories of human concept learning – Variation Theory and Analogical Learning Theory – can provide design guidance.

3.5 Coccinelle for Rust

Julia Lawall (INRIA – Paris, FR)

Joint work of Julia Lawall, Tathagata Roy

Coccinelle is a tool for code search and transformation that has been under development since the mid 2000s. The main novelty of Coccinelle was to design the transformation language around the notion of a patch, familiar to source-code developers, and to extend this with metavariables and information about types and control-flow. Coccinelle was originally designed to support large-scale transformation in the Linux kernel, and has been extensively used by Linux kernel developers. Today, we are investigating whether the same approach can be successful for Rust code. This talk reviews some of the main design decisions of Coccinelle for C, including writing the parser and pretty printer from scratch and the design of the control-flow graph. We then consider how those design decisions have been adapted to Rust, including more reuse of existing Rust tools, and the potential implications of those decisions.

3.6 An Academic Perspective on Code Search and AI

Tobias Kiecker (HU Berlin, DE)

Joint work of Tobias Kiecker, Lars Grunske

This talk examines the impact of artificial intelligence (AI), especially large language models (LLMs), on software engineering research in general and on code search in particular. It starts with a recap on how LLMs have advanced or replaced other code generation techniques in recent years. The talk then goes on to our previous research on code search, addressing how LLMs have influenced this area and might shape it further in the future. It concludes with an open challenge, namely the relatively low visibility of code search in academic research and teaching, and advocates for the integration of this topic into software engineering curricula.

3.7 My Code Search: Then, Now

Dongsun Kim (Kyungpook National University – Daegu, KR)

My “Code Search” journey has two phases. At the first phase, I have focused on traditional code search techniques, which take query strings and return locations of source code in local or global code repositories. I figured out that many users of code search tools experienced the vocabulary mismatch problem. To address this problem, I proposed the “two-step” query translation approach based on StackOverflow posts. I built two code search tools based on this idea: CoCaBu (for free-form queries) and FaCoY (for code-to-code search). These tools are effective in searching for code locations against a given (free-form and code) queries.

The second phase explores applications of code search. First, I proposed an approach to detecting and repairing wrong inconsistent method names. This approach searches for similar methods by method names and bodies after embedding them into vectors. Then, the approach compares neighboring sets of a method name and body to figure out the inconsistency between them. This approach successfully detects inconsistent names and suggests better names. My recent applications include improving LLMs with code search techniques: Memorization discovery and membership inference attack.

References

1 Zhou Yang, Zhipeng Zhao, Chenyu Wang, Jieke SHI, Dongsun Kim, DongGyun Han, David Lo, "Unveiling Memorization in Code Models", in the Proceedings of the 46th IEEE/ACM International Conference on Software Engineering (ICSE 2024), Lisbon, Portugal, April 14-20, 2024.

2 Zhou Yang, Zhipeng Zhao, Chenyu Wang, Jieke Shi, Dongsun Kim, DongGyun Han, David Lo: Gotcha! This Model Uses My Code! Evaluating Membership Leakage Risks in Code Models. CoRR abs/2310.01166 (2023)

3 Kui Liu, Dongsun Kim, Tegawendé F. Bissyandé, Taeyoung Kim, Kisub Kim, Anil Koyuncu, Suntae Kim and Yves Le Traon, "Learning to Spot and Refactor Inconsistent Method Names", in the Proceedings of the 41st International Conference on Software Engineering (ICSE 2019), Montréal, QC, Canada, May 25–31, 2019. Acceptance rate: 20.6% (109/529).

4 Kisub Kim, Dongsun Kim, Tegawendé F. Bissyandé, Eunjong Choi, Li Li, Jacques Klein, Yves Le Traon: FaCoY: a code-to-code search engine. ICSE 2018: 946-957

5 Raphael Sirres, Tegawendé F. Bissyandé, Dongsun Kim, David Lo, Jacques Klein, Kisub Kim, Yves Le Traon: Augmenting and structuring user queries to support efficient free-form code search. Empir. Softw. Eng. 23(5): 2622-2654 (2018)

3.8 A Journey through Searching Similar Code

Miryung Kim (University of California at Los Angeles, USA & Amazon Web Services – Palo Alto, USA)

This talk reflects on my group's research on searching similar code for the past 20 years, answering the following six questions: (1) What motivated us to research code search? (2) What were early attempts? (3) How serious is this problem? (4) How can we automate? (5) How can we examine variations at scale? (6) How to search similar code with a human in the loop?

We discuss that similar recurring updates motivated this line of work on searching similar code. As an early attempt, we created rule-based change abstractions and automatically inferred rules from diff-patches. We then quantified the effort needed to make similar changes in multiple contexts: co-evolution of forked projects, similar updates to clones, API evolution and ripple effects on client applications, and refactoring. We discussed our work on generalized patch synthesis to automate similar updates by learning from example patches. We realized the importance of examining search results at scale and designed a new visualization method of leveraging simultaneous overlay of similar code snippets. Then to enable construction of a search pattern with a human in the loop, we designed an active learning method that provides global distribution and what-if speculative analysis.

3.9 Code Search – Clone Search – Code Similarity

Jens Krinke (University College London, GB)

Joint work of Jens Krinke, Chaiyong Ragkhitwetsagul
Main reference Jens Krinke, Chaiyong Ragkhitwetsagul: "BigCloneBench Considered Harmful for Machine Learning", in Proc. of the 16th IEEE International Workshop on Software Clones, IWSC 2022, Limassol, Cyprus, October 2, 2022, pp. 1–7, IEEE, 2022.
URL https://doi.org/10.1109/IWSC55060.2022.00008

This talk presents the connection of code similarity to clone and code search. The application of clone search to investigate the provenance and quality of code on StackOverflow led to the development of a clone search engine evaluated with the often-used BigCloneBench dataset. However, this dataset is flawed due to how it has been constructed and the evaluation results are unreliable, particularly if the dataset is used to learn code similarity. Current work is on using LLMs to detect code similarity but another benchmark for functional similarity, CodeNet, is shown to be potentially illicit due to scraping copyrighted code. The talk concludes with preliminary results on using 36 LLMs to detect code similarity, which show that the LLMs are not yet ready for use as only six perform better than a random classifier.

3.10 Syntactic Code Search with Sequence-to-Tree Matching

Gabriel Matute (University of California – Berkeley, US)

Joint work of Gabriel Matute, Wode Ni, Titus Barik, Alvin Cheung, S. E. Chasins
Main reference Gabriel Matute, Wode Ni, Titus Barik, Alvin Cheung, Sarah E. Chasins: "Syntactic Code Search with Sequence-to-Tree Matching: Supporting Syntactic Search with Incomplete Code Fragments", Proc. ACM Program. Lang., Vol. 8(PLDI), pp. 2051–2072, 2024.
URL https://doi.org/10.1145/3656460

Lightweight syntactic analysis tools like Semgrep and Comby leverage the tree structure of code, making them more expressive than string and regex search. Unlike traditional language frameworks (e.g., ESLint) that analyze codebases via explicit syntax tree manipulations, these tools use query languages that closely resemble the source language. However, state-of-the-art matching techniques for these tools require queries to be complete and parsable snippets, which makes in-progress query specifications useless.

We propose a new search architecture that relies only on tokenizing (not parsing) a query. We introduce a novel language and matching algorithm to support tree-aware wildcards on this architecture by building on tree automata. We also present `stsearch`, a syntactic search tool leveraging our approach. In contrast to past work, our approach supports syntactic search *even for previously unparsable queries.* Our work offers evidence that lightweight syntactic code search can accept in-progress specifications, potentially improving support for interactive settings.

3.11 Scaling Embeddings for Github

Alexander Neubeck (GitHub – San Francisco, US)

There are a lot of papers and publications around RAG based systems. But most of the benchmarks, algorithms, and implementations focus on relatively small datasets (1-100 million embeddings) whereas at Github the scale is 100-1000x larger. At this scale, every tiny aspect of the RAG system must be revisited. Quality is now just one parameter in the equation, but no longer the most important one. One central part in RAG systems is the chunking strategy with the main focus to increase retrieval quality. However, at scale, stability and redundancy aspects become just as important. Picking a different strategy can decrease the cost easily by 10x and more. The talk shows for the various aspects of a RAG system which problems arise at scale and which questions need to be answered.

3.12 User Intent and Needs for Code Search

Nikitha Rao (Carnegie Mellon University – Pittsburgh, US)

Joint work of Vincent J. Hellendoorn, Martin Hirzel, Jason Tsay, Kiran Kate, Chetan Bansal, Thomas Zimmermann, Ahmed Hassan Awadallah, Nachiappan Nagappan, Joe Guan
Main reference Nikitha Rao, Jason Tsay, Kiran Kate, Vincent J. Hellendoorn, Martin Hirzel: "AI for Low-Code for AI", in Proc. of the 29th International Conference on Intelligent User Interfaces, IUI 2024, Greenville, SC, USA, March 18-21, 2024, pp. 837–852, ACM, 2024.
URL https://doi.org/10.1145/3640543.3645203
Main reference Nikitha Rao, Chetan Bansal, Thomas Zimmermann, Ahmed Hassan Awadallah, Nachiappan Nagappan: "Analyzing Web Search Behavior for Software Engineering Tasks", in Proc. of the 2020 IEEE International Conference on Big Data (IEEE BigData 2020), Atlanta, GA, USA, December 10-13, 2020, pp. 768–777, IEEE, 2020.
URL https://doi.org/10.1109/BIGDATA50022.2020.9378083
Main reference Nikitha Rao, Jason Tsay, Kiran Kate, Vincent J. Hellendoorn, Martin Hirzel: "AI for Low-Code for AI", in Proc. of the 29th International Conference on Intelligent User Interfaces, IUI 2024, Greenville, SC, USA, March 18-21, 2024, pp. 837–852, ACM, 2024.
URL https://doi.org/10.1145/3640543.3645203

Developers use search for various tasks such as finding code, documentation, debugging information, etc. First, we study user intents by conducting a large-scale analysis of web search behavior for software engineering tasks and propose a taxonomy of user intents. We then introduce a weak supervision based approach for detecting code search intent in search queries for C# and Java. Additionally, we present Search4Code, the first large-scale real-world dataset of code search queries mined from the Bing web search engine. Next, we extend our analysis beyond textual code and explore other forms of code representations such as low-code. We observe that different types of users (novices vs experts) may have different search needs, and demonstrate how LLMs can be useful in a visual (low-code) space, despite being trained on textual code, using LowCoder.

3.13 Code and Library Search

Christoph Treude (The University of Melbourne, AU)

Joint work of Brittany Reid, Marcelo d'Amorim, Markus Wagner, Christoph Treude
Main reference Brittany Reid, Marcelo d'Amorim, Markus Wagner, Christoph Treude: "NCQ: Code Reuse Support for Node.js Developers", IEEE Trans. Software Eng., Vol. 49(5), pp. 3205–3225, 2023.
URL https://doi.org/10.1109/TSE.2023.3248113

When developing software, finding the right pieces of code and the best libraries are important but challenging tasks. Code search lets developers find specific code snippets quickly, while library search helps them pick libraries that add more capabilities to their projects. To help, we've developed Node Code Query (NCQ), a tool that simplifies both tasks for Node.js developers. NCQ allows developers to search for NPM packages and code snippets, and it helps fix errors, set up testing environments quickly, and switch easily between searching and editing. Feedback from users shows that NCQ makes starting and finishing tasks faster, making it a valuable tool for Node.js developers. We've also started exploring methods to prioritize search results for diversity, ensuring users receive varied and useful results.

3.14 Querying code in Meta-SQL

Jan Van den Bussche (Hasselt University, BE)

Joint work of Jan Van den Bussche, Stijn Vansummeren, Gottfried Vossen
Main reference Jan Van den Bussche, Stijn Vansummeren, Gottfried Vossen: "Towards practical meta-querying", Inf. Syst., Vol. 30(4), pp. 317–332, 2005.
URL https://doi.org/10.1016/J.IS.2004.04.001

We recall some ideas presented more than 20 years ago on meta-querying. Collections of database queries, e.g., SQL statements from database catalogs, or query logs from SPARQL endpoints on the semantic Web, are also datasets of code that we may want to query. We are interested in expressive querying, so we represent queries as trees. We go one step further and also want to be able to query the behavior of queries. We describe Meta-SQL, a prototype language that uses SQL/XML for querying and transforming the tree structures of code, and which includes an added EVAL function to dynamically execute queries.

