DAGSTUHL
REPORTS

Volume 7, Issue 11, November 2017

New Challenges in Parallelism (Dagstuhl Seminar 17451)
Annette Bieniusa, Hans-J. Boehm, Maurice Herlihy, and Erez Petrank 1

Algorithmic Cheminformatics (Dagstuhl Seminar 17452)
Jakob L. Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler 28

Connecting Visualization and Data Management Research (Dagstuhl Seminar 17461)
Remco Chang, Jean-Daniel Fekete, Juliana Freire, and Carlos E. Scheidegger 46

A Shared Challenge in Behavioural Specification (Dagstuhl Seminar 17462)
Klaus Havelund, Martin Leucker, Giles Reger, and Volker Stolz 59

Artificial and Computational Intelligence in Games: AI-Driven Game Design (Dagstuhl Seminar 17471)
Pieter Spronck, Elisabeth André, Michael Cook, and Mike Preuß 86

Addressing the Computational Challenges of Personalized Medicine (Dagstuhl Seminar 17472)
Niko Beerenwinkel, Holger Fröhlich, and Susan A. Murphy 130

Reliable Computation and Complexity on the Reals (Dagstuhl Seminar 17481)
Norbert T. Müller, Siegfried M. Rump, Klaus Weihrauch, and Martin Ziegler 142

Dagstuhl Reports, Vol. 7, Issue 11 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
http://www.dagstuhl.de/dagpub/2192-5283

Publication date
March, 2018

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 `http://dnb.d-nb.de`.

Digital Object Identifier: 10.4230/DagRep.7.11.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)*
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
http://www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 17451

New Challenges in Parallelism

Edited by

Annette Bieniusa[1], Hans-J. Boehm[2], Maurice Herlihy[3], and Erez Petrank[4]

1 **TU Kaiserslautern, DE,** bieniusa@cs.uni-kl.de
2 **Google – Palo Alto, US,** hboehm@google.com
3 **Brown University – Providence, US,** mph@cs.brown.edu
4 **Technion – Haifa, IL,** erez@cs.technion.ac.il

Abstract

A continuing goal of current multiprocessor software design is to improve the performance and reliability of parallel algorithms. Parallel programming has traditionally been attacked from widely different angles by different groups of people: Hardware designers designing instruction sets, programming language designers designing languages and library interfaces, and theoreticians developing models of parallel computation. Unsurprisingly, this has not always led to consistent results. Newly developing areas show every sign of leading to similar divergence. This Dagstuhl Seminar will bring together researchers and practitioners from all three areas to discuss and reconcile thoughts on these challenges.

Memory Models and Platforms

Fundamental questions about the semantics of shared memory remain. For example, it becomes increasingly clear that atomic accesses to variables without memory ordering guarantees, or with very weak ordering guarantees, are important in practice. It is surprisingly common to find data structures, such as simple counters, that effectively consist of a single machine word. These continue to be "supported" in languages like Java and C++, but there remains no generally accepted way of defining their semantics, and the specifications in these languages are clearly inadequate. Fundamental questions about memory models and concurrent data structures continue to be unresolved. Many Java concurrent data structures provide weaker than interleaving (”sequentially consistent”) semantics that can only be fully understood with a thorough understanding of the memory model. This fact seems to be neither widely appreciated nor discussed. Are the "acquire/release" semantics often used in practice sufficient? Could we afford the overhead of providing the programmer with a simpler model? The more theoretical side of our discipline often uses concepts, such as "safe" and "regular" registers that are quite foreign to the way in which parallel programming languages are actually defined. Are these notions reconcilable?

Non-Volatile Memory and Concurrency

Non-volatile memory (NVM) technologies are expected to support persistence in byte- addressable memory at densities higher than DRAM and at competitive speed. It is expected that NVM will unify the DRAM and SSD into a one-level storage system of persistent main memory with no need for a hard drive, and directly accessible from the programming language. Such a change in the platforms has a significant impact on the design of software and in particular on concurrent algorithms. One implication is that standard functionalities need to be written for a single-tier memory rather then the standard two-tier paradigm. The design of widely available applications, such as database systems, assume that two-tier memory levels are present, and optimizations are based on the fact that these two memory levels have very different behaviors. Concurrency needs to be re-thought in the presence of the new memory structure. Another implication is

New Challenges in Parallelism, *Dagstuhl Reports*, Vol. 7, Issue 11, pp. 1–27
Editors: Annette Bieniusa, Hans-J. Boehm, Maurice Herlihy, and Erez Petrank
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

that software is now expected to deal with persistence. This has strong connections to thread synchronization issues, but has not been traditionally studied with concurrent algorithms. Addressing these challenges is crucial for building systems on non-volatile memories and we would like to explore potential solutions in the Seminar.

Seminar November 5–10, 2017 – http://www.dagstuhl.de/17451
1998 ACM Subject Classification D.1.3 Concurrent Programming, E.1 Data structures, D.2.4 Software/Program Verification, D.4.2 Storage management
Keywords and phrases concurrency, memory models, non-volatile memory
Digital Object Identifier 10.4230/DagRep.7.11.1
Edited in cooperation with Deepthi Akkoorath

1 Executive Summary

Annette Bieniusa
Hans-J. Boehm
Maurice Herlihy
Erez Petrank

Improving the performance and reliability of parallel and concurrent programs is an ongoing topic in multiprocessor software and hardware research. Despite numerous efforts, the semantics of weak memory models remains subtle and fragile. There has not been established a generally accepted way of defining their semantics, and the specifications of programming languages supporting weak memory models with shared accesses are clearly inadequate. In addition, new advances in hardware are adding further complexity. For example, recently, non-volatile memory (NVM) generated a lot of interest in different communities: Hardware designers coming up with instruction sets and layouts of NVM, system designers integrating NVM into the storage stack, programming language designers proposing library interfaces, and theoreticians developing a new theory of persistence under concurrent access and algorithms adapted for persistency.

This Dagstuhl Seminar on “Future Challenges in Parallelism” touched on different aspects on topics in this broad area of research. In this report, we briefly give a summary of the presentations and discussions that took place during the seminar.

Presentations started with an introductory broad overview talk on the non-volatile memory technology. This survey was followed by shorter talks ranging from hardware techniques for efficiently controlling write-back ordering from caches to theoretical foundations and design of specific data structures.

There was agreement that non-volatile memory is likely to become commercially important in the near future, and that it is tempting to exploit it to provide persistence of user data structures. However, there was little agreement on detailed assumptions and direction. The emphasis of the presentations was on manually designed data structures programmed to a near-hardware-level interface. Some participants expressed concerns that this was too low-level and that the community should instead focus on constructs at the level of durable transactions. Transactional semantics are likely to play an important role: When restarting an application from its persisted state, this state must be consistent in order to prevent data corruption and loss. Much of the work presented in the workshop assumed that we will have non-volatile memory combined with visibly volatile caches that require explicit flush operations to persist data. But it was pointed out that the problem could be greatly simplified by either providing sufficient battery backup to ensure that the entire cache is flushed on failure, or providing other hardware support.

Some of the participants discussed the definitions of correctness. On the one hand, the standard definition of durable linearizability is a strong requirement that typically brings a large performance overhead. On the other hand, the weaker buffered linearizability does not compose well. Other participants suggested some hardware modifications that could make the life of the programmer easier. For example, a discussion emerged on whether we could pin a cache line to make sure it is not written back to memory. We also tackled the programmability of systems with non-volatile memories. How difficult should it be to program them? Are application programmers expected to employ it directly or only via dedicated data structures provided in libraries? The experience report of porting the application memcached to non-volatile memory raised a lot of interest with the participants. It turned out that the task was rather difficult due to complex interactions between the different modules in the application, in particular between modules that required persistence and modules that did not. The lack of tools was strongly felt, and the obtained performance was not satisfactory. The conclusion was that applications had better be redesigned from scratch to work with non-volatile memory. The general feeling at the end of the seminar was that we are in the beginning of exciting times for research on non-volatile memories and that the discussions must and will continue.

Memory models formed the second major thread of presentations and discussions, with participants expressing widely different viewpoints and technical directions. At one extreme, Madan Musuvathi presented evidence that a simple interleaving-based “sequentially consistent” semantics can be provided at reasonable cost, together with an argument that this is a good direction for future programming languages. At the other extreme, Viktor Vafeiadis argued that a weaker “acquire-release” memory model is easier to reason about, an argument that was backed up by model-checking time measurements. Needless to say, this was followed by lively discussion resulting, we believe in at least a more thorough understanding of different perspectives by everyone involved. There were also several brief presentations and extensive discussion on different approaches for addressing the long-standing C++ and Java (among others) out-of-thin-air problem. Current semantics for these languages allow outcomes that are universally accepted as absurd, but which we do not know how to prohibit in any precise way. It is clear that none of the solutions are quite ready to be adopted, but there are encouraging results along several different paths. There is a consensus that this problem makes formal reasoning about programs nearly impossible and that it is a serious obstruction for tool development. There was less consensus about the extent to which it obstructs day-to-day programming efforts.

In conclusion, the seminar inspired discussions and proposed challenging problems to tackle for the research community. As the discussions showed, designing sound and performant parallel systems require the cooperation of researchers on hardware and software level, with both theoretical and practical analyses and evaluations.

2 Table of Contents

Executive Summary

Annette Bieniusa, Hans-J. Boehm, Maurice Herlihy, and Erez Petrank 2

Overview of Talks

Coherence, Consistency, and Deja Vu: Memory Hierarchies in the Era of Specialization
Sarita Adve 7

Global-Local View: Scalable Consistency for Concurrent Data Types
Deepthi Devaki Akkoorath 7

Remote Memory References at Block Granularity
Hagit Attiya 8

Analyzing Contention and Backoff in Asynchronous Shared Memory
Naama Ben-David and Guy E. Blelloch 8

Concurrency as First-class Entity
Annette Bieniusa 9

What consistency guarantees should concurrent data structure libraries provide?
Hans-J. Boehm 9

Remote memory in the age of fast networks
Irina Calciu 10

NVM ReConstruction: Object-Oriented Recovery for Non-Volatile Memory
Nachshon Cohen 10

Waiting Policies
Dave Dice 11

Performance Isolation on Modern Multi-Socket Systems
Sandhya Dwarkadas 11

Efficient Distributed Data Structures for Future Many-core Architectures
Panagiota Fatourou, Nikolaos D. Kallimanis, Eleni Kanellou, Odysseas Makridakis, and Christi Symeonidou 12

Persistent Lock-Free Data Structures for Non-Volatile Memory
Michal Friedman 12

Everything Better Than Everything Else
Tim Harris 13

Recoverable Mutual Exclusion in Sub-logarithmic Time
Danny Hendler and Wojceich Golab 13

Hybrid STM/HTM for Java
Antony Hosking 14

Concurrent data structures for scalable real-time analytics
Idit Keidar 14

Generic Concurrency Restriction
Alex Kogan 15

Flat Combining and Transactional Lock Elision – combining pessimistic and optimistic mechanisms together
Yossi Lev 15

Verifying Concurrent GC Running in Weakly Ordered Memories
J. Eliot B. Moss 16

How much does sequential consistency cost anyway
Madan Musuvathi 16

Efficient Architectural Support for Persistent Memory
Vijay Nagarajan 17

Concurrent Data structures for Non-Volatile Memory
Erez Petrank 17

Efficient Inspected Critical Sections in data-parallel GPU codes
Michael Philippsen and Thorsten Blass 17

The old challenge: How to support users?
Mirko Rahn 18

Durable Linearizability
Michael L. Scott 18

Just-Right Consistency: As available as possible, consistent when necessary
Marc Shapiro, Annette Bieniusa, Christoper Meiklejohn, Nuno Preguiça, and Valter Balegas 19

Causal atomicity
Simon Doherty, John Derrick, Brijesh Dongol, Heike Wehrheim 19

Chasing Away RAts: Semantics and Evaluation for Relaxed Atomics on Heterogeneous Systems
Matthew Sinclair 20

Memory Instrumentation As A First-Class Language Feature
Michael F. Spear 20

Sequential consistency considered harmful
Viktor Vafeiadis 21

Applying lock-free data structures to fabric-attached memory
Haris Volos 22

Isoefficiency in Practice: Configuring and Understanding the Performance of Task-based Applications
Felix Wolf 22

Effect Summaries for Thread-Modular Analysis
Sebastian Wolff, Lukáš Holík, Roland Meyer, and Tomáš Vojnar 23

Working groups

Distributed Concurrency
Michal Friedman and Virendra Marathe 23

Relaxed Atomicity
Madan Musuvathi 24

Panel discussions

Panel Discussion: Concurrency vs. Parallelism
Matthew Sinclair 25

Participants 27

3 Overview of Talks

3.1 Coherence, Consistency, and Deja Vu: Memory Hierarchies in the Era of Specialization

Sarita Adve (University of Illinois – Urbana-Champaign, US)

Joint work of Matthew D. Sinclair, Johnathan Alsop, Sarita Adve, and many others

Main reference Matthew D. Sinclair, Johnathan Alsop, Sarita V. Adve: "Efficient GPU synchronization without scopes: saying no to complex consistency models", in Proc. of the 48th International Symposium on Microarchitecture, MICRO 2015, Waikiki, HI, USA, December 5-9, 2015, pp. 647–659, ACM, 2015.