3.15 Code Search Perspectives from (Startup) Industry

Rijnard van Tonder (Mysten Labs – Palo Alto, US)

Main reference Rijnard van Tonder: "You Don't Know Search: Helping Users Find Code by Automatically Evaluating Alternative Queries", in Proc. of the 45th IEEE/ACM International Conference on Software Engineering: Software Engineering in Practice, SEIP@ICSE 2023, Melbourne, Australia, May 14-20, 2023, pp. 270–280, IEEE, 2023.
URL https://doi.org/10.1109/ICSE-SEIP58684.2023.00030

The value and future of Code Search lies in the concrete benefits provided to ordinary developers. Developers use code search for simple tasks (finding a function) and complex ones (regular expressions to refactor parts of code). The wide spectrum of use cases imply the need for levels of code search expressivity that cater to both novice and advanced users. We pose the question of how to provide greater value to developers (e.g., efficiency and time-savings for completing software tasks) in terms of search query expressivity. For example, we find that in practice, the majority of users do not regularly use regular expressions in search queries. Providing value (i.e., greater efficiency) to developers through code search implies discovering methods and evaluating usage (e.g., click behavior on results sets) to discover a balance of expressivity in code search. We share our outlook on what continues to work well in practice (fast literal code search via indexing), what's changing (LLMs and prompt queries), and challenges that remain difficult (e.g., discovering user intent, especially across heterogeneous users and organizations who use code search).

3.16 Searching for code that doesn't exist

Cristina Videira Lopes (University of California – Irvine, US)

Joint work of Cristina Videira Lopes, Rakib Misu, Iris Ma, James Noble
Main reference Md Rakib Hossain Misu, Cristina V. Lopes, Iris Ma, James Noble: "Towards AI-Assisted Synthesis of Verified Dafny Methods", Proc. ACM Softw. Eng., Vol. 1(FSE), pp. 812–835, 2024.
URL https://doi.org/10.1145/3643763

Large Language Models don't know much about Dafny, because not much code exists written in Dafny. I explain some experiments we did that drastically improve the LLMs' ability to generate formally verified algorithms written in Dafny.

3.17 Code Search at Google

Tobias Welp (Google – München, DE)

Code Search enables fast search for tokens, files, filtering for languages, etc. across large code bases and browsing through code with semantic information and cross references. It requires continuous investment in advancing the technology to keep up with code base growth. In comparison to Code Search, LLMs provide the opportunity to cover for wider knowledge gaps of the user, potentially addressing some of their Code Search needs better, but provide less precision with higher latency.

3.18 Codesearch in developer journeys

Ciera Jaspan (Google – Mountain View, US)

When we extract logs from developer tools, we can usee that code search is a common task across nearly every common developer journey. It's used when trying to answer questions while coding, to share information with others, to review code, to debug production issues, and to identify security problems, among many many others. Codesearch is used by developers mfultiple times a day for all of these tasks. However, we are frequently missing two pieces of information when understanding these developer journeys. First, while we can see that engineers are doing these tasks, we can't determine their intent. We can't tell what question they are trying to answer or what the context is that they want an answer for. Second, we can't tell when a task is "successful"; there is no way to distinguish between "I found my answer" and "I gave up". Until we can see these differences, it's very hard to determine whether new features of codesearch, especially ones powered by LLMs, are actively helping developers achieve their goals or whether they are getting in the way.

3.19 Code Search + Code Review = ♡

Bogdan Vasilescu (Carnegie Mellon University – Pittsburgh, US) and Reid Holmes (University of British Columbia – Vancouver, CA)

Tools that search through code, the history of software repositories, and other software artifacts (hereafter just "code search tools") have a long history of development and deployment in industrial practice. The data generated by code search tools is especially relevant given the large-scale, quickly-evolving nature of modern systems. However, one common design challenge facing most code search implementations is that it is easy to overwhelm users with too much information. But there is hope! Large language models and the conversational agents that usually go with them tend to be particularly useful at summarizing large volumes of information. Could they also help with amplifying code review with search? In this work we outline a vision for augmenting code review with extra information from code search tools coupled with advanced LLM-based techniques for analyzing and summarizing these data into information a patch-writer or reviewer could use to improve a proposed patch.

4 Working groups

4.1 Overview of Breakout Sessions

Kathryn T. Stolee (North Carolina State University – Raleigh, US), Boris Bokowski (Google – München, DE), José Cambronero (Microsoft – Redmond, US), Satish Chandra (Google – Mountain View, US), Jürgen Cito (TU Wien, AT), Luca Di Grazia (Universität Stuttgart, DE), Elena Leah Glassman (Harvard University – Allston, US), Georgios Gousios (TU Delft, NL), Reid Holmes (University of British Columbia – Vancouver, CA), Ciera Jaspan (Google – Mountain View, US), Tobias Kiecker (HU Berlin, DE), Dongsun Kim (Kyungpook National University – Daegu, KR), Miryung Kim (University of California at Los Angeles, USA & Amazon Web Services – Palo Alto, USA), Jens Krinke (University College London, GB), Julia Lawall (INRIA – Paris, FR), Gabriel Matute (University of California – Berkeley, US), Alexander Neubeck (GitHub – San Francisco, US), Michael Pradel (Universität Stuttgart, DE), Nikitha Rao (Carnegie Mellon University – Pittsburgh, US), Christoph Treude (The University of Melbourne, AU), Jan Van den Bussche (Hasselt University, BE), Rijnard van Tonder (Mysten Labs – Palo Alto, US), Bogdan Vasilescu (Carnegie Mellon University – Pittsburgh, US), Cristina Videira Lopes (University of California – Irvine, US), Tobias Welp (Google – München, DE), Bowen Xu (North Carolina State University – Raleigh, US), and Svetlana Zemlyanskaya (JetBrains GmbH – München, DE)

4.2 Code Search Needs of Different User Groups

This breakout focused on different groups of users of code search tools. This includes professional developers, students, legacy developers, and hobbyists. The tasks they are trying to accomplish include navigation, search, information acquisition, observability (e.g., Do I understand this problem to know how many people will be impacted? How much resources will solving this require? What are the impacts of this security vulnerability?), and debugging.

When addressing the needs of a particular user group, it is important to understand their entry point into code search. Given a specific micro-intent/goal, how will they access code search? Is it within a document as in Ctrl+F? Is it as a separate browser tab? From there, how can we help developers retain and regain their mental context? Last, how easy or hard is it to consume the search results?

4.3 Impact of Generative AI Tools on Code Search

We had three breakout groups with the following prompt for discussion: *"Come up with two concrete examples of how code search and LLMs are good and two where they are bad."* Each group reported out, and we summarize the main points.

4.3.1 The Good of LLMs + Code Search

The power of the LLMs can be used to assist users with understanding complex queries and patterns (e.g., regexes). Similarly, the LLM could be used to summarize or analyze the results from code search to aid with code comprehension. Another interesting use case could be for finding clones, which may be more likely to be generated by a LLM than by a real person. LLMs may uncover better search / query heuristics. Analyzing tradeoffs between different decisions (e.g., choosing packages). Good at presenting results in a personalized way (e.g, personalized summarization / aggregation etc). Help less expert users act more like power users.

The ability to support fuzzy searches with embeddings for re-ranking search results would be a useful combination of LLMs and Code Search. LLMs can also go to answering the actual question instead of just code retrieval. Test code generation was mentioned in particular. LLMs fall short when fact extraction / code navigation is necessary (or at least unnecessary).

4.3.2 The Bad of LLMs + Code Search

On the flip side, often, we need results of a query or prompt to be correct. Additionally, checking for the absence of something is hard (e.g., have all references to a depreciated API been updated?). Complete results are needed (audits, security, etc.)

There could be legal issues or privacy concerns that prevent sending data to LLM. Also, there were provenance concerns and concerns about update frequency.

There were also concerns for education with respect to whether students will learn how to read code without writing it.

4.4 Code Search at Different Stages of Software Development

This breakout group discussed the different stages of software development and how code search can be used at each stage. The group discussed different stages where code search is relevant, such as learning a new language, debugging, and code reuse. The group also discussed the different tools and techniques that can be used at each stage, such as code search engines, code search in IDEs, and code search in documentation.

Participants

- Boris Bokowski
Google – München, DE
- José Cambronero
Microsoft – Redmond, US
- Satish Chandra
Google – Mountain View, US
- Jürgen Cito
TU Wien, AT
- Luca Di Grazia
Universität Stuttgart, DE
- Elena Leah Glassman
Harvard University – Allston, US
- Georgios Gousios
TU Delft, NL
- Reid Holmes
University of British Columbia – Vancouver, CA
- Ciera Jaspan
Google – Mountain View, US
- Tobias Kiecker
HU Berlin, DE
- Dongsun Kim
Kyungpook National University – Daegu, KR
- Miryung Kim
University of California at Los Angeles, USA & Amazon Web Services – Palo Alto, US
- Jens Krinke
University College London, GB
- Julia Lawall
INRIA – Paris, FR
- Gabriel Matute
University of California – Berkeley, US
- Alexander Neubeck
GitHub – San Francisco, US
- Michael Pradel
Universität Stuttgart, DE
- Nikitha Rao
Carnegie Mellon University – Pittsburgh, US
- Kathryn T. Stolee
North Carolina State University – Raleigh, US
- Christoph Treude
The University of Melbourne, AU
- Jan Van den Bussche
Hasselt University, BE
- Rijnard van Tonder
Mysten Labs – Palo Alto, US
- Bogdan Vasilescu
Carnegie Mellon University – Pittsburgh, US
- Cristina Videira Lopes
University of California – Irvine, US
- Tobias Welp
Google – München, DE
- Bowen Xu
North Carolina State University – Raleigh, US
- Svetlana Zemlyanskaya
JetBrains GmbH – München, DE

Report from Dagstuhl Seminar 24181

Computational Metabolomics: Towards Molecules, Models, and their Meaning

Timothy M. D. Ebbels[*1], **Soha Hassoun**[*2], **Ewy A. Mathé**[*3], **Justin J. J. van der Hooft**[*4], **and Haley Chatelaine**[†5]

1 **Imperial College London, GB.** t.ebbels@imperial.ac.uk
2 **Tufts University – Medford, US.** soha@cs.tufts.edu
3 **National Institutes of Health – Bethesda, US.** ewy.mathe@nih.gov
4 **Wageningen University and Research, NL.** justin.vanderhooft@wur.nl
5 **NCATS – Bethesda, US.** haley.chatelaine@nih.gov

Abstract

Dagstuhl Seminar "Computational Metabolomics: Towards Molecules, Models, and their Meaning," (24181) is the fifth edition of the Computational Metabolomics seminars. Experts in fields ranging from cheminformatics, computer science, bioinformatics, analytical chemistry, and epidemiology attended to address the current state and future directions of this multi-disciplinary field. Specific topics of discussion were decided by participants but largely revolved around the seminar's titular themes of molecules (i.e., utilizing and annotating individual metabolites for use in models), models (i.e., generating systems from which to derive meaning), and meaning (i.e., deriving actionable insights to further understanding of biological systems). New to this seminar, topics of education and training, as well as the use of large language models to enhance access to resources, were also discussed. Participants identified community needs for a balance of standardization and flexibility in realms of repository-scale data deposition and analysis, spectral library generation, automation best practices, and biological pathway interpretation. Participants also identified a number of action items toward these ends, fostering international collaborations among them. For example, one topic evolved around creating a benchmarking dataset for structure annotation based on MS/MS spectral data. Discussions represented balanced perspectives, thanks to varied session facilitators and active participation of all members. The report contained herein reflects highlights of each session, including informal evening sessions and ideas for future directions.

Seminar April 28 – May 3, 2024 – https://www.dagstuhl.de/24181

2012 ACM Subject Classification Applied computing → Life and medical sciences

Keywords and phrases bioinformatics, cheminformatics, data integration, machine learning, mass spectrometry, metabolite identification, metabolomics, pathway analysis, repository-scale analysis, training and education

Digital Object Identifier 10.4230/DagRep.14.4.124

[*] Editor / Organizer
[†] Editorial Assistant / Collector

Computational Metabolomics: Towards Molecules, Models, and their Meaning, *Dagstuhl Reports*, Vol. 14, Issue 4, pp. 124–141
Editors: Timothy M. D. Ebbels, Soha Hassoun, Ewy A. Mathé, and Justin J. J. van der Hooft
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Timothy M. D. Ebbels (Imperial College London, GB)
Soha Hassoun (Tufts University – Medford, US)
Ewy A. Mathé (National Institutes of Health – Bethesda, US)
Justin J. J. van der Hooft (Wageningen University and Research, NL)

Metabolomics is the study of the small molecule composition of biological systems. These small molecules define biological functions, making metabolomics a broadly applied technology in the biomedical, environmental, and biotechnology fields of study. Metabolite measurements are typically produced using mass spectrometry (MS), usually coupled with liquid or gas chromatography (LC or GC), and/or nuclear magnetic resonance (NMR) spectroscopy. New technologies continue to be developed, leading to increased resolution of metabolite species detected, as well as increased sensitivity. The resulting datasets are high throughput and typically yield abundances of thousands of small chemical structures. For these reasons, the field of computational metabolomics continues to grow to address current and imminent issues in data stewardship, processing, analysis and interpretation.