URL http://dx.doi.org/10.1145/2830772.2830821

Memory models for homogeneous multicore systems have traced an arc of complexity that was often precipitated by hardware designs that did not pay enough attention to programmabilty or portability. As we now enter the brave new world of heterogeneous computing and specialization, it is deja vu. We describe how hardware designers are again on a trajectory of exposing too much hardware to software, resulting in complex hierarchies and consistency models. We draw on results from the DeNovo project to show that this trajectory is neither necessary nor effective. With appropriately designed coherence and careful interfaces, we can reap the efficiency benefits of specialization while retaining a simple memory model.

3.2 Global-Local View: Scalable Consistency for Concurrent Data Types

Deepthi Devaki Akkoorath (TU Kaiserslautern, DE)

Joint work of Deepthi Devaki Akkoorath, Annette Bieniusa, Carlos Baquero, José Brandão

Concurrent linearizable access to shared objects can be prohibitively expensive in a high contention workload. Many applications apply ad-hoc techniques to eliminate the need of synchronous atomic updates, which may result in non-linearizable implementations. We propose a *global-local view* model which leverages such patterns for concurrent access to objects in a shared memory system. In this model, each thread maintains different views on the shared object – a thread-local view and a global view. As the thread-local view is not shared, it can be updated without incurring synchronization costs. These local updates become visible to other threads only after the thread-local view is merged with the global view. By executing operations on the local state without synchronization, while only synchronizing with the shared state when needed, applications can achieve better scalability at the expense of linearizability – the default correctness criteria for concurrent objects.

3.3 Remote Memory References at Block Granularity

Hagit Attiya (Technion – Haifa, IL)

Joint work of Hagit Attiya, Gili Yavneh
Main reference To appear in OPODIS 2017.

The cost of accessing shared objects that are stored in remote memory, while neglecting accesses to shared objects that are cached in the local memory, can be evaluated by the number of remote memory references (RMRs) in an execution. We propose a new measure, called block RMRs, counting the number of remote memory references while taking into account the fact that shared objects can be grouped into blocks. On the one hand, this measure reflects the fact that the RMR incurred for bringing a shared object to the local memory might save another RMR for bringing another object placed at the same block. On the other hand, this measure accounts for false sharing: the fact that an RMR may be incurred when accessing an object due to a concurrent access to another object in the same block.

3.4 Analyzing Contention and Backoff in Asynchronous Shared Memory

Naama Ben-David (Carnegie Mellon University – Pittsburgh, US) and Guy E. Blelloch

Joint work of Naama Ben-David, Guy E. Blelloch
Main reference Naama Ben-David, Guy E. Blelloch: "Analyzing Contention and Backoff in Asynchronous Shared Memory", in Proc. of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017, pp. 53–62, ACM, 2017.
URL http://dx.doi.org/10.1145/3087801.3087828

Randomized backoff protocols have long been used to reduce contention on shared resources. They are heavily used in communication channels and radio networks, and have also been shown to greatly improve the performance of shared memory algorithms in real systems. However, while backoff protocols are well understood in many settings, their effect in shared memory has never been theoretically analyzed. This discrepency may be due to the difficulty of modeling asynchrony without eliminating the advantage gained by local delays.

In this talk, I will present a new cost model for contention in shared memory, which allows restricted adversarial asynchrony while capturing the effect of process delays. Using this model, I'll show that we can asymptotically separate the performance of a read-modify-write loop with and without exponential backoff, demonstrating that the model reflects a phenomenon that has evaded rigorous characterization in the past. I'll also introduce a new backoff protocol based on adaptive write-probabilities and show that this protocol outperforms classic exponential backoff under our model.

3.5 Concurrency as First-class Entity

Annette Bieniusa (TU Kaiserslautern, DE)

Joint work of Mathias Weber, Annette Bieniusa, Arnd Poetzsch-Heffter
Main reference Mathias Weber, Annette Bieniusa, Arnd Poetzsch-Heffter: "EPTL - A Temporal Logic for Weakly Consistent Systems (Short Paper)", in Proc. of the Formal Techniques for Distributed Objects, Components, and Systems - 37th IFIP WG 6.1 International Conference, FORTE 2017, Held as Part of the 12th International Federated Conference on Distributed Computing Techniques, DisCoTec 2017, Neuchâtel, Switzerland, June 19-22, 2017, Proceedings, Lecture Notes in Computer Science, Vol. 10321, pp. 236–242, Springer, 2017.
URL https://doi.org/10.1007/978-3-319-60225-7_17

Concurrency is a natural phenomenon. Information is essentially local to a process and becomes only visible once processes interact and communicate. Yet, programming models are typically choosing linearizability as standard semantics.

In my talk I am presenting a novel temporal logic, event-based parallel temporal logic (EPTL), that allows to reason about weakly-consistent systems. In contrast to other temporal logics like LTL or CTL, EPTL allows to model semantics of components that are truly concurrent while abstracting from implementation and communication details.

3.6 What consistency guarantees should concurrent data structure libraries provide?

Hans-J. Boehm (Google – Palo Alto, US)

Main reference Hans-J. Boehm, P0387R0: Memory Model Issues for Concurrent Data Structures (WG21 standards committee paper)
URL http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2016/p0387r0.html

The C++ standards committee is is attempting to add concurrent data structures, for example concurrent queues, to the standard library. This has raised interesting questions about desired correctness properties for such libraries. The traditional linearizability criterion is not entirely consistent with the underlying C++ memory model, and does not sufficiently address interactions with visibility of simple assignments. This is complicated by the fact that there is little agreement on the correct visibility guarantees, and that they seem to vary between data structures. We briefly addressed some of the issues, tradeoffs, and challenges.

3.7 Remote memory in the age of fast networks

Irina Calciu (VMware – Palo Alto, US)

Joint work of Marcos K. Aguilera, Nadav Amit, Irina Calciu, Xavier Deguillard, Jayneel Gandhi, Pratap Subrahmanyam, Lalith Suresh, Kiran Tati, Rajesh Venkatasubramanian, Michael Wei
Main reference Marcos K. Aguilera, Nadav Amit, Irina Calciu, Xavier Deguillard, Jayneel Gandhi, Pratap Subrahmanyam, Lalith Suresh, Kiran Tati, Rajesh Venkatasubramanian, Michael Wei: "Remote memory in the age of fast networks", in Proc. of the 2017 Symposium on Cloud Computing, SoCC 2017, Santa Clara, CA, USA, September 24 - 27, 2017, pp. 121–127, ACM, 2017.
URL http://dx.doi.org/10.1145/3127479.3131612

As the latency of the network approaches that of memory, it becomes increasingly attractive for applications to use remote memory–random-access memory at another computer that is accessed using the virtual memory subsystem. This is an old idea whose time has come, in the age of fast networks. To work effectively, remote memory must address many technical challenges. We hope to provide a broad research agenda around this topic, by proposing more problems than solutions.

3.8 NVM ReConstruction: Object-Oriented Recovery for Non-Volatile Memory

Nachshon Cohen (EPFL – Lausanne, CH)

Joint work of Nachshon Cohen, Virendra Marathe, James Larus

New non-volatile memory (NVM) technologies allow direct, durable storage of data in an application's heap. This type of random-access memory can simplify the construction of reliable applications that do not lose data at a system shutdown or power failure. Existing NVM programming frameworks for native languages are applicable to non-object oriented languages such as C, and do not gracefully support richer abstractions and constructs available in an object-oriented language like C++. Support for even mundane abstractions, such as pointers, leads to highly error prone programming practices. This paper presents a new persistent memory programming model designed to naturally align with the object-oriented programming style, while addressing programming pitfalls of prior approaches. At the heart of our approach is the notion of *object reconstruction*, the ability to transparently reconstruct a persistent object's state at runtime after a restart event. Object reconstruction enables several desirable features that significantly simplify the programmer's task of writing correct programs that leverage byte-addressability of persistent memory. It (i) enables support for key object-oriented features such as type inheritance and virtual functions in persistent types, (ii) accommodates co-location of nonpersistent fields within persistent type instances, (iii) allows representation of persistent pointers as virtual addresses, and (iv) enables type specific reconstruction of the entire state of a persistent object after a restart. Our prototype implementation as a C++ library demonstrates the versatility of object reconstruction at virtually zero performance overhead.

3.9 Waiting Policies

Dave Dice (Oracle Corp. – Burlington, US)

Main reference Dave Dice: "Malthusian Locks", CoRR, Vol. abs/1511.06035, 2015.
URL http://arxiv.org/abs/1511.06035

The "best practice" for waiting – threads waiting for another thread to change a location in shared memory – has shifted with modern architectures. We briefly survey the motivating architectural factors and suggest new practices.

3.10 Performance Isolation on Modern Multi-Socket Systems

Sandhya Dwarkadas (University of Rochester, US)

Joint work of Sharanyan Srikanthan and Kai Shen
URL http://www.cs.rochester.edu/u/sandhya

Recognizing that parallel applications are rarely executed in isolation today, I will discuss some practical challenges in making best use of available hardware and our approach to addressing these challenges. I describe two independent control mechanisms: a sharing- and resource-aware mapper (SAM) to effect task placement with the goal of localizing shared data communication and minimizing resource contention based on the offered load; and an application parallelism manager (MAP) that controls the offered load with the goal of improving system parallel efficiency. Our results emphasize the need for low-overhead monitoring of application behavior under changing environmental condiitons in order to adapt to environment and application behavior changes.

References

1 Coherence Stalls or Latency Tolerance: Informed CPU Scheduling for Socket and Core Sharing, Sharanyan Srikanthan, Sandhya Dwarkadas, and Kai Shen, Usenix Annual Technical Conference (Usenix ATC), Denver, CO, June 2016.

2 Data Sharing or Resource Contention: Toward Performance Transparency on Multicore Systems, Sharanyan Srikanthan, Sandhya Dwarkadas, and Kai Shen, Usenix Annual Technical Conference (Usenix ATC), Santa Clara, CA, July 2015.

3.11 Efficient Distributed Data Structures for Future Many-core Architectures

Panagiota Fatourou (University of Crete – Heraklion, GR), Nikolaos D. Kallimanis, Eleni Kanellou, Odysseas Makridakis, and Christi Symeonidou

Main reference Panagiota Fatourou, Nikolaos D. Kallimanis, Eleni Kanellou, Odysseas Makridakis, Christi Symeonidou: "Efficient Distributed Data Structures for Future Many-Core Architectures", in Proc. of the 22nd IEEE International Conference on Parallel and Distributed Systems, ICPADS 2016, Wuhan, China, December 13-16, 2016, pp. 835–842, IEEE Computer Society, 2016.
URL http://dx.doi.org/10.1109/ICPADS.2016.0113

We study general techniques for implementing distributed data structures on top of future many-core architectures with non cache-coherent or partially cache-coherent memory. With the goal of contributing towards what might become, in the future, the concurrency utilities package in Java collections for such architectures, we end up with a comprehensive collection of data structures by considering different variants of these techniques. To achieve scalability, we study a generic scheme which makes all our implementations hierarchical. We also describe a collection of techniques for further improving scalability in most implementations. We have performed experiments which illustrate nice scalability characteristics for some of the proposed techniques and reveal the performance and scalability power of the hierarchical approach. We distill the experimental observations into a metric that expresses the scalability potential of such implementations. We finally present experiments to study energy consumption aspects of the proposed techniques by using an energy model recently proposed for such architectures.

3.12 Persistent Lock-Free Data Structures for Non-Volatile Memory

Michal Friedman (Technion – Haifa, IL)

Joint work of Michal Friedman, Erez Petrank, Maurice Herlihy, Virendra Marathe, Nachshon Cohen
Main reference Michal Friedman, Maurice Herlihy, Virendra J. Marathe, Erez Petrank: "Brief Announcement: A Persistent Lock-Free Queue for Non-Volatile Memory", in Proc. of the 31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria, LIPIcs, Vol. 91, pp. 50:1–50:4, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.DISC.2017.50

Non-volatile memory is expected to coexist with (or even displace) volatile DRAM for main memory in upcoming ar- chitectures. This has led to increasing interest in the problem of designing and specifying durable data structures that can recover from system crashes. Data structures may be designed to satisfy stricter or weaker durability guarantees to provide a balance between the strength of the provided guarantees and performance overhead. The talk proposes three novel implementations of a concurrent lock-free data structures. These implementations illustrate algorithmic challenges in building persistent lock-free data structures with different levels of durability guarantees. In presenting these challenges, the pro- posed algorithmic designs, and the different durability guaran- tees, we hope to shed light on ways to build a wide variety of durable data structures.

3.13 Everything Better Than Everything Else

Tim Harris (Oracle Labs – Cambridge, GB)

Evaluating shared-memory data structures is difficult: there are lots of possible metrics, lots of possible ways to run experiments, and lots of ways in which low-level details of the hardware can have unexpectedly large impacts on performance. It is hard to untangle algorithmic improvements from coding prowess. I am going to talk about some of the ways in which I have been bitten by these problems in the past, and some of the techniques I use to try to organize my own experimental work.

3.14 Recoverable Mutual Exclusion in Sub-logarithmic Time

Danny Hendler (Ben Gurion University – Beer Sheva, IL) and Wojceich Golab

Joint work of Danny Hendler, Wojciech Golab
Main reference Wojciech M. Golab, Danny Hendler: "Recoverable Mutual Exclusion in Sub-logarithmic Time", in Proc. of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017, pp. 211–220, ACM, 2017.
URL http://dx.doi.org/10.1145/3087801.3087819

Recent developments in non-volatile main memory (NVRAM) media foreshadow the eventual convergence of primary and secondary storage into a single layer in the memory hierarchy that combines the performance benefits of conventional main memory with the durability of secondary storage. Traditional log-based recovery techniques can be applied correctly in such systems but fail to take full advantage of the parallelism enabled by allowing processing cores to access recovery data directly using memory operations rather than slow block transfers from secondary storage. As a result, harnessing the performance benefts of NVRAM-based platforms requires a careful rethinking of recovery mechanisms. Recoverable mutual exclusion (RME) is a variation on the classic mutual exclusion (ME) problem that allows processes to crash and recover. Prior work on the RME problem has established an upper bound of O(log N) remote memory references (RMRs) in an asynchronous shared memory model with N processes that communicate using atomic read and write operations, prompting the question whether sub-logarithmic RMR complexity is attainable using commonly supported read-modify-write primitives. We answer this question positively by presenting an RME algorithm that incurs O(log N/ log log N) RMRs in the cache-coherent model and uses standard read, write, Fetch-And-Store, and Compare-And-Swap instructions. The algorithm uses as a building block a new recoverable extension of Mellor-Crummey and Scott's queue lock that is interesting in its own right. We also present an O(1) RMRs algorithm that relies on double-word Compare-And-Swap and a double-word variation of Fetch-And-Store.

3.15 Hybrid STM/HTM for Java

Antony Hosking (Australian National University – Canberra, AU)

Joint work of Keith Chapman, Antony Hosking, Eliot Moss
Main reference Keith Chapman, Antony L. Hosking, J. Eliot B. Moss: "Hybrid STM/HTM for nested transactions on OpenJDK", in Proc. of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2016, part of SPLASH 2016, Amsterdam, The Netherlands, October 30 - November 4, 2016, pp. 660–676, ACM, 2016.
URL http://dx.doi.org/10.1145/2983990.2984029

Our work on hybrid STM/HTM for Java allows concurrently executing transactions to use low-cost HTM when it works, but revert to STM when it doesn't, even as other transactions are running STM/HTM. Our implementation is for an extension of Java having syntax for both open and closed nested transactions, and boosting, running on the OpenJDK. We demonstrate that HTM offers significant acceleration of both closed and open nested transactions, while yielding parallel scaling up to the limits of the hardware, whereupon scaling in software continues but with the usual penalty to throughput imposed by software mechanisms.

A useful takeaway from our work is the extent that open nesting enables more profitable use of HTM.

3.16 Concurrent data structures for scalable real-time analytics

Idit Keidar

Joint work of Dmitry Basin, Edward Bortnikov, Anastasia Braginsky, Guy Golan-Gueta, Eshcar Hillel, H. Porat, A. Spiegelman, Moshe Sulamy
Main reference Dmitry Basin, Edward Bortnikov, Anastasia Braginsky, Guy Golan-Gueta, Eshcar Hillel, Idit Keidar, Moshe Sulamy: "KiWi: A Key-Value Map for Scalable Real-Time Analytics", in Proc. of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Austin, TX, USA, February 4-8, 2017, pp. 357–369, ACM, 2017.
URL http://dl.acm.org/citation.cfm?id=3018761

Modern big data processing platforms employ huge in-memory key-value (KV) maps. Their applications simultaneously drive high-rate data ingestion and large-scale analytics. These two scenarios expect KV-map implementations that scale well with both real-time updates and large atomic scans triggered by range queries.

I will discuss research efforts at Yahoo Research addressing this challenge in the context of Druid, a high-performance, column-oriented, distributed data store. I will first discuss recently published work on KiWi – A Key-Value Map for Scalable Real-Time Analytics [1]. I will then discuss two on going projects: Oak – Off-Heap Allocated Keys, and Concurrent Data Sketches.

References

1 Dmitry Basin, Edward Bortnikov, Anastasia Braginsky, Guy Golan-Gueta, Eshcar Hillel, Idit Keidar, and Moshe Sulamy. Kiwi: A key-value map for scalable real-time analytics. In Sarkar and Rauchwerger [2], pages 357–369.

2 Vivek Sarkar and Lawrence Rauchwerger, editors. *Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Austin, TX, USA, February 4-8, 2017*. ACM, 2017.

3.17 Generic Concurrency Restriction

Alex Kogan (Oracle Labs – Burlington, US)

Joint work of Dave Dice, Alex Kogan

Contended locks often degrade the performance of a multithreaded application, leading to a so-called scalability collapse problem. This problem arises when a growing number of threads circulating through a saturated lock causes the overall application performance to fade or even drop abruptly. In this talk, I will introduce GCR (generic concurrency restriction), a mechanism that aims to avoid the scalability collapse. GCR, designed as a generic, lock-agnostic wrapper, intercepts lock acquisition calls, and decides when threads would be allowed to proceed with the acquisition of the underlying lock. Furthermore, I will describe GCR-NUMA, an adaptation of GCR for non-uniform memory access (NUMA) settings.

3.18 Flat Combining and Transactional Lock Elision – combining pessimistic and optimistic mechanisms together

Yossi Lev (Oracle Labs – Burlington, US)

Joint work of Alex Kogan, Yossi Lev
Main reference Alex Kogan, Yossi Lev: “Transactional Lock Elision Meets Combining”, in Proc. of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017, pp. 231–240, ACM, 2017.
URL http://dx.doi.org/10.1145/3087801.3087838

Flat combining (FC) is a technique that can significantly improve the performance of operations that conflict with each other when accessing a shared data structure – e.g., they all trying to modify the same field of the data structure. Transactional Lock Elision (TLE), on the other hand, significantly improve the performance when the operations applied to the data structure rarely conflict – that is, in the common case, they access different parts of the data structures; this is done by using an optimistic approach where hardware transactions execute the operations in parallel, and retry when they conflict with one another. Both techniques provides the desired property of allowing the programmer to write simple, sequential code that does not need to handle interference by other operations running in parallel – as if the operations are ran under a lock protecting all accesses to the data structure. In this presentation I show how we can combine the two techniques when dealing with data structures where some of their operations are likely to conflicts with each other, while others do not, and can there run in parallel using transactional lock elision. The new technique keeps the simplicity of programming without needing to reason about interference with other operations running in parallel, and significantly improves the performance comparing to use either of these techniques by its own.

3.19 Verifying Concurrent GC Running in Weakly Ordered Memories

J. Eliot B. Moss (University of Massachusetts – Amherst, US)

We are in the process of developing a suite of state-of-the-art concurrent garbage collectors for a language-independent virtual machine, intended to simply the implementation of managed languages. This VM, called Mu (for its efforts to be small, i.e., a "micro" VM), has an intermediate language similar to LLVM (close to hardware, but with unbounded local "registers" and SSA-form, and supporting C/C++-style memory-access ordering specifications) but tailored and extended for managed language support in a concurrent environment. Mu is targeting a range of modern processors, including x86 and ARM. Thus the GC must deal with hardware that reorders memory accesses.

Challenges in mechanical verification of this suite of GCs include the sheer size and subtlety of the necessary code, and he desire to reuse, as much as possible, proofs when dealing with code variations of largely local impact. What kinds of specifications and logics will aid in this effort? What proof strategies will make the size of the tasks manageable?

3.20 How much does sequential consistency cost anyway

Madan Musuvathi (Microsoft Research – Redmond, US)

Joint work of Madan Musuvathi, Lun Liu, Dan Marino, Todd Millstein, Satish Narayanaswamy, Ryan Newton, Ryan Scott, Abhay Singh, Michael Vollmer
Main reference Lun Liu, Todd D. Millstein, Madanlal Musuvathi: "A volatile-by-default JVM for server applications", PACMPL, Vol. 1(OOPSLA), pp. 49:1–49:25, 2017.
URL http://dx.doi.org/10.1145/3133873

Research on weak memory-models assume that sequential consistency (SC) is expensive for modern languages on modern hardware. Over the past few years, we have attempted to empirically answer the question: how much does SC cost anyway? While some of the results are what one would expect, others are surprising, at least to us. In this talk I will describe these results and what they might imply:

a) turning off C/C++ compiler optimizations that violate SC results in little performance overhead. This implies that complexities in memory-model design that solely arise due to compiler optimizations can be avoided

b) when done carefully, SC for Java results in an overhead of 12-28% on Intel architectures, and recent experiments suggest similar overheads on ARM architectures as well. Significant part of this overhead comes from inserting hardware fences in a few core libraries. This suggests that Java should allow programmers to escape into relaxed semantics only for carefully chosen code segments, just as it allows programmers to selectively escape into type-unsafe segments.

c) SC has no perceptible overheads for Haskell and possibly for other functional programming languages. This might imply that the strict type isolation that these languages provide between pure and imperative parts of a program could be incorporated into mainstream languages to simplify memory models.