This Dagstuhl Seminar is the 5th seminar in the computational metabolomics series. Previous seminars have addressed key topics in the field. These include leveraging spectral data to annotate and identify routinely measured metabolites, assessing interactions between metabolites and proteins through "metaboproteomic assays", as well as implementing multi-omic analyses and interpreting these data through enrichment analyses. This year, we not only dug deeper into the most current and relevant topics but also introduced new topics to the series. Our overall goal was to explore how to improve the utility of metabolomics data and its scientific relevance across many disciplines, alone or in combination with other data types, by leveraging machine learning and deep learning (ML/DL). To accomplish this, our seminar was organized into four categories. The first category was education, which was a newly introduced topic for this edition. Participants recognized the need for resources linking out to available education and training materials and discussed the inherent challenges of teaching multi-disciplinary topics, such as computational metabolomics. Our second category was molecules, which includes annotations and measurements of metabolites and molecules they associate with, such as proteins, genes, and other metabolites. New areas of emphasis included representation and classification of lipids, polymers and multi-constituent substances, and use of electron-activated disassociation methods for data collection. Our third category was models, which encompasses data quality, uncertainty in annotating metabolites, and relationships between metabolites and other molecules. Sessions in this category were the most prominent and included novel areas of repository-scale analyses, simulations of metabolomic data, and automation of data analysis workflows. Of note, practical sessions on how to best measure scientific impact and how/where to submit data publicly to increase utility of data and ability to develop new computational methods were held. Lastly, our fourth category was meaning, which represents resources, methods and tools that enable visualization and interpretation of large-scale metabolomic data in the context of biological, environmental or other sciences. This included the use and misuse of molecular networking and its current applications in the field. This year, new focus was placed on building interpretable models and leveraging increasingly available information on metabolite annotations, such as pathways and reactions. Of note, recent developments in large language model techniques were discussed throughout our categories, with a special session dedicated to prototyping an LLM with metabolomics-specific content that can be used for training the next generation of metabolomics experts.

As in previous years, and based on positive feedback, the seminar format was again flexible, and topics were finalized on the first day of the seminar. The audience was encouraged to bring forth topics, and attention was paid to rotate who moderated and took notes for the sessions. Moderators, as well as organizers, actively ensured that all participants had a voice in topic sessions. Due to the large number of topics and to keep the groups manageable in size, parallel sessions were held. At the end of each day, the last group meeting was held to summarize the day's discussion and to finalize the planning for the following day. One new aspect this year was the set-up of a Slack workspace that was used prior to, during, and after the seminar. This workspace facilitated communication and information exchange between participants. Overall, this seminar was highly successful, and participants were highly engaged. Topics and discussions generated much enthusiasm and concrete next steps for potential collaborations. The field of computational metabolomics is a very active field of study that is somewhat underrepresented in many of the main metabolomics conferences, especially when it comes to tool development. The opportunity to focus on the computational aspects was very well received and will surely continue to grow as the data generation and types are ever increasing.

2 Table of Contents

3 Overview of Talks

3.1 Biomarker discovery: current status of the field and experienced limitations

Carl Brunius (Chalmers University of Technology – Göteborg, SE) and Purva Kulkarni (Radboud University – Nijemgen, NL)

The session on "Combining Unsupervised and Supervised Models for biomarker discovery" was renamed to "Biomarker discovery: current status of the field and experienced limitations" after redefining the scope. The session started with defining the meaning of a biomarker, which is not a homogenous concept and brought forward different perspectives from the attendees.

The fact that the importance of biomarker validation should be significantly more than that of biomarker discovery was brought up since many biomarkers do not get approved easily. The scope for biomarker validation-related discussion included experimental and computational validation, application of analytical assays to determine the stability, reproducibility, effectiveness and possible costs for clinical translation. It also came up during the discussion that a lot is reported in repositories about biomarkers related to central metabolism but not much about metabolic biomarkers for exposure (exposomics studies). It was identified that there is a need for tools and databases to study drug mechanisms, possible side effects and consequences of changes in metabolism.

Regarding using metabolomics for biomarker discovery, several databases were discussed that contain reference values for listed metabolites. This was mainly in the context of metabolites commonly detected in bio-fluids over the life course. Apart from this, an initiative like the Consortium of Metabolomics Studies (COMETS) was brought up that contains information on available population-based cohorts. Additionally, it would be beneficial to know about publicly available datasets that have been used in the past for biomarker validation. It was discussed that, overall, biomarker research and translation into clinical practice needs to focus on increasing simplicity and stability and hence could focus more on targeted assays rather than untargeted metabolomics approaches.

In conclusion, the session emphasized the importance of biomarker validation and identifying approaches to have stable effective biomarkers, not just for disease diagnosis and therapy monitoring but also for risk prediction and prevention. Additionally, the discussion stressed the importance of exposure biomarkers from a more general epidemiological perspective, since exposure assessment is key to achieving relevant exposure-outcome risk assessments.

3.2 Generative molecular models

Roman Bushuiev (The Czech Academy of Sciences – Prague, CZ) and David Wishart (University of Alberta – Edmonton, CA)

In this session, we focused on the advancements and challenges in generating molecular structures from MS/MS spectra. We discussed various generative methods, including chemical language models based on SMILES strings and reinforcement learning approaches such as the Reinvent model. The discussion highlighted the need for new, more relevant metrics that reflect the ambiguity of a predicted molecular structure and new conditioning mechanisms that effectively leverage mass spectra. We explored related work in the field of drug design and discussed the relevance of mass spectrometry data beyond MS/MS. Insights from the session underscored the importance of de novo generation of molecules from mass spectra and identified key challenges.

3.3 Reporting knowledge and accounting for uncertainty in machine learning for metabolite annotations

Roman Bushuiev (The Czech Academy of Sciences – Prague, CZ) and Sebastian Böcker (Friedrich-Schiller-Universität Jena, DE)

This session delved into the challenge of reporting uncertainty in machine learning-based metabolite annotations from mass spectra. Discussions highlighted the importance of users understanding prediction uncertainty, particularly when dealing with spectra beyond the training data distribution. The focus was on the SIRIUS software, which estimates uncertainty by explaining spectral peaks based on a predicted molecular fingerprint. We explored user-friendly options for handling reported uncertainty, such as setting thresholds and enabling user-controlled visualization. Two main use cases were identified: identifying single compounds and conducting large-scale downstream statistical analysis, each requiring tailored uncertainty management. We brainstormed potential new methods for estimating uncertainty, including perturbation analysis of predicted structures and in silico fragmentation. Finally, we discussed uncertainty in the context of the undiscovered chemical space.

3.4 Degree of automation – misuse of tools, degrees of uncertainty, traceability, appropriate descriptors, etc.

Haley Chatelaine (NCATS – Bethesda, US) and Ewy A. Mathé (National Institutes of Health – Bethesda, US)

Automation holds strong promise in minimizing inevitable errors in collecting sample meta-data, protocol meta-data, and data processing parameters. This session involved a discussion of participants' existing use of automation, hurdles, and insights into necessities for best practices for its use on the aforementioned levels of study design. The traceability of automation, particularly when used in tandem with lab information management systems (LIMS), facilitates reproducibility. However, a clear core of common concepts that can be conserved across all metabolomics labs needs to be delineated so that labs know how they can optimize protocols for their unique fit-for-purposes. A repository of existing tools and best practices could also help facilitate this process. This is all with the caveat that automated processes need to incorporate appropriate diagnostics to identify sources of error in the process.

3.5 Visualization and networks for improved interpretability

Ronan Daly (University of Glasgow – Bearsden, GB) and Timothy M. D. Ebbels (Imperial College London, GB)

This session mainly focused on the direct visualization of networks related to metabolomics (and possibly multi-omics). These networks tend to have metabolites as nodes, with edges showing a relationship between them. Two of the main sources of network topology are knowledge driven, from bases such as KEGG or Reactome, and data driven using edge sources such as correlation or differential-correlation analyses. The "hairball" effect is known to be endemic in these data-based analyses, with methods such as partial correlation and filtering used to mitigate this. Tools to display metabolic networks were noted to be in short supply; programs such as Cytoscape and neo4j are used but do not cover all needs. New, easy-to-use tools that can flexibly display a range of different networks are needed. These should be able to use diverse knowledge bases as input, whilst being flexible as to other data that can be displayed, including annotating nodes and edges with complex data types and having layouts that are robust to initialisation. Including knowledge- and data-driven nodes and edges would be useful to display well-known and novel connections.

3.6 Combining unsupervised and supervised for metabolite annotation

Niek de Jonge (Wageningen University, NL) and Justin J. J. van der Hooft (Wageningen University and Research, NL)

This session discussed how we could combine unsupervised models with supervised models for annotating MS2 mass spectra and the potential of using multistage MSn mass fragmentation spectra as an input for such models. So far there have been trained unsupervised models (e.g., Spec2Vec, MS2LDA, and now very recently DreaMS) and supervised models such as MS2DeepScore. However, the approach of fine tuning unsupervised models has not been common for mass spectrum annotation but could be very promising. We discussed examples in other fields, like image labeling, and discussed how we can apply these principles to mass spectrum annotation. We have two concrete plans on how to do this. One approach would be to link mass spectral embeddings with molecular embeddings, the other approach is to fine tune unsupervised models to perform specific tasks like predicting chemical similarity. During the discussion, we highlighted potential routes, opportunities, and pitfalls when doing this. Overall, the group feels that this is a very promising avenue for further research where there will be place for both unsupervised and supervised models and combinations thereof.

3.7 Education in computational metabolomics

Timothy M. D. Ebbels (Imperial College London, GB), Ewy A. Mathé (National Institutes of Health – Bethesda, US), Stacey N. Reinke (Edith Cowan University – Joondalup, AU), and Denise Slenter (Maastricht University, NL)

The aim of this session was to identify ways in which the community can provide education and training content to a varied audience. Education deals with long-term and broad content that focuses on concept development while training deals with short-term and focused content that is skills-focused. Education pieces are thus relatively static and do not change much over time while training pieces are constantly evolving. Challenges in metabolomic education and training include dealing with many types of users with different levels of education on metabolomics analyses, difficulties in producing documentation, and variability in awareness of available content and workshops (e.g., spread across the internet and countries and not always findable). Notably, the desire to to bring the community together on providing consistent and reliable education and training content was strong. All participants saw the need to have a better overview of existing teaching materials available. Training people from different backgrounds (medical, chemistry, biology, computational, or a mixture) can be quite challenging. More embedding in existing curricula would be a good way to spark interest and enthusiasm for (computational) metabolomics. Having an online collection of existing materials, such as the Software Data Exchange through the Metabolomics Association of North America (MANA SODA), which is also checked for availability (e.g. R-package availability, function documentation) would help in keeping materials in line with the fast developments of software. Action items from this session include uniting the Metabolomics Society and Affiliates to consolidate available resources and piloting a chatbot using content that is provided from participants in this meeting.

3.8 Polymers and multi-constituent substances

Tytus Mak (NIST – Gaithersburg, US), Emma Schymanski (University of Luxembourg, LU), and Egon Willighagen (Maastricht University, NL)

A concerted effort is currently underway to create the computational infrastructure for supporting spectral data for multi-constituent substances (MCS), with the initial focus on polymers. NIST is in the early stages of acquiring data for building spectral libraries, with an initial focus on pyrolysis GC-MS but expanding to other instrument platforms such as LC-MS. NIST is also developing new algorithms to support the matching of polymers and polymer mixtures, which by their nature consists of a multitude of spectra at multiple retention times even for "pure" samples, rather than the one-compound-one-spectrum paradigm of traditional spectral libraries. Critical to this effort is the cheminformatics necessary to precisely define the molecular structures of polymers, and support via PubChem and other molecular databases. This effort is being led by collaborators at Maastricht University and University of Luxembourg.