3.21 Efficient Architectural Support for Persistent Memory

Vijay Nagarajan (University of Edinburgh, GB)

Joint work of Arpit Joshi, Vijay Nagarajan, Stratis Viglas, Marcelo Cintra
Main reference Arpit Joshi, Vijay Nagarajan, Stratis Viglas, Marcelo Cintra: "ATOM: Atomic Durability in Non-volatile Memory through Hardware Logging", in Proc. of the 2017 IEEE International Symposium on High Performance Computer Architecture, HPCA 2017, Austin, TX, USA, February 4-8, 2017, pp. 361–372, IEEE Computer Society, 2017.
URL http://dx.doi.org/10.1109/HPCA.2017.50
Main reference Arpit Joshi, Vijay Nagarajan, Marcelo Cintra, Stratis Viglas: "Efficient persist barriers for multicores", in Proc. of the 48th International Symposium on Microarchitecture, MICRO 2015, Waikiki, HI, USA, December 5-9, 2015, pp. 660–671, ACM, 2015.
URL http://dx.doi.org/10.1145/2830772.2830805

Emerging persistent memory technologies enable fast, fine-grained durability compared to slow block-based devices. Programming with persistent memory, however, requires primitives that provide guarantees about what has been made durable. Two primitives that programmers understand well are: ordering (persist barrier) and atomicity (atomic durable transactions). A persist barrier guarantees that stores before the barrier, update persistent memory before stores that follow it. An atomic durable transaction guarantees that stores within the transaction update persistent memory atomically. In this talk, we will show how these two primitives can be efficiently implemented using architectural support.

3.22 Concurrent Data structures for Non-Volatile Memory

Erez Petrank (Technion – Haifa, IL)

Joint work of Erez Petrank, Michal Friedman, Nachshon Cohen, Maurice Herlihy, Virendra Marathe

Non-volatile memory is expected to coexist with (or even displace) volatile DRAM for main memory in upcoming architectures. This has led to increasing interest in the problem of designing and specifying durable data structures that can recover from system crashes. Definitions for durable linearizability have been recently proposed by Israelevitz et al. They have also proposed an automatic transformation for making lock-free data structures resilient to crashed on non-volatile memories. However, the automatic construction is typically inefficient. In this lecture we report an effort to design data structures that are appropriate for this setting, and also discuss some additional definitions.

3.23 Efficient Inspected Critical Sections in data-parallel GPU codes

Michael Philippsen (Universität Erlangen-Nürnberg, DE) and Thorsten Blass

Main reference Thorsten Blass, Michael Philippsen, Ronald Veldema: "Efficient Inspected Critical Sections in data-parallel GPU codes", in Proc. of the 30th Int'l Workshop on Languages and Compilers for Parallel Computing (LCPC 2017), to be published as LNCS volume, Springer, 2018.

Optimistic concurrency control and STMs rely on the assumption of sparse conflicts. For data-parallel GPU codes with many or with dynamic data dependences, a pessimistic and lock-based approach may be faster, if only GPUs would offer hardware support for GPU-wide fine-grained synchronization. Instead, current GPUs inflict dead- and livelocks on attempts to implement such synchronization in software.

The paper demonstrates how to build GPU-wide non-hanging critical sections that are as easy to use as STMs but also get close to the performance of traditional fine-grained locks. Instead of sequentializing all threads that enter a critical section, the novel programmer-guided Inspected Critical Sections (ICS) keep the degree of parallelism up. As in optimistic approaches threads that are known not to interfere, may execute the body of the inspected critical section concurrently.

3.24 The old challenge: How to support users?

Mirko Rahn (Fraunhofer ITWM – Kaiserslautern, DE)

In High Performance Computing many application developers are experts in their current domain rather than in computer science. To deal with parallelism adds a layer of complexity that is not easy to manage for them. We show some common misconceptions that occur in real life programs. Some of the bugs are easy to spot for well trained readers. So where are the tools that support the users?

3.25 Durable Linearizability

Michael L. Scott (University of Rochester, US)

Joint work of Joseph Izraelevitz, Hammurabi Mendes, Michael L. Scott
Main reference Joseph Izraelevitz, Hammurabi Mendes, Michael L. Scott: "Linearizability of Persistent Memory Objects Under a Full-System-Crash Failure Model", in Proc. of the Distributed Computing - 30th International Symposium, DISC 2016, Paris, France, September 27-29, 2016. Proceedings, Lecture Notes in Computer Science, Vol. 9888, pp. 313–327, Springer, 2016.
URL https://doi.org/10.1007/978-3-662-53426-7_23

The prospect of ubiquitous nonvolatile main memory suggests the possibility of maintaining long-lived data unmediated by the file system, but only if data is carefully managed to ensure consistency in the wake of a crash. In keeping with "real world" systems, we introduce the notion of *durable linearizability* to govern the safety of concurrent objects when all transient state (of all threads) is lost on a crash; we also introduce a buffered variant in which the recoverable state is consistent but not necessarily up to date. At the implementation level, we present *explicit epoch persistency*, a formal model that builds upon and generalizes prior work, together with an automated transform to convert any linearizable, nonblocking but data-race–free concurrent object into one that is (buffered) durably linearizable. We also present a design pattern, analogous to linearization points, for the construction of other, more optimized objects. On top of this formal foundation, we present a series of software mechanisms–JUSTDO logging, iDO logging, and periodic persistence–for fast, composable persistence.

3.26 Just-Right Consistency: As available as possible, consistent when necessary

Marc Shapiro (University Pierre & Marie Curie – Paris, FR), Annette Bieniusa, Christoper Meiklejohn, Nuno Preguiça, and Valter Balegas

In a distributed data store, the CAP theorem forces a choice between strong consistency (CP) and availability and responsiveness (AP) when the network can partition. To address this issue, we take an application-driven approach, Just-Right Consistency (JRC). JRC defines a consistency model that is sufficient to maintain the application's specific invariants, and otherwise remaining as available as possible.

JRC leverages knowledge of the application. Two invariant-maintaining programming patterns, ordered updates and atomic grouping, are compatible with asynchronous updates, orthogonally to CAP. In contrast, checking a data precondition on replicated state is CAP-sensitive. However, if two updates do not negate each other's precondition, they may safely execute concurrently. Updates must synchronise only if one negates the precondition of the other.

The JRC approach is supported by the CRDT data model that ensures that concurrent updates converge; by Antidote, a cloud-scale CRDT data store that guarantees transactional causal consistency; and by developer tools (static analysers and domain-specific languages) that help guarantee invariants.

3.27 Causal atomicity

Simon Doherty, John Derrick, Brijesh Dongol, Heike Wehrheim

Correctness conditions for concurrent objects (like linearizability or opacity) require "seemingly atomic", though concurrent, access to shared memory. So far, these correctness conditions build on an interleaving model of concurrency: concurrent executions are represented as totally ordered sequences of invocations and returns of operations (so called histories). Behind this concept of histories is the assumption of a notion of global time and the total observability of orderings of operations. This assumption, however, fails to hold for weak memory models in which there are only partial orders on operations, as for instance generated by a happens-before relation.

We propose a generalization of concurrent correctness conditions to weak memory called causal atomicity. It is based on Lamport's execution structures which are sets of events together with two relations: a partially ordered precedence relation together with a relation describing communication among events (like reads-from relations). Causal atomicity compares execution structures to sequential specifications. Alike linearizability and opacity, we have shown causal atomicity to be compositional in that the composition of causally atomic objects yields causally atomic structures.

3.28 Chasing Away RAts: Semantics and Evaluation for Relaxed Atomics on Heterogeneous Systems

Matthew Sinclair (AMD Research – Bellevue, US)

Joint work of Matthew D. Sinclair, Johnathan Alsop, Sarita V. Adve
Main reference Matthew D. Sinclair, Johnathan Alsop, Sarita V. Adve: "Chasing Away RAts: Semantics and Evaluation for Relaxed Atomics on Heterogeneous Systems", in Proc. of the 44th Annual International Symposium on Computer Architecture, ISCA 2017, Toronto, ON, Canada, June 24-28, 2017, pp. 161–174, ACM, 2017.
URL http://dx.doi.org/10.1145/3079856.3080206

An unambiguous and easy-to-understand memory consistency model is crucial for ensuring correct synchronization and guiding future design of heterogeneous systems. In a widely adopted approach, the memory model guarantees sequential consistency (SC) as long as programmers obey certain rules. The popular data-race-free-0 (DRF0) model exemplifies this SC-centric approach by requiring programmers to avoid data races. Recent industry models, however, have extended such SC-centric models to incorporate relaxed atomics. These extensions can improve performance, but are difficult to specify formally and use correctly. This work addresses the impact of relaxed atomics on consistency models for heterogeneous systems in two ways. First, we introduce a new model, Data-Race-Free-Relaxed (DRFrlx), that extends DRF0 to provide SC-centric semantics for the common use cases of relaxed atomics. Second, we evaluate the performance of relaxed atomics in CPU-GPU systems for these use cases. We find mixed results – for most cases, relaxed atomics provide only a small benefit in execution time, but for some cases, they help significantly (e.g., up to 51% for DRFrlx over DRF0).

3.29 Memory Instrumentation As A First-Class Language Feature

Michael F. Spear (Lehigh University – Bethlehem, US)

Joint work of Michael F. Spear, Xiaochen Guo, Aviral Shrivastava, Gang Tan

Across a wide range of domains, researchers require the ability to quickly instrument specific loads and stores in a program, in order to achieve novel memory behaviors. In this talk, I will introduce the first fruit of the Abstract Instrumented Memory Interface (AIMI) project: an LLVM plugin that allows researchers to perform region-based memory instrumentation of arbitrary C and C++ programs. I will first discuss motivation, then implementation, and finally discuss performance and design tradeoffs, with a focus on the AIMI submodule for Transactional Memory.

3.30 Sequential consistency considered harmful

Viktor Vafeiadis (MPI-SWS – Kaiserslautern, DE)

Sequential consistency (SC) is typically presented as the ideal semantics for shared-memory concurrency, because it is has a simple and understandable definition. This definitional simplicity, however, does not necessarily extend when SC accesses are incorporated in a weak memory model; see, e.g., [1]. In this talk, I highlighted some of the pitfalls of SC, and why it might not be such an ideal model after all. I further discussed *release/acquire* consistency (RA) [2] as an alternative model that has some tangible benefits over SC and may thus be a better model for shared-memory concurrency.

First, I showed two automated verification problems that are easier to solve for RA than for SC. Specifically, checking whether a given execution of a program annotated with reads-from edges is consistent is an NP-complete problem for SC, but is decidable in polynomial time for RA. As a result, bounded model checking for RA can be made to run significantly faster than the state of the art SC model checking tools; see, e.g., [3].

Second, regarding manual software verification, I argued that RA permits the two most useful forms of reasoning–namely *local* reasoning and *causal* reasoning–which are nicely encompassed in concurrent separation logics, such as RSL [4] and GPS [5, 6]. On the other hand, it forbids *global* reasoning [7]; i.e. reasoning in terms of which action of a thread was executed first. Global reasoning, which is sound according to SC, is a poor form of reasoning about the correctness of programs: it is complicated and error-prone; and as such is rarely used to prove the correctness of concurrent programs. By forbidding this kind of reasoning, therefore, RA may actually be a better programming model.

Finally, I briefly argued that *multicopy atomicity*–a crucial aspect of SC–prevents scalability, because it enforces a global order on every two totally independent writes by independent processors. Yet, multicopy atomicity does not seem relevant for reasoning about the correctness of concurrent programs and could thus be dropped without a perceivable difference in semantics.

References

1 Repairing sequential consistency in C/C++11. Ori Lahav, Viktor Vafeiadis, Jeehoon Kang, Chung-Kil Hur, Derek Dreyer. In PLDI 2017, pp. 618–632. ACM 2017.

2 Ori Lahav, Nick Giannarakis, Viktor Vafeiadis. Taming release-acquire consistency. In POPL 2016, pp. 649–662. ACM, 2016.

3 Michalis Kokologiannakis, Ori Lahav, Konstantinos Sagonas, and Viktor Vafeiadis. Effective stateless model checking for C/C++ concurrency. Proc. ACM Program. Lang. 2, POPL, Article 17. ACM, 2018.