3.9 Building interpretable models using pathways

Ewy A. Mathé (National Institutes of Health – Bethesda, US) and Timothy M. D. Ebbels (Imperial College London, GB)

Metabolomic data are difficult to interpret in terms of a biological "story." Biological pathways, here defined as lists of molecules participating in a common function, can be useful for this. Pathway enrichment is one example, but there are many other ways of employing this information. This discussion focused on highlighting gaps and needs for improving pathway enrichment and interpretation of metabolomic data and for encouraging novel approaches to the problem. Examples include adding pathophysiological pathways (e.g. necrosis) to databases, improving completeness of metabolite-pathway mappings, propagating uncertainty (e.g. from annotation) through to the pathway level, and quantification of specificity of the pathway signal. Suggested action items included a review paper highlighting these gaps, benchmarking study(ies) assessing uncertainty in the pathway methods, and methodological work addressing the question of how to quantify specificity.

3.10 Simulation of metabolomics data

Ewy A. Mathé (National Institutes of Health – Bethesda, US), Ronan Daly (University of Glasgow – Bearsden, GB), and Stacey N. Reinke (Edith Cowan University – Joondalup, AU)

The aim of this session was to explore and define the uses of simulated data and how generating gold-standard experimental metabolomics data with ground truth could help in numerous tasks. Globally, simulated metabolomic data provide a ground truth against which concepts can be demonstrated and are thus useful for benchmarking. Different contexts and types of analyses require different simulated datasets, with the axis from raw to processed data being particularly important. Coupled to this is the fact that mass spectrometry data is very heterogeneous, with possibly very flexible tools needed. Currently, most researchers simulate data in an ad-hoc manner, thereby making comparison of tasks and their utility difficult. Many uses of simulated datasets were defined, including evaluating algorithmic pipelines, statistical inference, and teaching/education. While some publications describe simulation tools, they are relatively few, and there is a lack of a centralized repository for benchmark and simulated data. A review of approaches to simulating metabolomic data as well as an associated repository of simulation tools were suggested as useful action items.

3.11 Using pathways/networks/reactions for metabolite identification

Hosein Mohimani (Carnegie Mellon University – Pittsburgh, US)

This session covered several angles on how the community can use pathways knowledge-bases to aid in metabolite annotation. Structure-based approaches predict the product of enzymatic reactions solely based on the substrate structure and knowledge about the reaction host/environment and enzymes involved. Metabolomics-based approaches exist that group mass features and filter for known mass differences that correspond to biotransformations. The consensus was that it might be too early to explore whole reactome databases, and looking into single reactions should be considered instead, as a low hanging fruit. Additionally, training datasets are currently really small, and additional experimental data collection efforts are needed to improve them. These experimental data should be collected on simple systems (e.g., individual enzymes), but more complex systems (e.g., whole microbiome) would not be useful, as it would be very difficult to annotate them.

3.12 Computational methods for biomarker discovery

María Eugenia Monge (CIBION – Buenos Aires, AR), Daniel Raftery (University of Washington – Seattle, US), and Denise Slenter (Maastricht University, NL)

This session focused on the accumulation of biomarker data, largely for use in clinical assays, and some of the challenges in this area, both in the sparsity of reliable biomarker information in existing databases and in the computational aspects in developing strong biomarker panels. There was a suggestion to focus more on the collection of large targeted assays that provide more reliable data and in particular, absolute concentrations. But some also recognized that untargeted metabolomics can deliver novel metabolite biomarkers. Reference ranges do exist for a large number of metabolites, and they are collected in MarkerDB. However, there are issues in these data, such as different sets of ranges that depend on experimental parameters, such that the ranges can be rather large. Regarding computational approaches, some suggestions were made for FDR correction and how metabolite classes (triglycerides, acylcarnitines) can be used instead of the individual metabolites. Similarly, ratios of metabolites are increasingly of interest, but methods to focus on the most important ratios are very important to avoid false discovery rate issues with the potentially huge number of potential ratios that could be calculated. It was agreed that more data need to be made available, possibly by soliciting researchers who have access to them, such as those who have already deposited data to the existing databases. Issues of confidentiality and patient consent may restrict these data to summaries, i.e., without individual-level metabolite values. Nevertheless, there seems to be some specific ways forward to improve the biomarker discovery and validation process, and better data quality and coverage will provide much better inputs for a variety of computational methods to develop strong biomarker panels.

3.13 Molecular networking in the GNPS environment: applications, considerations, and future directions

Raphael Reher (Universität Marburg, DE) and Justin J. J. van der Hooft (Wageningen University and Research, NL)

Molecular networking is a powerful method for visualizing and annotating the chemical space in non-targeted mass spectrometry (MS) data. We noted that "mass fragmentation-based spectral grouping" was a more accurate name, but Molecular Networking is well established in the community. In this session, various aspects related to molecular networking, including applications, flavors, downstream tasks, and visualization tools, were discussed. The applications are wide: i) finding analogues, ii) creating a mindmap of the data, iii) mass feature annotation (using the network topology), and iv) finding small groups of highly-interconnected nodes representing xenobiotics. We concluded that for some applications the network as such is not necessarily needed. We concluded that Molecular Networking combines several tasks in one (organization and visualization) and has a broad variety of applications that each tap into either one or both of the tasks. We noted that the default settings are not suitable for most tasks; however, we do not have good ways yet of assessing what thresholds, settings, and similarity scores would work best for which scenarios. We discussed the use of different similarity metrics, including various mass spectral scores, but also the "shared substructures" (inferred by MS2LDA), fingerprints (inferred by SIRIUS), or biochemical distance. Furthermore, we discussed that graph-based approaches have been applied to mass spectral networks in a very limited fashion, and this could be an interesting route to explore as an alternative to the current spectral grouping.

3.14 LLMs

Stacey N. Reinke (Edith Cowan University – Joondalup, AU) and Nicola Zamboni (ETH Zürich, CH)

Following discussions in the Education and Training sessions earlier in the week, Dagstuhl attendees in this session provided content, and an LLM prototype was designed. In this session, a demonstration of the prototype was conducted. We discussed challenges, considerations, and what is needed to improve the prototype in the future.

3.15 Multi-omics integration

Stacey N. Reinke (Edith Cowan University – Joondalup, AU) and Juan Antonio Vizcaino (EMBL-EBI – Hinxton, GB)

Multi-omics integration refers to analyzing more than one set of omics data to create a more comprehensive understanding of a biological system. Use cases of multi-omics integration vary from functional biochemistry in model systems to discovery-driven analysis in human studies. Several approaches, methods, and tools exist to integrate omics data; however, these are often complex and can be inaccessible to non-experts. Other challenges of performing multi-omics integration include a high data dimensionality (many variables), inherent bias to higher variable data blocks, and complexity of results. Future directions include making methods and tools more accessible so that a wider range of researchers can use them.

3.16 Mass spectral reference library generation and mass spectral quality

Robin Schmid (The Czech Academy of Sciences – Prague, CZ) and Tytus Mak (NIST – Gaithersburg, US)

This session recognized and discussed current issues concerning the generation of mass spectral reference libraries in the metabolomics “small” molecule space. Current libraries are static and often lack a reference to the original raw data. Furthermore, data (spectral) processing is done without providing reproducible processing parameters.

Briefly, the library generation workflow should be streamlined and made accessible to experimental mass spectrometrist, by abstracting many steps, including (a) compound and experimental metadata curation by structure cleanup and database lookup, (b) providing recommendations and SOPs for popular MS platforms and methods to guide data acquisition, and (c) create a Github repository to collect resources and tools for data processing and spectral quality scoring for library generation

Finally, we are planning to connect these tools into a workflow and plan the infrastructure and possible front ends. The remaining key issues are defining the required and optional metadata to be captured. We agreed that automatic processing and library generation has limitations, and manual curation needs to be guided and tracked, to preserve expert knowledge. There was a consensus that the 8 action items discussed during our session would significantly impact the mass spectrometry community if implemented. The 8 action items included: (1) contributing tools for depositing meta-data in the library, (2) defining a minimum requirement for compound and experimental meta-data, (3) dcoumenting best practices for MS data processing for library generation, (4) listing libraries and databases for data pushes, (5) dcoumenting who is open to sharing compounds, (6) creating a layout of software design and infrastructure, (7) setting up an online meeting for interested parties, and (8) creating a repository of spectral libraries and collecting feedback to verify interest in sending standard aliquots for library generation. We could easily increase the public MS libraries n-fold by inviting analytical chemists.

3.17 Using networks/reactions for metabolic networks and data interpretation

Denise Slenter (Maastricht University, NL), Timothy M. D. Ebbels (Imperial College London, GB), and Egon Willighagen (Maastricht University, NL)

Chemists appreciate the individual biochemical reactions covered in various databases. Combining all of these reactions in one network can be useful to overcome the relatively arbitrary boundaries inherent in categorizing biochemical pathways. However, the categorisation of reactions in pathways can be very relevant for the biological interpretation of (metabol)omics data.

Several issues were brought up that influence the usability of the currently captured machine-readable reaction knowledge. To name a few: the utility of capturing (missing/unknown) stereochemistry, mutable identification annotations, the ability to merge data based on name matching, the frequent inability to reuse existing models, low metabolite coverage, unknown metabolite biotransformations, and unconnected reactions.

Furthermore, the tools and techniques developed for other -omics fields are not directly applicable for metabolomics, due to sparsity and coverage issues. Finding reactions that are not in equilibrium from metabolomics data and changes thereof is still complicated today, and teaching materials and integration thereof in relevant curricula is lacking.

3.18 Towards MS/MS annotation benchmark at NeurIPS 2024

Michael Andrej Stravs (Eawag – Dübendorf, CH) and Roman Bushuiev (The Czech Academy of Sciences – Prague, CZ)

In this session, we discussed questions associated with the goal to submit a dataset and benchmark for computational metabolomics to NeurIPS 2024. The principal motivation is to attract interest from the machine learning community. For this purpose, the dataset and tasks must pose a minimal barrier of entry. It was decided to focus on a dataset with columns of molecular structure, mz, intensities and collision energy, with detailed datatype descriptions. Tasks should be formulated as prediction of specific columns as labels from a set of input columns. The tasks should include structure-to-spectrum prediction and spectrum to candidate list ranking. Disagreement reigned about whether a surrogate task such as fingerprint prediction should be included. Further, de novo structure generation was discussed as an attractive, but hard-to-evaluate, task. As another issue, the test-train split needs to be considered carefully to avoid "too similar" molecules in separate folds; MCES-based or scaffold-based splits were discussed. Importantly, naive baselines such as k-nearest neighbor are required to contextualize model performance.

3.19 Access to public data and how it should be submitted

Juan Antonio Vizcaino (EMBL-EBI – Hinxton, GB), Alice Limonciel (biocrates life sciences – Innsbruck, AT), and Ewy A. Mathé (National Institutes of Health – Bethesda, US)

In this session, we discussed the most frequent use cases of data reuse today: reproducing the result of a study, meta-analysis studies, reanalysing untargeted datasets to find new compounds, reanalysis with a different objective than the one from the original study, and machine-learning approaches using the mass spectra as the basis. Apart from GNPS-related efforts, there are limited examples about data reuse in metabolomics

If we had to choose among different types of metadata, it was mentioned that biological information was preferable since the methodological information is more difficult to capture in high-detail and most often it is required to read the manuscript.

The possibility of having a two-tier system for data submissions was discussed to increase the amount of data in the public domain. The people doing the higher-tier submission should get an incentive: a DOI, extra curation and extra promotion ("dataset of the week", or something similar).

It was also felt afterwards in the plenary session that we still should aspire to publish and share data in a way that enables cross-dataset (meta)studies and repository-scale analyses

3.20 Big data in metabolomics: repository-scale analyses

Juan Antonio Vizcaino (EMBL-EBI – Hinxton, GB) and Ralf Weber (University of Birmingham, GB)

The session on "Big data in metabolomics: repository-scale analyses" was well-attended, with people coming from multiple backgrounds. Discussions started with some current examples of data reuse, ranging from re-processing raw data, annotation and discovery of metabolites, machine-learning approaches to train models and the (lack of) meta-studies in metabolomics. These efforts surfaced some issues, both on a technical level and around (missing) metadata annotation and interruptions in the chain of evidence linking from raw data to annotated metabolites. In proteomics, the evidence chain goes backwards from the identified proteins and peptides, to the MS/MS spectra used in the identification and via the universal spectrum identifier (USI) directly to a spectrum in a deposited raw data file.

A possible carrot to improve the situation can (as always) be improved tooling, allowing to improve capturing metadata earlier in the process and export to the repositories, and, as an even bigger carrot, analysis software like MetaboAnalyst.