4 Viktor Vafeiadis and Chinmay Narayan. Relaxed separation logic: A program logic for C11 concurrency. In OOPSLA 2013, pp. 867–884. ACM, 2013.

5 Aaron Turon, Viktor Vafeiadis, and Derek Dreyer. GPS: Navigating weak memory with ghosts, protocols, and separation. In OOPSLA 2014, pp. 691–707. ACM, 2014.

6 Jan-Oliver Kaiser, Hoang-Hai Dang, Derek Dreyer, Ori Lahav, and Viktor Vafeiadis. Strong logic for weak memory: Reasoning about release-acquire consistency in Iris. In ECOOP 2017. LIPIcs, vol. 74, pp. 17:1–17:29. Schloss Dagstuhl, 2017.

7 Ori Lahav and Viktor Vafeiadis. Owicki-Gries Reasoning for Weak Memory Models. In ICALP 2015, Part II. LNCS, vol. 9135, pp. 311–323. Springer, 2015.

3.31 Applying lock-free data structures to fabric-attached memory

Haris Volos (HP Labs – Palo Alto, US)

Next-generation rack-scale architectures will enable a fabric-attached non-volatile memory pool accessible by all compute resources. Fabric-attached memory will make interesting new programming styles possible. In this talk, I will report on our experience with one such style obtained by applying lock-free data-structure techniques to fabric-attached memory.

3.32 Isoefficiency in Practice: Configuring and Understanding the Performance of Task-based Applications

Felix Wolf (TU Darmstadt, DE)

Joint work of Sergei Shudler, Alexandru Calotoiu, Torsten Hoefler, Felix Wolf
Main reference Sergei Shudler, Alexandru Calotoiu, Torsten Hoefler, Felix Wolf: "Isoefficiency in Practice: Configuring and Understanding the Performance of Task-based Applications", in Proc. of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), Austin, TX, USA, pp. 131–143, ACM, February 2017.
URL http://dx.doi.org/10.1145/3018743.3018770

Task-based programming offers an elegant way to express units of computation and the dependencies among them, making it easier to distribute the computational load evenly across multiple cores. However, this separation of problem decomposition and parallelism requires a sufficiently large input problem to achieve satisfactory efficiency on a given number of cores. Unfortunately, finding a good match between input size and core count usually requires significant experimentation, which is expensive and sometimes even impractical. In this paper, we propose an automated empirical method for finding the isoefficiency function of a task-based program, binding efficiency, core count, and the input size in one analytical expression. This allows the latter two to be adjusted according to given (realistic) efficiency objectives. Moreover, we not only find (i) the actual isoefficiency function but also (ii) the function one would yield if the program execution was free of resource contention and (iii) an upper bound that could only be reached if the program was able to maintain its average parallelism throughout its execution. The difference between the three helps to explain low efficiency, and in particular, it helps to differentiate between resource contention and structural conflicts related to task dependencies or scheduling. The insights gained can be used to co-design programs and shared system resources.

3.33 Effect Summaries for Thread-Modular Analysis

Sebastian Wolff (TU Braunschweig, DE), Lukáš Holík, Roland Meyer, and Tomáš Vojnar

Main reference Lukás Holík, Roland Meyer, Tomás Vojnar, Sebastian Wolff: "Effect Summaries for Thread-Modular Analysis - Sound Analysis Despite an Unsound Heuristic", in Proc. of the Static Analysis - 24th International Symposium, SAS 2017, New York, NY, USA, August 30 - September 1, 2017, Proceedings, Lecture Notes in Computer Science, Vol. 10422, pp. 169–191, Springer, 2017.
URL https://doi.org/10.1007/978-3-319-66706-5_9

Thread-modular verification is the state of the art approach for verifying lock-free data structures. However, existing approaches have problems with scalability when applied in environments with manual memory management (i.e. no garbage collection).

To overcome this limitation, we identified a common programming idiom in lock-free data structures: so-called copy-and-check blocks. We show how to exploit such patterns for thread-modular analyses and report on our findings.

4 Working groups

4.1 Distributed Concurrency

Michal Friedman (Technion – Haifa, IL) and Virendra Marathe (Oracle Corp. – Burlington, US)

Summary:

The discussion, though meandering through several different topics, seemed to gravitate toward the similarities between shared memory and distributed systems. The final takeaway was that with ongoing and future technology changes, distributed systems are looking much more like shared memory systems, and vice versa. This unlocks the potential to apply wisdom learned in one setting to the other.

Details:

Maurice Herlihy's observations of similarities between the environments and challenges faced today by the world of Blockchains has strong resonance with the world of concurrency. Many of the solutions applied to the latter seem applicable to the former.

What hardware offers you and how it can be leveraged? For instance, for many-core NUMA systems, Barrelfish presents a distributed view of a shared memory system. Maybe we can think of NUMA systems as distributed systems (which is already happening).

Can lessons learned from distributed system settings be applied in shared memory setting? Today's multi-core systems are essentially distributed, with larger systems and different coherence domains (with accelerators showing up more and more in our systems), and explicit programmer intervention between these different domains.

Should we make a distinction between distributed systems and shared memory systems? There are several points in the spectrum. Even when you are presenting a simple shared memory model, discussing the implementation complexities that go into making that happen would be an interesting topic. Trade offs of programming complexity: make locality simpler at the expense of going through hoops for global access, or vice versa. In another words, better locality vs. complicated communication. Also, people have the knowledge to transfer from multicore to distributed systems. Can they do the opposite? yes!

An important difference between distributed and shared memory systems is their failure models – fail-stop models for shared memory systems, whereas the failure models of distributed systems emphasize fault tolerance, including hardware failure tolerance. However, fault isolation and security in the new containerized cloud computing world is another form of distributed computing specific issues that are applicable to shared memory systems.

Furthermore, modern hardware trends toward fast and direct remote access (RDMA) is changing the interface to distributed systems as well, where they are now looking increasingly like shared memory NUMA systems. As the cost of communication drops with these technologies, the incentive to make interfaces look similar to a single shared memory system increases.

4.2 Relaxed Atomicity

Madan Musuvathi (Microsoft Research – Redmond, US)

During the Dagstuhl we had two great discussions on weak memory models and the challenges in both providing an easy to understand interface to programmers while allowing the compiler and hardware enough freedom to implement useful optimizations. While performance of a memory model can be easily measured, a measure of its naturalness or ease of understanding to programmers is much harder to define. One proposal was to measure the complexity in the number of English words required to precisely define the memory model to a first year undergraduate student. For instance, sequential consistency can be explained with "memory as a table" abstraction, where the threads issue memory operations one at a time and these operations atomically read or update the table. Weaker memory models should seek for such simple explanations. On the other hand, an independent concern raised was for languages like C/C++ to allow programmers the ability to harness the raw power of the hardware, which falls very much in line with the philosophy of these languages. There was a lot of excitement about a proposal from Viktor Vafeiadis for a "rectified" C++ memory model that provides a logical foundation for solving the "out-of-thin-air" issues in C++ and Java. The model uses promises to model speculation and an execution is valid only when all threads eventually keep all of their promises. It also has the nice property that a restricted version of separation logic that only allows local reasoning is sound for this model. Another valuable direction of research is to identify the common patterns of relaxed variable usage in programs today to identify good patterns similar to "data-race freedom" which when enforced will provide easy-to-understand memory models.

5 Panel discussions

5.1 Panel Discussion: Concurrency vs. Parallelism

Matthew Sinclair (AMD Research – Bellevue, US)

This discussion session focused on parallelism and concurrency, both in practice and in how we teach it. Broadly, the discussion can be sub-divided into two areas of discussion: how and what we should be teaching parallelism to better equip our students for the post-university world, and what exactly is the difference between concurrency and parallelism.

Initially, the discussion revolved around teaching and using parallelism. In his talk prior to the discussion and again during the discussion, Mirko Rahn emphasized that many students are not adequately prepared to write (and debug) parallel programs in the real world. In particular, he highlighted three inter-related issues: lack of understanding about the cost of inter-processor communication, how long latency operations affect the order operations appear, and the cost of doing synchronization to ensure that operations appear in a consistent order across many processors. To Mirko, all of these problems are ultimately related to the latency of inter-processor communication, because it hurts performance, impacts what values a processor sees, and potentially requires additional, costly synchronization. Petr Kuznetsov pointed out that some of this boils down to how we teach students – and that we need to decide if we want to teach concurrency and parallelism or show students how it works. In his opinion, focusing on the underlying concepts is more important, which he does in a course he teaches that uses Java. However, Matt Sinclair pointed out that some of the universities are starting to introduce parallelism pervasively throughout the curriculum seem to focus more on the practical, performance-driven benefits. Michael Scott subsequently postulated that there is no one language or approach to teach these concepts, but that a steady, incremental approach is best – at earliest levels, we should introduce students to deterministic, independent parallelism (which have no subtleties); in later courses introduce more subtle but structured paradigms (e.g., messaging between distributed processors in distributed computing course; DRF model; event-driven programs in HCI courses); and finally in an OS course teach Peterson's algorithm (but only after making them comfortable with structured parallelism). Panagiota Fatourou presented an alternate, more theory-centric approach: she teaches an upper-level course on concurrency that starts with simpler concepts like mutual exclusion and uses small snippets of code to demonstrate how to write correct and incorrect code. Then she moves onto more complicated concepts, with liberal use of team parallel programming and examples to give the students practical experience.

The above discussion on parallel programming and concurrency led to a spirited discussion, initially raised by Victor Luchangco, about what exactly parallelism is compared to concurrency. Michael Scott emphasized that there is no one definition that separates them, and that reasonable people will disagree about them. Victor Luchangco and Michael Spear talked about concurrency in the context of robots (and later git repositories), as they believe that robotics represents one situation where students (especially students who are unfamiliar or less experienced with writing even sequential code) must think about events that are not dependent on one another (and thus could be done simultaneously). Michael Spears also focused on inexperienced programmers, and how they will likely only see a parallelized QuickSort algorithm as a performance optimization, not a way of thinking. In comparison, distributed systems (like robots) are much clearer about why concurrency is necessary/important. Subsequently, Michael Scott opined that the key here is to present algorithms where control flow shows the potential parallelism – clear thinking is the most important part, not performance (which echoes Petr's point above). This struck discussion struck me as potentially the most interesting – even a room full of experts had problems coming to a consensus about what the definition of these terms are, as well as what the best ways to introduce them to students.