3.21 Scientific impact: Methods for article citations and annotations of why you're being cited

Egon Willighagen (Maastricht University, NL) and Carl Brunius (Chalmers University of Technology – Göteborg, SE)

In this session, Egon Willighagen introduced the topic with the history of the Citations Typing Ontology (CiTO), from the publication in 2010 [1], via the Journal of Cheminformatics pilot [2] and BioHackrXiv [3], to the indexing of the annotations in Wikidata and visualized by Scholia (link 1). The CiTO allows annotating citations with the intention of the citation. For example, you can indicate that you cite an article because you use data or database described in that article or that you use a method introduced in that article. It also allows you to indicate that you agree or disagree with that article, or that you ridicule that article. Using the annotations to create citation networks that reflect research reuse was discussed. For example, we can track the history of a software library (like the Chemistry Development Kit) or a database (like RaMP-DB) by following "cito:extends" citations. And when thinking about the impact of an article, citations by articles that reuse your work show more impact than citations that cite you as "related work." During the meeting, we looked at how nanopublications can allow authors to provide citation intention annotations after publication (where J. Cheminform. and BioHackrXiv allowed adding this annotation as part of the publication itself). Nanopublications are an approach using linked data [4] to publish a single fact (hence a "nano"-publication) but with full provenance. We looked at a recently developed template (link 2) that provides a graphical user interface. It requires the author of the nanopublication to authenticate with their ORCID account. This latter point is considered essential to the participants, who realize anyone can make nanopublications. The six participants wrote several nanopublications reflecting eight annotated citations (link 3). One of these gives the intent of one citation that it cites the article for information (cito:citesForInformation) (link 4).

Links:

1. `scholia.toolforge.org/cito`
2. `https://shorturl.at/x5MRB`
3. `https://shorturl.at/RPZHZ`
4. `https://shorturl.at/nutoC`

References

1 `https://doi.org/10.1186/2041-1480-1-S1-S6`
2 `https://doi.org/10.1186/s13321-023-00683-2`
3 `https://doi.org/10.5281/zenodo.10072013`
4 `https://doi.org/10.1109/escience.2018.00024`

3.22 Structural representation of lipid classes and enumeration

Egon Willighagen (Maastricht University, NL) and Michael Anton Witting (Helmholtz Zentrum München, DE)

Lipids cover a large combinatorial space based on different backbones, headgroups and fatty acyls. Current tools for the analysis of structural details in lipids only allow a limited depth in the annotation of structural features, such as location of functional groups, position and stereochemistry of double bonds, position of hydroxyl groups, etc. New fragmentation technologies such as EAD, OAD or UPVD allow to delve deeper into structural annotation. However, still some uncertainty remains. This session focused on ways to capture this uncertainty in chemical representations such as SMILES or InChIs. This is a problem not only specific to lipids, but also many other molecule classes. Extended SMILES and a new implementation for InChI isotopologue and isotopomer specifications have been discussed. Solutions will allow investigators, in the future, to potentially report structures together with uncertainty measures. Further action points include the collection of explicit real-life examples to further optimize the definitions that will be required for capturing uncertainty in structures.

3.23 Beyond Collision-Induced Dissasication with Electron-Activated Disassociation

Nicola Zamboni (ETH Zürich, CH)

The discussion of this session pivoted on the opportunities and needs associated with the adoption of electron-induced dissociation (aka EAD or EIEIO) for structural annotation of molecules. Much work has been devoted to lipids, and the results are far beyond expectations. It was discussed how to best address the analysis of different classes of molecules by the use of computational and ML methods. This includes (i) sharing EID data of representative classes with computational groups, (ii) measuring the information content of such spectra and comparing to CID, (iii) evaluating how CID-centric ML tools deal with the specific characteristics of EID spectra.

Participants

- Wout Bittremieux
University of Antwerp, BE
- Sebastian Böcker
Friedrich-Schiller-Universität Jena, DE
- Carl Brunius
Chalmers University of Technology – Göteborg, SE
- Roman Bushuiev
The Czech Academy of Sciences – Prague, CZ
- Haley Chatelaine
NCATS – Bethesda, US
- Ronan Daly
University of Glasgow – Bearsden, GB
- Niek de Jonge
Wageningen University, NL
- Kai Dührkop
Friedrich-Schiller-Universität Jena, DE
- Timothy M. D. Ebbels
Imperial College London, GB
- Soha Hassoun
Tufts University – Medford, US
- Florian Huber
Hochschule Düsseldorf, DE
- Pär Jonsson
Sartorius Stedim Data Analytics – Umeå, SE
- Purva Kulkarni
Radboud University – Nijemgen, NL
- Jessica Lasky-Su
Brigham and Women's Hospital & Harvard Medical School – Boston, US
- Alice Limonciel
biocrates life sciences – Innsbruck, AT
- Liping Liu
Tufts University – Medford, US
- Tytus Mak
NIST – Gaithersburg, US
- Ewy A. Mathé
National Institutes of Health – Bethesda, US
- Hosein Mohimani
Carnegie Mellon University – Pittsburgh, US
- María Eugenia Monge
CIBION – Buenos Aires, AR
- Steffen Neumann
IPB – Halle, DE
- Louis-Felix Nothias
CNRS & Université Côte d'Azur – Nice, FR
- Daniel Raftery
University of Washington – Seattle, US
- Raphael Reher
Universität Marburg, DE
- Stacey N. Reinke
Edith Cowan University – Joondalup, AU
- Hannes Röst
University of Toronto, CA
- Juho Rousu
Aalto University, FI
- Robin Schmid
The Czech Academy of Sciences – Prague, CZ
- Emma Schymanski
University of Luxembourg, LU
- Denise Slenter
Maastricht University, NL
- Jan Stanstrup
University of Copenhagen, DK
- Michael Andrej Stravs
Eawag – Dübendorf, CH
- Marynka Ulaszewska-Tarantino
Thermo Fisher Scientific – Milan, IT
- Justin J. J. van der Hooft
Wageningen University & Research, NL
- Dries Verdegem
VIB – KU Leuven, BE
- Juan Antonio Vizcaino
EMBL-EBI – Hinxton, GB
- Ralf Weber
University of Birmingham, GB
- Egon Willighagen
Maastricht University, NL
- David Wishart
University of Alberta – Edmonton, CA
- Michael Anton Witting
Helmholtz Zentrum München, DE
- Nicola Zamboni
ETH Zürich, CH

Report from Dagstuhl Seminar 24182

Resilience and Antifragility of Autonomous Systems

Simon Burton*[1], Radu Calinescu*[2], and Raffaela Mirandola*[3]

1 **University of York, GB.** simon.burton@york.ac.uk
2 **University of York, GB.** radu.calinescu@york.ac.uk
3 **Karlsruhe Institute of Technology, KIT, DE.** raffaela.mirandola@kit.edu

Abstract

In healthcare, transportation, manufacturing, and many other domains, autonomous systems have the potential to undertake or support complex missions that are dangerous, difficult, or tedious for humans. However, to achieve this potential, autonomous systems must be *resilient*: they must continue to provide the required functionality despite the anticipated and unforeseen disturbances encountered within their operating environments. This ability to achieve user goals in open-world environments can be further increased by making autonomous systems *antifragile*. Antifragile systems benefit from exposure to uncertainty and disturbances, by learning from encounters with such difficulties, so that they can handle their future occurrences faster, more efficiently, with lower user impact, etc. This Dagstuhl Seminar brought together leading researchers and practitioners with expertise in autonomous system resilience, antifragility, safety and ethics, self-adaptive systems, and formal methods, with the aim to: (1) develop and document a common understanding of resilient and antifragile autonomous systems (RAAS); (2) identify open challenges for RAAS; (3) discuss promising preliminary approaches; and (4) propose a research agenda for addressing these challenges.

Seminar April 28 – May 3, 2024 – https://www.dagstuhl.de/24182

2012 ACM Subject Classification General and reference → Reliability; General and reference → Metrics; General and reference → Validation; Computer systems organization → Embedded and cyber-physical systems; Computer systems organization → Dependable and fault-tolerant systems and networks; Software and its engineering; Theory of computation → Logic; Mathematics of computing → Probability and statistics; Computing methodologies → Artificial intelligence; Computing methodologies → Machine learning; Human-centered computing

Keywords and phrases artificial intelligence, antifragility, autonomous systems, disturbance, ethics, formal methods, machine learning, nondeterminism, resilience, safety, self-adaptive systems, validation and verification, uncertainty

Digital Object Identifier 10.4230/DagRep.14.4.142

1 Executive Summary

Simon Burton
Radu Calinescu
Raffaela Mirandola

The increasing complexity in the environment, tasks, and technology related to autonomous systems results in limitations in the statements that can be made regarding dependability during design time. In particular, these systems may operate within environments for which only incomplete models exist, that may change over time or may be subject to unforeseen

[*] Editor / Organizer

Resilience and Antifragility of Autonomous Systems, *Dagstuhl Reports*, Vol. 14, Issue 4, pp. 142–163
Editors: Simon Burton, Radu Calinescu, and Raffaela Mirandola
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

interactions and disturbances. As a result, such systems must be engineered to be trustworthy despite residual insufficiencies in their design, and in the presence of unexpected events due to their dynamically evolving operating context.

Related domains concerned with system autonomy in uncertain environments have already taken inspiration from nature to endow artificial systems with self-* properties (e.g. self-optimisation, -repair, -protection, -configuration, and -adaptation). Such self-* capabilities enable systems to improve their performance and dependability at runtime while reducing the need for low-level human intervention – properties that are closely related to resilience and antifragility.

This Dagstuhl Seminar aimed to unify the international research on **resilient and antifragile autonomous systems** (RAAS), leading to faster scientific advancements and industrial adoption. To this end, the seminar brought together leading researchers and practitioners with expertise in autonomous system resilience, antifragility, safety, and ethics, from disciplines including computer science, safety science, and ethics, to share and discuss each other's understanding of, methods for, and open challenges related to RAAS. Initial presentations were used to set the scene by proposing basic definitions, industry perspectives, and engineering views on cyber-resilience. These were followed by group and plenary discussions to explore these concepts in more detail.

A clear set of agreed definitions is essential in order to make progress as a community in this area. *Resilience* can be broadly seen as the ability to absorb disturbances and unexpected events whilst maintaining essential properties of the system. Using such conditions to harden the system against future events can be viewed as *antifragility*. These definitions highlight that antifragility is a concept referring to systems designed to operate under "open-world" assumptions, where the responsibility of maintaining a given property, despite disturbances (resilience) mostly shifts from design time to runtime, and relies on the presence in the system of some suitable degree of autonomy (self-* capability). As such, antifragility can be viewed as the ability of a system to *self-improve* its resilience over (run)time. Discussions converged to the idea that in order to define resilience and antifragility, we should build on the work of Control Theory, specifically how systems recover from (potentially previously unknown) disturbances. Thus, we postulated that both resilience and antifragility should be defined over the metrics of settling time, percentage of settling, percentage of overshoot, and percentage of overshoot with respect to the properties of interest in the event of disturbances to the system. Discussions on how to use formal methods to construct systems that guarantee these desired properties generated many challenging questions that are to be followed up in future research.

Initial work in the seminar explored more precise definitions of RAAS that also included the consideration of uncertainty and causality, and where a collection of properties may need to be optimised as a whole. Such trade-offs are particularly evident when considering safety, ethical, and legal aspects of RAAS. In some cases, autonomous systems must remain operational in order to stay safe. A resilient system could remain within its safety bounds when disrupted, whilst maintaining a minimal level of utility. An antifragile system could use repeated disturbances to lower risk over time whilst increasing overall utility. Similar trade-offs and optimisations will be found when considering legal and ethical concerns for RAAS and these could lead to specific technical requirements on the system. For example, for a system that adapts its function over time, avoiding the loss of agency in human stakeholders needs to be ensured.

Engineering antifragile systems requires specialised consideration in each phase of the traditional software and system development process. This includes requirements, design, implementation, and testing. Artificial Intelligence (AI) – in terms of machine learning, symbolic AI techniques, and combinations thereof – has the potential to provide a basis for both recognising disturbances and deciding the system adaptations needed to mitigate these disturbances. The seminar participants see potential for AI to be used in all phases of the MAPE-K (monitor-analyse-plan-execute supported by knowledge) cycle of self-adaptive systems. Furthermore, a control-theoretic reasoning approach could be used to verify whether a particular adaptation manager pushes the resilience error (i.e., the difference between observed and preferred resilience) below some threshold, or whether the resilience level stabilises at a reference value.

The seminar concluded that much work is still required to advance research in the area of RAAS, and to foster RAAS adoption in industrial applications. This includes:

- Agreeing on terminology and definitions that build upon and extend our traditional understanding of dependable systems;
- Formally defining metrics for resilience and antifragility that can be used to design and verify RAAS;
- Engineering methods and candidate technologies for implementing RAAS;
- Considering the safety, legal, and ethical implications of RAAS, including both their positive potential and their associated risks.

The participants agreed to pursue these important and challenging issues in future collaborations, including joint publications, workshops, and journal special issues.

2 Table of Contents

Executive Summary

Overview of Talks

Working Groups

Birds-of-a-Feather Groups

3 Overview of Talks

3.1 Characterizing Antifragile ICT Systems: Conceptual and Architectural Models

Vincenzo Grassi (University of Rome "Tor Vergata" – Rome, IT, vincenzo.grassi@uniroma2.it) and Diego Perez-Palacin (Linnaeus University – Växjö, SE, diego.perez@lnu.se)

Joint work of Vincenzo Grassi, Raffaela Mirandola, Diego Perez-Palacin
Main reference Vincenzo Grassi, Raffaela Mirandola, Diego Perez-Palacin: "A conceptual and architectural characterization of antifragile systems", J. Syst. Softw., Vol. 213, p. 112051, 2024.
URL https://doi.org/10.1016/J.JSS.2024.112051

Antifragility is one of the terms that have recently emerged with the aim of indicating a direction that should be pursued toward the objective of designing ICT systems that remain trustworthy despite their dynamic and evolving operating context. We present a characterization of antifragility, aiming to clarify from a conceptual viewpoint the implications of its adoption as a design guideline and its relationships with other approaches sharing a similar objective. To this end, we discuss the inclusion of antifragility (and related concepts) within the well-known dependability taxonomy presented in [1], which was proposed a few decades ago with the goal of providing a reference framework to reason about the different facets of the general concern of designing trustworthy systems able to cope with changes.

Indeed, we believe that a primary need for a software engineer involved in the design of such systems is to have a commonly agreed-on repertoire of terms and underlying concepts, which makes clear which system aspects each term intends to capture, whether some term is a specialization (qualifications) of some other, or if it denotes a means for attaining a property indicated by another term. In this perspective, our position is that the crisp conceptual reference provided by the dependability taxonomy should not be lost or obfuscated but rather updated and expanded, if necessary. The extension we discuss allows us to integrate the *antifragility* term and the underlying concepts into that taxonomy, thus maintaining its role of a unified place where the relationships among different goals and approaches aimed at designing and building ICT systems able to cope with changes can be better understood and compared.

Then, based on this conceptual clarification, we also discuss how to promote the engineering of antifragile systems. To this end, we first present a reference model for antifragile ICT systems inspired by the three-layer reference model for self-managing systems proposed in [2], and then we delineate a path based on the Digital Twin technology for the realization of antifragile systems.

A thorough presentation of the issues discussed in this talk can be found in [3].

References

1 A. Avizienis, J.-C. Laprie, B. Randell and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing, in IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, pp. 11-33, Jan.-March 2004.

2 J. Kramer and J. Magee, "Self-Managed Systems: an Architectural Challenge," Future of Software Engineering (FOSE '07), pp. 259-268, 2007.

3 V. Grassi, R. Mirandola and D. Perez-Palacin. A conceptual and architectural characterization of antifragile systems, in Journal of Systems and Software, Vol. 213, Article Nº 112051, 2024.

3.2 Emulation, Standards, and Ethics for Resilient and Antifragile Autonomous Systems

Lee Barford (Keysight Technologies – London, GB)

This talk covers the development and verification of robust and resilient autonomous systems from an industry perspective, with a focus on the role of emulation, standards, and ethics in the process. Many autonomous systems assess or control the physical world in real-time through sensors, actuators, or antennas. To validate them, emulations of the environments in which such systems operate must be provided, the sensors being fed accurate physical excitations and the simulation behind the emulation being updated by the actuator behaviours. In this manner, scenarios too dangerous or expensive to do in real life can be used to validate or provide high-quality synthetic training data. An example such emulation/validation system for autonomous drive is presented. In the case of robust and resilient autonomous systems, the need for such hardware-in-the-loop emulation is even greater, as then validating robustness requires that the system be run through scenarios too rare to appear in normal volumes of training data.

To realize resilience and antifragility in industrial-produced systems, standards for resilience and antifragility need to be developed and adopted. Such standards should be able to be turned into scenarios for training, fine-tuning, functional tests, and conformance tests for a particular robust and resilient system. Tools for design and model analysis need to be developed to ease achieving standards compliance. Where an autonomous system can monitor itself and upload status information for continuous improvement of the system, key performance indicators of resilience and antifragility that relate back to the standards and system requirements need to be created that are informative but (1) have a low burden on deployed system, and (2) require a low comms bandwidth in normal situations.

Creators of robust and resilient systems would also benefit from values-based design, where systems are designed from the beginning with the anticipation of impacts on human values in mind. This approach benefits investors, managers, engineers, customers, and the public by introducing ethical clarity at the requirements-gathering phase, when changes are easier and cheaper to make than later in the design process. A process of identification of stakeholders and their values is necessary to identify and prioritize socio-ethical risks that then become requirements for resilience and antifragility.

3.3 Towards Operational Cyber Resilience

Kerstin I. Eder (University of Bristol, GB & Trustworthy Systems Laboratory – Bristol, GB, Kerstin.Eder@bristol.ac.uk)

Joint work of Kerstin I. Eder, Carsten Maple, Peter Davies, Chris Hankin, Greg Chance, Gregory Ephiphaniou[1]
Main reference Kerstin Eder: "CyRes: towards operational cyber resilience", in Proc. of the 1st International Workshop on Verification of Autonomous & Robotic Systems, VARS '21, Association for Computing Machinery, 2021.
URL https://doi.org/10.1145/3459086.3460119

Existing approaches to cyber security in the automotive sector are not fit to deliver the resilience required for safe mass deployment of advanced driving features and smart mobility services. In this presentation I introduced an innovative multi-directional approach to operational cyber resilience, the CyRES methodology, which aims to enable the delivery of robust and resilient engineering practices in this sector from design, via manufacture to operation. CyRES is based on three principles: increasing the probability of Detection, Understanding and Acting on cyber events; increasing the number of Engineered Significant Differences; and invoking a continuum of Proactive Updates. I motivated, illustrated and explained these principles on examples, focusing mainly on the first two principles. CyRES is an exciting opportunity for engineers and computer scientists to re-target widely studied, mature methods, such as those developed by the self-adaptive systems community, for cyber security. My main objective was to raise awareness of the many intellectual challenges associated with realising these principles, and to highlight some of the ways for attendees to contribute to the realisation of the CyRES vision.

Further details on CyRES and the underlying principles can be found in [1].

References

1 K. Eder. CyRes: Towards operational cyber resilience. In Proceedings of the 1st International Workshop on Verification of Autonomous & Robotic Systems (VARS'21). Association for Computing Machinery, New York, NY, USA, Article 11, 1–3.
`https://doi.org/10.1145/3459086.3460119`.

[1] as part of the Cyber Resilience in Connected and Autonomous Mobility project ResiCAV, which was supported by funding from The Centre for Connected and Autonomous Vehicles (CCAV) run by Zenzic and Innovate UK, project number 133899.

4 Working Groups

4.1 Concepts, terminology & definitions for resilience and antifragility 1

Vincenzo Grassi (University of Rome "Tor Vergata", IT)
Ada Diaconescu (Telecom Paris, FR)
Felicita Di Giandomenico (CNR – Pisa, IT)
Gabriel Moreno (Carnegie Mellon University – Pittsburgh, US)
Elena Navarro (University of Castilla-La Mancha, ES)
Sebastián Uchitel (University of Buenos Aires, AR)

The group focused its discussion on the following issues:

- arriving at a suitable definition of antifragility;
- clarifying the relationships of this concept with other concepts, e.g., resilience, dependability, self-adaptation, and learning;
- identifying possible "parallel" specializations of the antifragility concept for different domains.

About the first issue, the discussion led us to conclude that it is useful to start with a declarative definition of antifragility that states what we expect from a system to consider it antifragile, remaining neutral with respect to how-to make it antifragile, and with respect to measures of its antifragility degree. To this end, the following definition emerged from the discussions:

Antifragility is the ability of a system to self-improve its resilience over (run)time.

This definition highlights that antifragility is a concept referring to systems designed to operate under "open-world" assumptions, where the responsibility of achieving a given property (resilience) mostly shifts from design time to runtime, and relies on the presence in the system of some suitable degree of autonomy (self-* capability).

This same definition also provides a perspective on dealing with the other two issues considered in the discussion.

For the second issue, it establishes, in particular, a relationship between antifragility and resilience by assigning to antifragility the role of an attribute of the process followed to attain and/or improve resilience. Hence, antifragility denotes a system's ability to incrementally achieve at runtime higher levels of resilience. Importantly, antifragility does *not* imply that the system *is* resilient, only that it is able to *improve* its resilience over time.

Concerning the "learning" concept, the given definition purposely avoids its use to leave space for approaches to antifragility that are not necessarily based on the explicit use of learning methodologies (even if we recognize that they can play a significant role).

For the third issue, it suggests that the specialization of antifragility for different domains can be at least partially deferred to the different characterizations (and measures) of resilience for those domains. This specialization looks at the property to be achieved. Besides this, another possible specialization could concern the process to be followed to that end, which could depend on the considered domain.

The discussion in the group also touched on issues concerning the how-to aspect. Emerged suggestions about approaches that could be pursued to achieve antifragility include:

- MAPE-K;
- Observer Controller;
- Learn (acquire knowledge), Reason, Act;
- Data collection, generalization, action;
- Exploration and exploitation;
- Evolutionary algorithms.

4.2 Concepts, terminology & definitions for resilience and antifragility 2

Ralf H. Reussner (KIT – Karlsruher Institut für Technologie, DE)
Amel Bennaceur (The Open University – Milton Keynes, GB)
Mario Gleirscher (Universität Bremen, DE)
Antje Loyal (Continental Automotive Technologies – Frankfurt, DE)
Raffaela Mirandola (KIT – Karlsruher Institut für Technologie, DE
Diego Perez-Palacin (Linnaeus University – Växjö, SE)
Patrizia Scandurra (University of Bergamo – Dalmine, IT)

The group discussed formalisations of the meaning of antifragility based on the paper "A conceptual and architectural characterization of antifragile systems" by Vincenzo Grassi, Raffaela Mirandola and Diego Perez-Palacin. In particular, changes of the environment were formalised as an extension of this paper. Main insights (including the following plenary discussion) were the clarification of the difference between resilience and antifragility. While both concepts may deal with unknown unknowns to a certain degree, antifragility is characterised through learning from prior events to improve. This can be seen as a higher-order adaptation mechanism, using prior events to change the adaptation mechanism to achieve a higher level of quality. This implies that the boundaries of subsets "dead" (i.e., catastrophic failure) and "survivable" of the system state space are changed through this higher order adaptation. We identified examples ranging from technical systems (autonomous vehicles, learning in e-scooters) to society (emergency forces learning from previous operations).

4.3 Safety concerns for resilient and antifragile autonomous systems

Kerstin I. Eder (University of Bristol, GB)
Simon Burton (University of York, GB)
Marc Carwehl (Humboldt-Universität zu Berlin, DE)
Andreas Heyl (Robert Bosch GmbH – Stuttgart, DE)
Ravi Mangal (Carnegie Mellon University – Pittsburgh, US)
Shiva Nejati (University of Ottawa, CA)
Gricel Vázquez (University of York, GB)

This session explored safety concerns for resilient and antifragile autonomous systems. We focused on the question "How do resilience and antifragility help with safety?" Our observations covered risk over time, with a specified maximum level of risk (safety boundary) beyond which the system was considered unsafe, as well as utility over time, with a minimum required level of utility (liveness) beyond which the system was considered no longer useful.

The baseline for our discussion was a resilient system that, when disrupted, remains within its safety boundary with respect to the operational risk while utility degrades to zero, rendering the system useless. Depending on the application, zero utility may imply that risk falls to zero along with utility, or that risk is maintained on the original level. The former is exemplified in scenarios where not doing anything is safe, while the latter represents scenarios where not doing anything is not safe.