Participants

- Sarita Adve
University of Illinois – Urbana-Champaign, US
- Deepthi Devaki Akkoorath
TU Kaiserslautern, DE
- Hagit Attiya
Technion – Haifa, IL
- Naama Ben-David
Carnegie Mellon University – Pittsburgh, US
- Annette Bieniusa
TU Kaiserslautern, DE
- Hans-J. Boehm
Google – Palo Alto, US
- Irina Calciu
VMware – Palo Alto, US
- Nachshon Cohen
EPFL – Lausanne, CH
- Dave Dice
Oracle Corp. – Burlington, US
- Sandhya Dwarkadas
University of Rochester, US
- Panagiota Fatourou
University of Crete – Heraklion, GR
- Pascal Felber
University of Neuchâtel, CH
- Michal Friedman
Technion – Haifa, IL
- Tim Harris
Oracle Labs – Cambridge, GB
- Danny Hendler
Ben Gurion University – Beer Sheva, IL
- Maurice Herlihy
Brown University – Providence, US
- Antony Hosking
Australian National University – Canberra, AU
- Idit Keidar
Technion – Haifa, IL
- Alex Kogan
Oracle Labs – Burlington, US
- Petr Kuznetsov
Télécom ParisTech, FR
- William M. Leiserson
MIT – Cambridge, US
- Yossi Lev
Oracle Labs – Burlington, US
- Victor Luchangco
Oracle Labs – Burlington, US
- Virendra Marathe
Oracle Corp. – Burlington, US
- Maged M. Michael
Facebook – New York, US
- J. Eliot B. Moss
University of Massachusetts – Amherst, US
- Madan Musuvathi
Microsoft Research – Redmond, US
- Vijay Nagarajan
University of Edinburgh, GB
- Erez Petrank
Technion – Haifa, IL
- Michael Philippsen
Universität Erlangen-Nürnberg, DE
- Mirko Rahn
Fraunhofer ITWM – Kaiserslautern, DE
- Michael L. Scott
University of Rochester, US
- Marc Shapiro
University Pierre & Marie Curie – Paris, FR
- Nir Shavit
MIT – Cambridge, US
- Matthew Sinclair
AMD Research – Bellevue, US
- Michael F. Spear
Lehigh University – Bethlehem, US
- Viktor Vafeiadis
MPI-SWS – Kaiserslautern, DE
- Haris Volos
HP Labs – Palo Alto, US
- Heike Wehrheim
Universität Paderborn, DE
- Reinhard Wilhelm
Universität des Saarlandes, DE
- Felix Wolf
TU Darmstadt, DE
- Sebastian Wolff
TU Braunschweig, DE

Report from Dagstuhl Seminar 17452

Algorithmic Cheminformatics

Edited by

Jakob L. Andersen[1], Christoph Flamm[2], Daniel Merkle[3], and Peter F. Stadler[4]

1 **Tokyo Institute of Technology, JP,** jlandersen@elsi.jp
2 **Universität Wien, AT,** xtof@tbi.univie.ac.at
3 **University of Southern Denmark – Odense, DK,** daniel@imada.sdu.dk
4 **Universität Leipzig, DE,** studla@bioinf.uni-leipzig.de

Abstract

Dagstuhl Seminar 17452 "Algorithmic Cheminformatics" brought together leading researchers from both chemistry and computer science. The seminar was the second in a series of the Dagstuhl seminars and had a focus on concurrency theory as chemical systems are highly concurrent by nature. Within computer science we focused on formal approaches for chemistry and concurrency theory, including process calculi and Petri nets. The participants surveyed areas of overlapping interests and identified possible fields of joint future research.

Seminar November 5–10, 2017 – http://www.dagstuhl.de/17452

1998 ACM Subject Classification F.1.2 Modes of Computation, F.4.2 Grammars and Other Rewriting Systems, F.4.3 Formal Languages, G.2.2 Graph Theory, J.2 Physical Sciences and Engineering

Keywords and phrases Modelling, Simulation, Networks, Semantics / Formal Methods

Digital Object Identifier 10.4230/DagRep.7.11.28

1 Executive Summary

Jakob L. Andersen (Tokyo Institute of Technology, JP)
Christoph Flamm (Universität Wien, AT)
Daniel Merkle (University of Southern Denmark – Odense, DK)
Peter F. Stadler (Universität Leipzig, DE)

Cheminformatics is the application of algorithms, combinatorial approaches, and formal methods from Computer Science to problems in Chemistry. While being formally a very old research field, building the theoretical foundations for Cheminformatics seen from the perspective of state-of-the-art theoretical Computer Science is not at all established research. The second edition of the seminar on "Algorithmic Cheminformatics" brought together researchers working in Chemistry, Cheminformatics, and most importantly the relevant fields in Computer Science related to it. In contrast to the first Dagstuhl meeting in 2014, we specifically focused on the analysis of the behaviour of chemical systems in terms of reaction networks. This includes both networks inferred from experimental data, as well as networks implicitly specified by for example formal grammars. We integrated experts in concurrency theory, in particular using process calculi, Petri nets, and related formal approaches. State-of-the-art results in these fields are hardly used to infer qualitative and/or quantitative properties of chemical reaction systems, which are highly concurrent systems

Algorithmic Cheminformatics, *Dagstuhl Reports*, Vol. 7, Issue 11, pp. 28–45
Editors: Jakob L. Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

by nature. Most current modeling approaches in chemistry are either very abstract and aimed at formal algebraic properties of reaction networks, or use precise modeling on a very fine grained level such as the quantum mechanical one where computational costs prevent handling of more than a few molecules. In this seminar we therefore seeked to advance discrete modeling approaches for Systems Chemistry. In addition to bringing together the experts in the respective fields from Computer Science, we also invited wet-lab chemists in order to cross-fertilize the fields and generate mutually beneficial activities.

2 Table of Contents

Executive Summary

Jakob L. Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler 28

Overview of Talks

Chemical Reaction Networks, Pathways, and Realisability
Jakob L. Andersen 32

Bio-Model-Kit – A Framework for BioModel-Engineering
Mary Ann Blätke 32

Introduction to Process Algebras
Cinzia Di Giusto 33

On the way to Synthetic Biology: Enzyme Cascade Optimisation at the University of Greifswald Robotic Platform LARA
Mark Dörr 33

Reduction of Models of Intra-Cellular Signaling Pathways
Jérôme Feret 34

Introduction to recent developments in Computational and Systems Chemistry
Christoph Flamm 34

Behavioural Model Checking of Dynamic systems, with a Focus on Reaction Networks in a Multiscale Scenario
David Gilbert 35

Parameterizing Rule-based Systems
Harold Fellermann 35

How Might Petri Nets Enhance Your Systems Biology Toolkit
Monika Heiner 36

Autocatalytic Sets and Chemical Organizations
Wim Hordijk 37

Computing Optimal Synthesis Plans
Rojin Kianian 38

Graph Transformation, Rule Composition, and Stereochemistry
Daniel Merkle 38

The Symbolic Method & Random Generation
Markus E. Nebel 39

Behavioural Equivalences in a Nutshell
Marco Peressotti 39

Exploring the Reaxys network of chemical reactions: Computational history of chemical reactions
Guillermo Restrepo 40

Open Science and Chiminey a Collaborative Platforms for Biochemistry
Heinrich W. Schmidt 40

Brief Comments on Origin of Life Iinterests in Computational Cheminformatics
D. Eric Smith 41

CRNs: A Contact Point Between Cheminformatics and Recent Topics of Interest in the NEQ Statistical Physics Community
D. Eric Smith 42

Bisimulations for Differential Equations
Mirco Tribastone 42

Open Instrumentation
Klaus-Peter Zauner 43

Participants 45

3 Overview of Talks

3.1 Chemical Reaction Networks, Pathways, and Realisability

Jakob L. Andersen (Tokyo Institute of Technology, JP)

Joint work of Jakob L. Andersen, Christoph Flamm, Daniel Merkle, Peter F. Stadler

We present a modelling framework for pathways in chemical reaction neworks. The model is based on a generalisation of network flows to directed multihypergraph, and a restriction to integer hyperflows allows for analysis of pathwys on a mechanistic level. We addiionally introduce a set of necessary constraints for a pathway to be considered either catalytic or autocatalytic. Using ideas from Petri nets we then analyse the realisability of pathways from a concurrency perspective.

3.2 Bio-Model-Kit – A Framework for BioModel-Engineering

Mary Ann Blätke (IPK Gatersleben, DE)

Representing the complexity of biomolecular components in a consistent and straightforward way is one of the challenges in biomodel engineering. We developed a modular modelling framework for biomodel engineering, called BioModelKit, addressing this problem. In the BioModelKit framework, modules are designed for the purpose of model composition. A module is a molecule-centred Petri net model representing the functionality and interactions of a biomolecule. Defined interfaces shared among the modules allow the modular composition of models without manual adjustments. We distinguish between mechanistic and causal module types, to capture the different types of networks and omic fields. Each module includes an annotation file holding descriptive information and cross-links to related references. The algorithmic model mutation of composed models allows mimicking the effect of gene knock-outs and structural mutations of a biomolecule. Modularly composed models can also be transformed into spatial models to represent the movement of biomolecules, the cell geometry, compartments, and the distribution of molecules. Modules can be constructed using direct and reverse engineering approaches. Existing models, e.g SBML or Boolean models, as well as OMIC data can be another source to obtain modules. The modular model composition and its extensions are implemented in a web-based tool that is supported by a MySQL database holding the Petri net graphs and annotations of submitted modules. The BioModelKit framework mentioned extensions overs a versatile and unifying tool for biomodel engineering.

3.3 Introduction to Process Algebras

Cinzia Di Giusto (Laboratoire I3S – Sophia Antipolis, FR)

We introduce the main concepts behind process algebras.

Process Algebras are mathematical formalism that allow the specification, the analysis and the verification of properties of concurrent and distributed systems. Processes are agents that work in parallel, exchange informations and take decisions depending on the acquired informations. The semantics of such processes is given in terms of structural operational semantics (SOS) rules and labelled transition systems are built and composed by using the different operators of the given process algebra.

In order to compare processes, behavioural equivalences are used to abstract from unwanted details and identify those processes that behaves "similarly". We introduce trace and bisimilarity equivalences.

Finally we briefly present behavioural types which specifically describe protocols in distributed systems. The type system ensures well-typed processes to enjoy properties such as communication safety, protocol fidelity, and progress.

3.4 On the way to Synthetic Biology: Enzyme Cascade Optimisation at the University of Greifswald Robotic Platform LARA

Mark Dörr (Universität Greifswald, DE)

Main reference M. Dörr, M. P. C. Fibinger, D. Last, S. Schmidt, J. Santos-Aberturas, D. Böttcher, A. Hummel, C. Vickers, M. Voss, U. T. Bornscheuer: "Fully Automatized High-Throughput Enzyme Library Screening Using a Robotic Platform", Biotechnol. Bioeng. , 113 (7), 1421–1432, 2016.
URL http://dx.doi.org/10.1002/bit.25925

Automated evolutionary experiments with (ultra) high-throughput in synthetic biology and protein design are state-of-the-art approaches in the bio-chemical laboratory. They generate new challenges on the technical realisation, e.g., detection of very tiny samples and their changes over time, and on the computational side, e.g., data evaluation of the large amounts of data points, machine learning and "machine based understanding" of the underlying experiments and finally closed feedback cycles of automated evolution. We demonstrated our different technical approaches, like our robotic protein screening platform LARA (lara.uni-greifswald.de) [2] for macroscopic samples. And microfluidic set-ups utilizing millions of nano- and picoliter sized droplets with individual chemical composition (Prof. Bornscheuer lab, Biochemistry, Uni-Greifswald). Droplet formation and double droplet formation (water-in-oil-in water emulsions) has been explained. The experimental challenges of "Synthetic Biology" and the introduction of new genes that are completely foreign to a species have been demonstrated by the example of the oligo-epsilon-caprolactone production in E.coli [1], [2]). Tuning different protein expression levels can be addressed by splitXFP systems (XFP: colour X Fluorescent Protein) [3]. This is an important tool to adjust a good substance flux through a synthetic metabolic pathway. Furthermore protein long-term stability is an issue that was addressed by automated temperature selection experiments.

References

1 S. Schmidt, C. Scherkus, J. Muschiol, U. Menyes, T. Winkler, W. Hummel, H. Gröger, A. Liese, H.-G. Herz, U. T. Bornscheuer. *An Enzyme Cascade Synthesis of ϵ-Caprolactone and Its Oligomers*. Angew. Chem. Int. Ed. 2015, 54 (9), 2784–2787.

2 M. Dörr, M. P. C. Fibinger, D. Last, S. Schmidt, J. Santos-Aberturas, D. Böttcher, A. Hummel, C. Vickers, M. Voss, U. T. Bornscheuer. *Fully Automatized High-Throughput Enzyme Library Screening Using a Robotic Platform*. Biotechnol. Bioeng. 2016, 113 (7), 1421–1432.

3 J. Santos-Aberturas, M. Dörr, G. S. Waldo, U. T. Bornscheuer. *In-Depth High-Throughput Screening of Protein Engineering Libraries by Split-GFP Direct Crude Cell Extract Data Normalization*. Chemistry and Biology 2015, 22 (10), 1406–1414.

3.5 Reduction of Models of Intra-Cellular Signaling Pathways

Jérôme Feret (ENS – Paris, FR)

We use the flow of information between the states of sites in bio-molecular compounds in rule-based models of signaling pathways, so as to identify which correlations have a real impact on the dynamics of the models. The others can be safely abstracted away which leads to a factorisation of the underlying differential systems. As a result, we get a fully automatic procedure to generate a reduced ODE systems, without having to consider the ODE equations of the initial network.

3.6 Introduction to recent developments in Computational and Systems Chemistry

Christoph Flamm (Universität Wien, AT)

Over the last decade the focus of experimental approaches in systems chemistry and the field of origin of life research has shifted from the study of single molecules with close structural proximity to biological building blocks to the high-throughput analysis of complex mixtures of molecules which often possess little to no direct biological relevance. This development, i.e. the manipulation of diverse molecular mixtures, is not mirrored by theoretical advances in computational chemistry. Novel computational approaches capable of handling concurrency and causality in complex reactive mixtures are required to deconvolute the intricate chain of events that resulted in the experimental observation. Computational approaches could be even further exploited as unbiased discovery tools for novel chemical mechanisms instead of only arbitrating between competing experimental hypothesis. The current excitement in deep machine learning approaches does unfortunately as-well not carry over to chemistry since most of the chemical knowledge is sealed away in proprietary databases and not accessible on the large scale in the public domain. The problematic sketched above will be discussed in depth in my presentation illustrating certain aspects with experimental and computational examples from recent literature.

3.7 Behavioural Model Checking of Dynamic systems, with a Focus on Reaction Networks in a Multiscale Scenario

David Gilbert (Brunel University – Uxbridge, GB)

I describe how model checking is used over the dynamic behaviour of models of biochemical reaction systems, which can be modelled in Petri nets, but also in other formalisms, and uses descriptions in temporal logic. The approach can also be applied to time-series laboratory data. Our approach can be applied to model checking the behaviours of very large models, for example of whole genome metabolism of E.coli, using libraries of properties of interest of both metabolite concentrations as well as of reaction activity over time.

I also show how this behaviour-based model checking can be applied to multi-level [and multi-scale] models which have been constructed using Coloured Petri nets, with an example of planar cell polarity in Drosophila fly wing.

I also show how temporal logic descriptions can be automatically derived from sets of examples.

Finally I put this methodology in the context of design for synthetic biology, and within a multiscale scenario, and speculate how this could be relevant to cheminformatics.

References

1 D. Gilbert, M. Heiner, Y. Jayaweera, C. Rohr. *Towards dynamic genome-scale models.* Briefings in Bioinformatics, 2017. https://doi.org/10.1093/bib/bbx096

2 O. Parvu, D. Gilbert. *A Novel Method to Verify Multilevel Computational Models of Biological Systems Using Multiscale Spatio-Temporal Meta Model Checking.* PloS one 11 2016:(5). https://doi.org/10.1371/journal.pone.0154847

3 Q. Gao, D. Gilbert, M. Heiner, F. Liu, D. Maccagnola and D. Tree. *Multiscale Modelling and Analysis of Planar Cell Polarity in the Drosophila Wing.* IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2012:10:2. http://doi.org/10.1109/TCBB.2012.101

4 D. Maccagnola, E. Messina, Q. Gao and D. Gilbert. *A Machine Learning Approach for Generating Temporal Logic Classifications of Complex Model Behaviours.* In Proceedings of WSC 2012 (Winter Simulation Conference December 9-12, 2012, Berlin).

3.8 Parameterizing Rule-based Systems

Harold Fellermann

Joint work of Harold Fellermann, Annunziata Lopiccolo, Benjamin Shirt-Ediss

Rule-based modelling is a powerful technique for the simulation and analysis of complex chemical and molecular biological systems. Its strength results from its ability to capture a polynomial number of reactions among a combinatorial set of reactants in few rules that model general interactions. Parameterizing rule-based systems with experimental data is an important yet relatively little explored field.

An example of a chemical system that lends itself to rule-based modelling is our current work toward a molecular memory devices (stack data structure) based on DNA hybridization and strand displacement reactions [1]. Parallel to our experiments, we develop rule-based models that capture device operation with three to four rules. In order to use such models for hypothesis testing, thorough parameterization of reaction rates is paramount.

After briefly summarizing approaches toward parameter optimization, I emphasize the difficulties of systematically performing parameter fits of rule-based models against experimental data: namely (a) proper treatment of stochasticity, (b) potential infinity of the system's state space, (c) generalization of rate constants to context-dependent rate functions, (d) difficulty in quantitatively comparing the model state space against experimental electrophoresis data.

These difficulties are the motivation for a semi-automated qualitative fitting procedure of the models: with an interactive interface, the modeller can explore the impact that parameter choices have on a simulated experiment by visually comparing the experimental images to virtual electrophoresis images generated from model predictions.

We demonstrate that this approach leads to literature consistent parameters and a good qualitative fit. Importantly, the procedure can be used to support hypothesis testing if alternative models are probed against the same set of data.

References

1 H. Fellermann, A. Lopiccolo, J. Kozyra, N. Krasnogor. *In vitro implementation of a stack data structure based on DNA strand displacement* in Martyn Amos *Unconventional Computation and Natural Computation* pp. 87–89. Springer, Heidelberg, 2016.

3.9 How Might Petri Nets Enhance Your Systems Biology Toolkit

Monika Heiner (BTU Cottbus, DE)

In my talk I will give a Petri net perspective on the exciting research field of "Biomodel Engineering".

Petri nets enjoy an intuitive graphical representation and formal semantics alike, thus they are a popular notation for biochemical reaction networks, such as gene regulatory, signal transduction or metabolic networks.

In our scenario, Petri nets serve as umbrella formalism combining different modelling paradigms, where each perspective contributes to a better understanding of the biochemical system under study. In this spirit of BioModel Engineering, we developed over the last two decades our unifying Petri net framework comprising the traditional time-free Petri nets (PN) as well as quantitative, i.e. time-dependent Petri nets such as stochastic Petri nets (SPN – opening the door to the Markovian world), continuous Petri nets (CPN – opening the door to the world of ordinary differential equations), and hybrid Petri nets (HPN – combining the previous two formalisms), as well as their their coloured counterparts. The crucial idea is to have a family of related models, sharing structure, but differing in their kinetic details.

Coloured Petri nets permit, among others, the convenient and flexible encoding of spatial attributes, and thus the modelling of processes evolving in time and space, which are usually treated as stochastic or deterministic partial differential equations (PDE). In our approach, the discretisation of space happens on the modelling level, while traditionally the discretisation is left for the PDE integration methods.

Our framework is supported by a related Petri net toolkit comprising Snoopy, Charlie and Marcie. It has been applied to a couple of case studies.

For more details, please see http://www-dssz.informatik.tu-cottbus.de/BME/BME-tutorial

References

1 M. Heiner and D. Gilbert. *How Might Petri Nets Enhance Your Systems Biology Toolkit*. In Proc. PETRI NETS 2011, Springer, LNCS 6709, pp. 17–37, 2011. https://doi.org/10.1007/978-3-642-21834-7_2

2 D. Gilbert, M. Heiner, Y. Jayaweera and C. Rohr. *Towards dynamic genome scale models*. Briefings in Bioinformatics, 2017. https://doi.org/10.1093/bib/bbx096

3.10 Autocatalytic Sets and Chemical Organizations

Wim Hordijk (Konrad Lorenz Institut – Klosterneuburg, AT)

Joint work of Hordijk, Wim; Steel, Mike; Dittrich, Peter
Main reference W. Hordijk, M. Steel, P. Dittrich, "Autocatalytic sets and chemical organizations: Modeling self-sustaining reaction networks at the origin of life", New Journal of Physics 20:015011, 2018.
URL https://doi.org/10.1088/1367-2630/aa9fcd

Autocatalytic sets are self-sustaining reaction networks in which all molecules mutually catalyze each other's formation from a basic food source. They are believed to have played an important role in the origin of life. Much mathematical and computational work has been done on autocatalytic sets in the form of RAF theory [1], but they have also been constructed experimentally, either with RNA molecules [2] or with peptides [3].

One result coming out of RAF theory is that often such self-sustaining sets consist of a hierarchical structure of smaller autocatalytic subsets [4]. From a dynamical point of view, the so-called "closed autocatalytic sets" are of primary interest, as they represent the dynamically stable states that are observed in stochastic dynamic simulations of autocatalytic sets.

A related formalism for modeling self-maintaining chemical reaction networks is that of chemical organization theory (COT) [5]. Recently, we have shown several formal relationships between RAF theory and COT, in particular between closed autocatalytic subsets and chemical organizations [6]. Moreover, this close relationship leads to a precise method to enumerate all closed subsets within an autocatalytic set by calculating all its chemical organizations.

In my talk I presented the main ideas behind closed autocatalytic sets, how they play an important role in the stochastic dynamics of autocatalytic sets, and how they can be enumerated using chemical organization theory.

References

1 Hordijk and Steel. BioSystems 152:1–10, 2017.

2 Vaidya et al. Nature 491:72–77, 2012.

3 Ashkenasy et al. PNAS 101:10872–10877, 2004.

4 Hordijk et al. Acta Biotheoretica 60:379–392, 2012.

5 Dittrich and Speroni di Fenizio. Bulletin of Mathematical Biology 69:1199–1231, 2007.

6 Hordijk et al. New Journal of Physics 20:015011, 2018.

3.11 Computing Optimal Synthesis Plans

Rojin Kianian (University of Southern Denmark – Odense, DK)

Joint work of Rolf Fagerberg, Christoph Flamm, Rojin Kianian, Daniel Merkle, Peter F. Stadler

In synthesis planning, the goal is to synthesize a target molecule from available starting materials, possibly optimizing costs such as price or environmental impact of the process.

We demonstrate that synthesis planning can be phrased as a combinatorial optimization problem on hypergraphs by modeling individual synthesis plans as directed hyperpaths embedded in a hypergraph of reactions (HoR) representing the chemistry of interest. As a consequence, a polynomial time algorithm to find the K shortest hyperpaths can be used to compute the K best synthesis plans for a given target molecule. Having K good plans to choose from has many benefits: it makes the synthesis planning process much more robust when in later stages adding further chemical detail, it allows one to combine several notions of cost, and it provides a way to deal with imprecise yield estimates.

Our modeling is not restricted to bond set based approaches, which are otherwise the most widespread—any set of known reactions and starting materials can be used to define a HoR, as long as the reactions are modeled with only one product.This generalization enables computation of synthesis plans using the collected knowledge of chemists, provided by Reaxys. In this way, the expert knowledge of reactions that are well tested in the lab may be combined into plans in ways that have not been thought of before and that are not biased by tradition.

3.12 Graph Transformation, Rule Composition, and Stereochemistry

Daniel Merkle (University of Southern Denmark – Odense, DK)

Joint work of Jakob L. Andersen, Christoph Flamm, Daniel Merkle, Peter F. Stadler
Main reference Jakob L. Andersen, Christoph Flamm, Daniel Merkle, Peter F. Stadler: "Chemical Transformation Motifs – Modelling Pathways as Integer Hyperflows", IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. PP(99), IEEE, 2017.
URL http://dx.doi.org/10.1109/TCBB.2017.2781724

Graph transformation form a natural model for chemical reaction systems and provide a sufficient level of detail to track individual atoms. Among alternative graph transformation formalisms the Double Pushout approach, which is firmly grounded in category theory, is particularly well-suited as a model of chemistry. The formal foundations of defining composition of transformation rules using ideas from concurrency theory are presented. In addition of a generic construction several special cases that each have an intuitive chemical interpretation will be considered. The usefulness of these specialised operations by automatically calculating coarse-grained transformation rules for complete chemical pathways, that preserve the traces of atoms through the pathways, will be illustrated. This type of computation has direct practical relevance for the analysis and design of isotope labelling experiments. Examples will be illustrated with MØD, a software package developed for graph-based cheminformatics.

References

1 J. L. Andersen, C. Flamm, D. Merkle, P. F. Stadler. *Chemical Transformation Motifs – Modelling Pathways as Integer Hyperflows*. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2018.

2 J. L. Andersen, C. Flamm, D. Merkle, P. F. Stadler. *Rule Composition in Graph Transformation Models of Chemical Reactions*. MATCH Communications in Mathematical and in Computer Chemistry, 2018, to be published.