Enhanced resilience means that the system, when disrupted, remains within its safety boundary and maintains utility at the desired level. Repeated exposures to shocks are absorbed by the system over time, with periods of higher risk being coupled to lower utility, and periods of lower risk being associated with higher utility. Such systems are not designed to improve performance over time, though they recover each time they are exposed to a disturbance.

When an antifragile system is disrupted, it gradually, though not necessarily monotonically lowers the risk and improves utility. Antifragile systems can operate in a variety of different safe subsets of operational states, which can be modified and extended by a controller associated with the system.

A formalisation must capture both safety and utility (liveness) properties of the autonomous system in such a way that these properties constitute a measurable representation of the application-specific requirements for the given system.

Open questions include "How to make resilient and antifragile systems safe?" and "How to maintain safety after changes in systems with resilience and antifragility?"

4.4 Legal and ethical concerns for resilient and antifragile autonomous systems

Ana Cavalcanti (University of York, GB)
Lee Barford (Keysight Technologies – London, GB)
Radu Calinescu (University of York, GB)
Matteo Camilli (Politecnico di Milano, IT)
Sebastian Hahner (KIT – Karlsruher Institut für Technologie, DE)
Lina Marsso (University of Toronto, CA)
Catia Trubiani (Gran Sasso Science Institute – L'Aquila, IT)

We have first considered the aspects of the life-cycle of design and verification of a system that are relevant when considering legal and ethical issues. We have identified the extensive list below, but started our discussions with the question "What is the relationship between legal and ethical concerns and resilience/antifragility?" We have agreed that legal and ethical concerns are present in all systems, but in autonomous systems, requiring adaptation at runtime, these issues cannot be resolved a priori. The loss of agency raises threats. On the other hand, we noted that runtime adaptation, resilience and antifragility also can create opportunities, since an intelligent system can provide additional services.

Our list of concerns goes from requirements all the way to runtime adaptation:

1. Elicitation of normative requirements: Is this even possible? How can we deal with subjectivity and the multi-cultural context? Who should participate in the elicitation?
2. What does it mean for normative requirements to be "suitable"?
3. Generalisation of infrastructure for items 1 and 2.
4. How to bridge the gap between engineers & the social scientists?
5. Synthesis of compliant autonomous system behaviour.
6. Verification of compliance.
7. Formal foundations.
8. EthicsOps (adaptation).

In the interest of time, we focussed on items 2, 6, and 7. Our goal in each case was to define why each of the topics was important, and why it was challenging. In the discussion of requirements, we identified a notion of suitability. We say that a set of requirements is suitable when it has the following characteristics:

- of ethical and legal relevance
- affectable by the system
- relevant to stakeholders
- machine understandable
- conflicts are removed or managed
- free of redundancy
- unambiguous
- sufficient, that is, reduces risk of legal or ethical harm
- not overly conservative, that is, does not unnecessarily restrict the services that can be offered.

The difficulty to achieve suitable sets of requirements arises from the fact that it is difficult even for people to decide what is legal and what is ethical, stakeholders from multidisciplinary backgrounds need to be involved, and there may be competing business and institutional interests.

For verification, we identified the importance over and above the usual arguments. Verification can support quantification of risks, minimisation of harmful impacts on values, design or decision space exploration, and evaluation of impact on engineering process. Difficulties arise from the fact that adaptation leads to scalability issues, as the potential set of behaviours at runtime increases. In addition, resilience and anti-fragility require proper consideration of uncertainty. Testing and conformance in general require oracles, but the potentially subjective nature of the requirements and dealing with continuous quantities makes it hard. Finally, the very specification of the new properties (resilience, antifragility) is a challenge.

Some of these aspects were identified as imposing challenges in particular to the use of mathematical foundations in verification, namely, scalability, uncertainty, time, subjectivity, and complexity of specifications. We have finally taken the time to discuss the state of the art, and concluded by looking forward to additional discussions regarding the topics that we did not cover.

4.5 Formal methods for autonomous system resilience and antifragility

Sebastián Uchitel (University of Buenos Aires, AR)
Radu Calinescu (University of York, GB)
Ana Cavalcanti (University of York, GB)
Mario Gleirscher (Universität Bremen, DE)
Lina Marsso (University of Toronto, CA)
Catia Trubiani (Gran Sasso Science Institute – L'Aquila, IT)
Gricel Vázquez (University of York, GB)

The group decided to revisit the definitions of resiliency and antifragility that had been discussed informally the previous day, with the aim of providing a more formal perspective and a relation to specific quality attributes such as performance, safety, and ethics. We also decided that we would ground these definitions on a specific system and a concrete quality to make the discussion and definitions concrete. We aimed to first think of how given two systems, we would compare them with respect to resilience and antifragility, and to postpone the discussion of how such systems may be constructed using formal methods until the end of the session.

The running example we discussed was that of a robot monitoring and interacting with patients in an Emergency Department waiting room at a hospital. Amongst the multiple quality attributes that can be considered in such a system, we decided to address only one to start, assuming that some of the ideas that we would elaborate would then be transferred to a multi-dimensional setting. We chose *patients served per hour* as a quality metric.

Discussions converged to the idea that in order to define resilience and anti-fragility, we should build on the work of Control Theory on how systems recover from disturbances. Thus, we postulated that both resilience and antifragility should be defined (Figure 1) over the

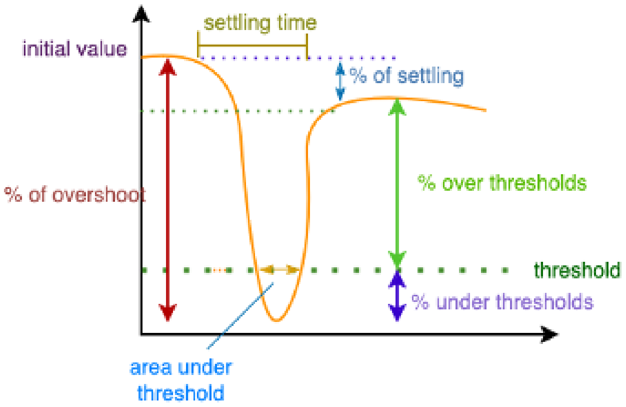


Figure 1 Control-theory inspired metrics for resilience.

metrics of settling time, percentage of settling, percentage of overshoot, and percentage of overshoot on the disruption signals. However, given that we are interested in hard worst-case scenarios too, in addition to the classical set-point concept, we introduced an acceptable threshold value and associated metrics: percentage under threshold, area under threshold, and time under threshold.

We discussed domain-specific examples of resilience defined as summations over disturbances measured as linear combinations settling time and percentage, and antifragility as a comparison between the resilience of two consecutive periods, noting that in this way antifragility can be thought of, informally, as the first derivative of resilience. Another point of discussion is that antifragility must be defined by comparing resilience over periods of time that are long enough to capture statistically relevant sets of disruptions.

Discussion on how to use formal methods to construct systems with these desired properties left many important questions unanswered: What would an appropriate methodology be to guarantee such properties at design time, if a system were to include adaptive mechanisms to achieve antifragility? On what formal methods foundations would it rely upon? Can these properties be decomposed and assigned to different system components to allow for independent construction, incremental improvements, and modular reasoning? Would a hierarchical structure of control loops in which a manager controls for resilience and a "manager of the manager" controls for antifragility, be appropriate?

4.6 Nature-inspired methods for autonomous system resilience and antifragility

Ada Diaconescu (Telecom Paris, FR)
Sebastian Hahner (KIT – Karlsruher Institut für Technologie, DE)
Raffaela Mirandola (KIT – Karlsruher Institut für Technologie, DE
Gabriel Moreno (Carnegie Mellon University – Pittsburgh, US)
Elena Navarro (University of Castilla-La Mancha, ES)
Ralf H. Reussner (KIT – Karlsruher Institut für Technologie, DE)
Patrizia Scandurra (University of Bergamo – Dalmine, IT)

The discussion focused on the following topics:

- Acknowledging nature-inspired approaches already available in related domains;
- Providing natural examples illustrating the trade-off between performance and resilience/antifragility;
- Identifying specific aspects of antifragility and resilience, and determining how they fit within the more general architectures of self-* systems;
- Setting the basis for evaluating and comparing system antifragility capabilities.

Related domains concerned with system autonomy in uncertain environments have already taken inspiration from nature to endow artificial systems with self-* properties (e.g. self-optimisation, -repair, -protection, -configuration, -adaptation). Such self-* capabilities enable systems to improve their performance and dependability at runtime while reducing the need for low-level human intervention. Relevant domains include autonomic, organic and self-aware computing, self-adaptive and control systems, evolutionary computing, swarm robotics, morphogenetic engineering and artificial life.

Within this context, antifragility is concerned with the particular kind of self-* processes that enable systems to self-improve their resilience over time, by capitalising on past experiences – including, but not limited to, self-adaptation to unexpected, rare events. Hence, antifragility processes pertain to the meta-control layer defined within most multi-level self-* system architectures.

Importantly, resilience and anti-fragility compete with performance optimization concerns. Relevant examples include: species specialised for eco-systemic niches versus more versatile species surviving through fluctuating environments; engineered artifacts that rely on specific, highly-integrated components versus loosely-coupled artifacts supporting diverse assemblies; highly-synchronized railway systems maximizing traffic versus less precise ones that tolerate more delays.

With respect to general self-* processes, a system's antifragility is specific in its ability to draw benefits from past reactions to disturbances, so as to improve its future reactions to more or less similar disturbances.

We provide a formalization basis for the above concept as follows (Note: this extends previous works of seminar members and discussions within other groups). Considering a self-* system that reacts to disturbances (e.g. in its environment) that are within a domain E by changing its state (e.g. via a controller C) within a solution domain S. If the system encounters a disturbance outside E, hence within another domain E', then the controller C may no longer be able to find a solution within domain S. In an antifragile system, a meta-controller can search through a wider solution domain (meta-S) and find another controller C', which can adapt the system through states within another subdomain S' in response to disturbances that include those in E'. The system's antifragility-specific support (that we called a "red dot") includes all mechanisms that define the system's maximum solution search domain (meta-S); and that enable the search process from one solution sub-domain (S) to another (S').

We may evaluate a system's antifragility support ("red dot") depending on how much it extends the system's maximum state-space domain (meta-S); and on how effective and efficient is its search process through this domain is (finding S' within meta-S). Finally, antifragility strategies may vary, ranging from brute-force replication and variation (e.g. insects) and all the way to sophisticated predictive approaches (e.g. humans).

4.7 AI solutions for autonomous system resilience and antifragility

Andreas Heyl (Robert Bosch GmbH – Stuttgart, DE)
Simon Burton (University of York, GB)
Felicita Di Giandomenico (CNR – Pisa, IT)
Vincenzo Grassi (University of Rome "Tor Vergata", IT)
Ravi Mangal (Carnegie Mellon University – Pittsburgh, US)
Shiva Nejati (University of Ottawa, CA)

We decompose the problem of using AI for designing resilient and antifragile autonomous systems (RAAS) into four separate questions, namely, "Why should we use AI for RAAS?", "Where should we use AI in RAAS?", "Which types of AI should be used in RAAS and what should they be used for?", and "Where should we start to use AI in RAAS?"

There is a need to use AI in RAAS because these systems are complex and need to handle various sources of uncertainties. AI solutions can be effective in dealing with uncertainty, and are likely to play a key role in transitioning from fail-safe systems to resilient systems and from resilient systems to antifragile systems.

Second, AI can be used at various stages of the RAAS lifecycle, namely, design-time, run-time, and operation-time. While AI can help with developing the systems and simulating their behaviour at design-time, it can also help with monitoring the system for shocks and helping with recovery at run-time and operation-time. Further, AI can be used in different parts of a RAAS. Assuming the standard managed and managing system architecture for RAAS, AI can be an essential component of the managed system (for instance, to perform perception and/or control in a resilient and antifragile cyber-physical system) and/or be a part of the managing system. Finally, resiliency and antifragility need not just be properties of individual systems but could also be desirable properties for systems of systems and entire ecosystems. For instance, while we want an autonomous car to be resilient and antifragile, we can also require these properties to hold for the entire fleet of cars. AI can operate at various levels of this hierarchy; for instance, at the fleet level, AI could help with analysing the large volumes of performance data being collected by the fleet.

Third, focusing on the use of AI in the managing system, we first assume that managing systems in RAAS will follow the standard MAPE-K structure. Given this architecture, AI can help with each step of the MAPE-K cycle. For instance, machine learning models can be used to monitor the state of the system and detect if the system behaviour is outside the nominal bounds. Symbolic AI techniques can be used to analyse and extract the relevant knowledge from the knowledge base for the purpose of planning. An important insight is that each component of the MAPE-K can have different requirements and therefore different forms of AI might be suitable for the different components (data-driven AI for monitoring vs symbolic AI for analysis and planning). We also foresee managing systems that integrate humans and AI. An important question to study is how we can mitigate the limitations of AI such as non-robustness, tendency to hallucinate, and unpredictability when we deploy it in the MAPE-K cycle.

Finally, AI needs to be introduced to RAAS in an incremental fashion, ensuring that the introduction of AI itself does not lead to an increased lack of resiliency or fragility of the system. Towards this end, we need to define effective metrics for evaluating the behaviour of RAAS and continuously measure these metrics to evaluate the effect of AI. In the initial stages, it might be prudent to restrict the use of AI to a limited number of components of a RAAS (for instance, managing the knowledge base in the managing system with a MAPE-K architecture) or for offline analysis of the data collected during RAAS operation. As systems become more complex, ensuring resiliency and antifragility are likely to require the system to perform lifelong learning and potentially necessitate the use of AI in all system components (i.e., in an end-to-end manner).

4.8 Engineering resilient and antifragile autonomous systems

Amel Bennaceur (The Open University – Milton Keynes, GB)
Lee Barford (Keysight Technologies – London, GB)
Matteo Camilli (Politecnico di Milano, IT)
Marc Carwehl (Humboldt-Universität zu Berlin, DE)
Kerstin I. Eder (University of Bristol, GB)
Diego Perez-Palacin (Linnaeus University – Växjö, SE)

Engineering antifragile systems requires specialised consideration in each of the traditional software development process phases. The group explored the Requirements, Design, Implementation and Testing phases, and investigated requirements and KPIs for antifragility, how existing approaches can support these, and their limitations. In particular, antifragility involves learning and adaptation at runtime, in response to disturbances, for all stages of the engineering process.

From a requirements point of view, one challenge is defining suitable specifications that scope problems enough for driving design and testing, while allowing the system to evolve and adapt for future environments. From a design point of view, designs would need to satisfy uncertain specifications, enabling adaptation to new environments. From an implementation point of view, the challenge is striking a balance between scoping the problem to enable assurance, and allowing for adaptation at runtime. From a testing and analysis point of view, the challenge is to provide evidence of whether the system has improved when it has faced unspecified situations, especially when the specifications are uncertain.

5 Birds-of-a-Feather Groups

5.1 Formalising the relation of uncertainty, knowledge, decisions and antifragility

Ralf H. Reussner (KIT – Karlsruher Institut für Technologie, DE)
Simon Burton (University of York, GB)
Felicita Di Giandomenico (CNR – Pisa, IT)
Kerstin I. Eder (University of Bristol, GB)
Vincenzo Grassi (University of Rome "Tor Vergata", IT)
Ravi Mangal (Carnegie Mellon University – Pittsburgh, US)
Raffaela Mirandola (KIT – Karlsruher Institut für Technologie, DE)
Patrizia Scandurra (University of Bergamo – Dalmine, IT)

This birds-of-the-feather session discussed the relation of uncertainty, assumptions of decisions during the development process of cyber-physical systems, and antifragility. We concluded that uncertainty can be modelled as a property of assumptions which are formulated to make a justified decision in the development process. We agreed that the difference of resilience and antifragility is the ability of a system to learn from external events to improve reactions to such events. Such learning can be expressed in a changed uncertainty of the assumptions.

We identified several classes of metrics for antifragility: (i) metrics based on the improved reaction (including an improved quality), (ii) metrics based on the generality of learning, (iii) metrics based on the severity of the events dealt with (if this can be measured independently from the quality degrading impact), and (iv) metrics based on the sensitivity on events (i.e., the effort needed to react).

5.2 Resilience and antifragility of ethics-aware Human-AI collaborations

Amel Bennaceur (The Open University – Milton Keynes, GB)
Lee Barford (Keysight Technologies – London, GB)
Radu Calinescu (University of York, GB)
Ana Cavalcanti (University of York, GB)
Antje Loyal (Continental Automotive Technologies – Frankfurt, DE)
Lina Marsso (University of Toronto, CA)
Elena Navarro (University of Castilla-La Mancha, ES)
Shiva Nejati (University of Ottawa, CA)
Catia Trubiani (Gran Sasso Science Institute – L'Aquila, IT)

Ethics-aware Human-AI collaboration compounds multiple dimensions of uncertainty. First, uncertainty about ethical norms and their operationalisation. Second, uncertainty about human behaviour and values. Third, uncertainty about AI systems themselves, and the incomplete knowledge about the data, parameters, and performance in deployment. Furthermore, those uncertainties are interrelated, and none of their aspects can be considered in isolation. This group focused on unravelling those uncertainties, illustrating them through examples, and investigating how existing reasoning techniques can help support some of those uncertainties.

Starting from eliciting requirements of ethics-aware Human-AI collaboration, one of the challenges is operationalisation into well-specified systems with well-defined capabilities and ethical/functional rules. Another challenge is about the assumptions (and obligations) about how humans interacting with the system behave. One source of disturbance is humans deviating from those assumptions.

We argued for the need for a theory for ethics-aware Human-AI collaboration grounded in mathematical modelling. We explored the reasoning features that need to be supported such as time, probabilities, non-determinism, interaction, and conformance. We also explored some of the techniques which might be used to support that reasoning, such as verification, synthesis, or goal-based requirements engineering. We also reviewed some of the available formalism that might support those reasoning features, including Markov and other stochastic models, hybrid process algebra, and fuzzy description logic. As none of the existing formalisms seems to support the reasoning needed for ethics-aware Human-AI collaboration, the question remains regarding how to integrate/unify and extend those different formalisms to address the different dimensions of uncertainty.

The group also explored the main building blocks for supporting the engineering of ethics-aware Human-AI collaboration based upon architectural patterns and their reification into reference implementation and reusable components. Similarly the reification of mathematical models into tools and domain-specific languages is needed for the specification of the ethics and functional requirements. Finally, standards and guidelines may guide the specification and engineering of those ethics-aware Human-AI collaborations.

5.3 Reasoning about antifragility from a control perspective

Mario Gleirscher (Universität Bremen, DE)
Marc Carwehl (Humboldt-Universität zu Berlin, DE)
Ada Diaconescu (Telecom Paris, FR)
Sebastián Uchitel (University of Buenos Aires, AR)
Gricel Vázquez (University of York, GB)

This birds-of-a-feather group had the objective of *developing a universal notion of antifragility based on the closed-loop control framework* widely applied in control theory and engineering. The discussions were aimed at a method for developing and evaluating controllers for an upcoming next generation of complex adaptive software systems, expected or even required to be increasingly resilient [1].

A collection of autonomous mobile robots working in a hospital was considered as an example following practical trends. These care robots must deliver documents, serve food to patient rooms, and interact with patients and staff.

The group's working hypothesis was that *antifragility* of such an application can be rephrased as a stability property of *resilience* as a quantity measured via an observed quality attribute of the application. This correspondence enables control-theoretic reasoning, for example, verifying whether a particular *adaptation manager* pushes the *resilience error* (i.e., the difference between observed and preferred resilience) below some threshold or whether the resilience level stabilises at a reference value.

During the discussion, we sketched a preliminary formal framework in support of this hypothesis. The framework is based on the notion of a signal, upon which the detection and evaluation of *disruptions* can be defined. The aggregated evaluation of the disruptions should then result in a characterisation of resilience and, moreover, allow one to observe antifragility. In particular, the outlined framework implies the notion of antifragility as the monotonic decrease of the resilience error, respectively, the monotonic increase of resilience over time, relative to a control loop, an adaptation manager, and a resilience profile. Overall, this notion resembles the desire of asymptotic stability of the control loop under consideration.

An important research challenge identified by the group is finding an appropriate adaptation manager for a particular control loop, such that the outlined monotonicity conditions are satisfied. In summary, the proposed control-theoretic perspective of resilience and antifragility enables the utilisation of further tools from control engineering in the search and design of adaptation managers responsible for improving resilience over time.

References

1 Jean-Claude Laprie. From dependability to resilience. In *Dependable Systems and Networks (DSN), 38th IEEE/IFIP Int. Conf.*, pages G8–G9, 2008.

5.4 Uncertainty propagation to support achieving antifragility

Sebastian Hahner (KIT – Karlsruher Institut für Technologie, DE)
Matteo Camilli (Politecnico di Milano, IT)
Andreas Heyl (Robert Bosch GmbH – Stuttgart, DE)
Antje Loyal (Continental Automotive Technologies – Frankfurt, DE)
Gabriel Moreno (Carnegie Mellon University – Pittsburgh, US)
Diego Perez-Palacin (Linnaeus University – Växjö, SE)

Uncertainty Flow Diagrams [1] are a recently proposed syntax, inspired by data flow diagrams and activity diagrams, that allows representing the system from the view of uncertainty to understand the existence of different uncertainties, their propagation along the operations and data flow, and to analyze uncertainty interaction. Some examples of systems that can benefit from the study of uncertainty propagation in their antifragility process implementation are self-adaptive systems such as znn.com and the software in the autonomous driving perception and decision modules. The former can combine different tactics to form strategies enhancing the service quality (e.g., by using different cloud providers, or changing the content quality), thus evolving its adaptation strategy at runtime, which also alters the uncertainty propagation. The latter uses sensor fusion to adapt to different environmental conditions and unanticipated change at runtime, e.g., due to sensor failures.

A possible benefit is to use the results of the uncertainty propagation analysis to measure the system antifragility, together with other metrics. A second benefit is to use the information resulting from an uncertainty propagation upstream analysis to identify the system elements that are increasingly contributing to the effect of uncertainty in the system decisions and trigger a system improvement process focusing on those elements.

In this group, we built on the aforementioned examples to compare three alternative approaches to measure antifragility: quality vs. time, uncertainty propagation depth, and covered conditional space. The first approach is the "classical" way to measure antifragility.

The second approach assumes that uncertainty which is mitigated earlier indicates a more antifragile system. The third approach regards the resilience of the system as being associated with properly handled uncertainty scenarios. Our initial findings indicate that all three approaches are equally suited for measuring antifragility, and can be interchanged. We proposed that approaches like the uncertainty propagation depth can also help the managing system (in a state-of-the-art MAPE-K model) assess and enhance its antifragility. Further research should investigate the flow and combination of different uncertainty sources and representations, and use the outcome of this investigation to build more resilient and antifragile systems.

References

1 Javier Cámara, Sebastian Hahner, Diego Perez-Palacin, Antonio Vallecillo, Maribel Acosta, Nelly Bencomo, Radu Calinescu, and Simos Gerasimou. Uncertainty Flow Diagrams: Towards a Systematic Representation of Uncertainty Propagation and Interaction in Adaptive Systems In *19th International Conference on Software Engineering for Adaptive and Self-Managing Systems*, 2024.

Participants

- Lee Barford
Keysight Technologies – London, GB
- Amel Bennaceur
The Open University – Milton Keynes, GB
- Simon Burton
University of York, GB
- Radu Calinescu
University of York, GB
- Matteo Camilli
Politecnico di Milano, IT
- Marc Carwehl
Humboldt-Universität zu Berlin, DE
- Ana Cavalcanti
University of York, GB
- Felicita Di Giandomenico
CNR – Pisa, IT
- Ada Diaconescu
Telecom Paris, FR
- Kerstin I. Eder
University of Bristol, GB
- Mario Gleirscher
Universität Bremen, DE
- Vincenzo Grassi
University of Rome "Tor Vergata", IT
- Sebastian Hahner
KIT – Karlsruher Institut für Technologie, DE
- Andreas Heyl
Robert Bosch GmbH – Stuttgart, DE
- Antje Loyal
Continental Automotive Technologies – Frankfurt, DE
- Ravi Mangal
Carnegie Mellon University – Pittsburgh, US
- Lina Marsso
University of Toronto, CA
- Raffaela Mirandola
KIT – Karlsruher Institut für Technologie, DE
- Gabriel Moreno
Carnegie Mellon University – Pittsburgh, US
- Elena Navarro
University of Castilla – La Mancha, ES
- Shiva Nejati
University of Ottawa, CA
- Diego Perez-Palacin
Linnaeus University – Växjö, SE
- Ralf H. Reussner
KIT – Karlsruher Institut für Technologie, DE
- Patrizia Scandurra
University of Bergamo – Dalmine, IT
- Catia Trubiani
Gran Sasso Science Institute – L'Aquila, IT
- Sebastián Uchitel
University of Buenos Aires, AR
- Gricel Vázquez
University of York, GB