3.13 The Symbolic Method & Random Generation

Markus E. Nebel (Universität Bielefeld, DE)

Joint work of Daniel Merkle, Markus Nebel

In this talk we show how methods based on generating functions and complex analysis can be used to count and randomly generate various kinds of molecules taking stereoisomers into account. Based on so-called symbolic equations for the recursive structure of the molecules, precise asymptotic formulae for the number of different conformations of given size (number of carbon atoms) can be derived. Furthermore, the underlying recursive description together with the resulting generating functions can be used to efficiently generate random molecule of given size according to uniform and non-uniform probability distributions. We discuss fixed size as well as approximate size (Boltzmann) sampling approaches.

3.14 Behavioural Equivalences in a Nutshell

Marco Peressotti (University of Southern Denmark – Odense, DK)

Two of the fundamental questions driving research in Concurrency Theory are “what is a process?” and “when is the behaviour of two processes equivalent?”. Although a definitive answer is still elusive, remarkable progress has been made in the last three and a half decades. A cornerstone of the theory and perhaps one of the most illustrative example, of behavioural equivalence is Milner and Park’s bisimulation. This notion was initially proposed for the study of non-deterministic processes and, as this theory matured, more detailed models of concurrent systems have been developed, some including quantitative data such as cost, probabilities, and time. Today, these techniques are widely used in computer science, but also in other fields, including artificial intelligence, cognitive science, mathematics, modal logics, philosophy, and physics; mainly to explain phenomena involving some kind of circularity or infinite object.

The talk provides an introduction to the theory of behavioural equivalences with special attention to bisimulation and coinduction.

3.15 Exploring the Reaxys network of chemical reactions: Computational history of chemical reactions

Guillermo Restrepo (Universität Leipzig, DE)

Joint work of Guillermo Restrepo, Eugenio J. Llanos, Peter F. Stadler, Wilmer Leal

The chemical community devotes most of its efforts to synthetic chemistry, therefore knowledge about reactants, catalysts, solvents and several other related aspects of chemical reactions is of important relevance. Part of this knowledge is its history that involves determining the aspects that have shaped chemical reactions to their current state; which are tasks for the history of chemistry. However, analysing the chemical reactions that have been reported in the scientific literature is not any more a subject of the conventional history of chemistry, for the number of substances and reactions grows exponentially. Here we show that a computational approach to the history of chemical reactions sheds light on the patterns behind the development and use of substances and reaction conditions. We explored the more than 45 million reactions gathered in Reaxys database and came across with historical patterns for substances, types of substances, catalysts, solvents, temperatures and pressures of those reactions. It is found that chemists have traditionally used few reactants to produce many different substances. In such synthesis more combinations of about four chemical elements are explored. Despite the exponential growth of substances and reactions, little variation of catalysts, solvents, and reactants is observed throughout time. Regarding reaction conditions, the vast majority of reactions fall into a narrow domain of temperature and pressure, namely normal conditions. We also found and quantified the effect of world wars (WWs) upon chemical novelty during war periods. WW1 took production of new substances and reactions back around 30 years and WW2 around 15. We anticipate this study and especially its methodological approach to be the starting point for the computational history of chemical reactivity, where social and economical contexts are integrated.

3.16 Open Science and Chiminey a Collaborative Platforms for Biochemistry

Heinrich W. Schmidt (RMIT University – Melbourne, AU)

Scientific experiments and engineering modeling increasingly include data and software. A so-called fourth paradigm of scientific discovery has emerged based on these. It enables researchers to accelerate the three other paradigms, empirical, theoretical and computing by learning models from data and by software-defined workflows, sharing of data and collaborating globally at a scale not possible just a few decades ago. We developed Chiminey with quantum physicists and molecular biologists, to ease scaling up stochastic physics and chemistry algorithms from small data sets and laptop use to run straight on massive numbers of nodes in high-performance compute centres or the Australian Open Science Cloud.

This talk draws from two presentations, the first providing some broader insights into the Open Science movement with its FAIR, i.e., findable, accessible, interoperable and reusable resources (data and software) and community support; the second drilling down into specifics of human-centric parallel stochastic modelling with Chiminey.

Some aspects of Chiminey can be found here http://doi.org/10.1016/j.bdr.2017.01.004 and here http://github.com/chiminey/chiminey.

3.17 Brief Comments on Origin of Life Iinterests in Computational Cheminformatics

D. Eric Smith (Santa Fe Institute, US)

The task is to go from the beginning of order in near-surface geochemistry, to the pruned and hierarchical order of biochemistry.

In this short presentation I review the properties of biochemistry that suggest that certain chemistry is easier than other, at either the mechanistic or the network level, and that this has entrained the large-scale structure of metabolism and evolution.

A second topic is the role of metals in biochemistry and in our inference about the earliest relevant geochemistry, and the progress and goals for incorporating them into graph-grammar systems.

In order to try to find a relevant sense of causation for the chemistry and topology of core metabolism, drawing on the above evidence, however, we would need to know how readily the same properties that suggest an easiest or best solution (reaction redundancy, network completion, autocatalysis, etc.), which exist in actual biochemistry, would be found at arbitrary other points in the space of possible planetary chemical precursor systems. This is needed to disambiguate the confound between a pattern that is laid down in geochemistry as s kind of chemical path of least resistance, versus a frozen accident of evolutionary dynamics in a later era.

The combination of being able to model metals as gateways, at the elementary reaction level, and to simulate very large networks and to efficiently query for network properties that putatively aid evolution or stability, is the advantage that graph grammars seem most likely to promise.

A particular goal in the use of metals is to systematically search for catalysis-related phenomena – once they are demonstrated to exist in some context – for other contexts in which the same phenomena may be found. We might want to follow what has successfully been done in small-molecular organocatalysis, to abstract mechanism so that we can see when the same fundamental mechanism can be achieved at all levels from free ions, to soluble metal-ligand complexes, to mineral unit cells, and ultimately to metal-center cofactors or enzymes that have been tuned by evolution. When common mechanisms exist, this provides a path for continuity in which catalysts can be replaced and evolved, while leaving the overall chemical synthetic network in place from prebiotic contexts all the way to extant biochemistry.

3.18 CRNs: A Contact Point Between Cheminformatics and Recent Topics of Interest in the NEQ Statistical Physics Community

D. Eric Smith (Santa Fe Institute, US)

This presentation had three parts. The first was a brief review of the problem framework known as Chemical Reaction Network theory, and a few of its major results (mostly in convex analysis) from the deterministic methods begun in the 1970s. The second part was an introduction to some areas of interest in modern non-equilibrium statistical mechanics, particularly the question of how much about dynamics can be understood by reference to the local-equilibrium entropy, and what entropy concept takes its place when even local conditions are distant from equilibrium. The third part was a small worked example, showing how the methods of Integer Linear Programming that are common in the topological analysis of CRNs and Petri Nets, can be used to characterize the transport of probability in what are known as “complex-balanced” steady states, the largest class of non-equilibrium systems for which the major elements of Equilibrium Thermodynamic structure are retained.

The worked example illustrated three results in the relation between topology and the probabilistic dynamics of CRNs:

The first is that complex-balanced flows preserve a combinatorial density of states which is a unifying feature of conventional Gibbsian thermal equilibria. The second is that complex-balanced flows preserve a factorization between the Large-Deviations behavior of the environment and that of the system, which is expressed in the property that all steady states are exponential tilts of one another, or of an invariant combinatorial density of states. The third result is that under the same conditions, probability flow at steady states mirrors the flow of the Balanced Integer Hyperflows. This redundant situation, in which all transport quantities are governed by the same small collection of integer coefficients, may be understood as another expression of a well-known fact (proved by Anderson, Craciun, and Kurtz) that the steady states can be written as product-Poisson densities (or slices through them), which means that they are “minimal-information” distributions, with all higher-order moment behaviors dictated in an invariant manner by the values of the mean particle numbers.

3.19 Bisimulations for Differential Equations

Mirco Tribastone (IMT – Lucca, IT)

Large-scale dynamical models hinder our capability of effectively analyzing them and interpreting their behavior. We present an algorithm for the simplification of polynomial ordinary differential equations by aggregating their variables. The reduction can preserve observables of interest and yields a physically intelligible reduced model, since each aggregate corresponds to the exact sum of original variables.

Tool implementation: http://sysma.imtlucca.it/tools/erode/

3.20 Open Instrumentation

Klaus-Peter Zauner (University of Southampton, GB)

Joint work of Martin Fischlechner, Jonathan West, Klaus-Peter Zauner
URL https://dropletkitchen.github.io/

A brief overview of recent progress in artificial chemical networks based on lipid coated Belousov Zhabotinsky (BZ) droplets in oil showed the use of laser-cut acrylic templates to position hundreds of BZ droplets in desired network configurations [1]. From this simple application of digital fabrication for custom laboratory tools the presentation focused on the potential of the confluence of low-cost embedded systems and modern fabrication techniques such as 3D printing and laser-cutting for the fast evolution of scientific instrumentation.

Novel directions in experimentation often require highly specialized custom made equipment with a concomitant high cost. As a consequence both access to the new technique and the number of allowed design iterations for the instrument are typically severely limited. This situation is about to change and the impact for experimental work is likely to resemble the impact the arrival of the personal computer in the laboratory had with moving data analysis from punch cards on a mainframe to a spread sheet on the researchers desk. A large number of software tools subsequently originated because scientists could iteratively develop themselves the tool that answered their idiosyncratic requirements. By sharing these tools as open source software they were refined by the community to a level that would be unaffordable for any particular research programme and at the same time the tools are widely available.

At present we are at the verge of a similar development with regard to laboratory instruments. Very capable embedded systems that can be programmed with ease[1] complement the flexibility of digital fabrication. The convenience of interfacing with actuators and sensors makes it possible to forsake precision in the components while maintaining overall systems performance through closed-loop control. This in turn makes it possible to use affordable 3D printers and place them directly in the laboratory where the products are used, thus facilitating a cycle where deficiencies in a tool or instrument can result in a modification of the instrument within days, if not hours. By making the files required for the fabrication of the mechanical and electronic hardware, as well as the software available this feedback cycle can transcend the laboratory; and arguably will lead to the refinement of idiosyncratic laboratory tools as was the case with scientific open source software.

Some examples of open instruments–in the sense that the software and hardware plans required to replicate the instruments are available and can be modified and redistributed–for use with microfluidic chips are available[2]; for a collection of links to various instruments see [3].

Aside from the faster development of the instruments, the low cost of open instrumentation offers additional advantages. Equipment is documented in detail which supports both the reproduction of experiments and the evaluation of results. Users can have their personal instrument and whole class sets can be produced instead of sharing a single piece of equipment. Universities and schools with limited funding can access state-of-the art methods. Experiments can be replicated in parallel at a scale that might otherwise not be fundable. In our own research we hope to exploit the adaptability of open instrumentation to develop autonomous experimentation with microfludics and flow chemistry experiments.

[1] joint work with Kai Ming Chang and Maurits R.R. de Planque

References

1 D. P. George, 2014 *MicroPython*. https://micropython.org/

2 M. Fischlechner, J. West, K.-P. Zauner, 2016 *DropletKitchen*. https://dropletkitchen.github.io/

3 Pearce, Joshua, 2013 *Building research equipment with free, open-source hardware*. http://www.appropedia.org/Building_research_equipment_with_free,_open-source_hardware

Participants

- Jakob Lykke Andersen
Tokyo Institute of Technology, JP
- Sarah Berkemer
MPI für Mathematik in den Naturwissenschaften, DE
- Mary Ann Blätke
IPK Gatersleben, DE
- Erick Chastain
Rutgers University – Piscataway, US
- Cinzia Di Giusto
Laboratoire I3S – Sophia Antipolis, FR
- Peter Dittrich
Universität Jena, DE
- Mark Dörr
Universität Greifswald, DE
- Rolf Fagerberg
University of Southern Denmark – Odense, DK
- Harold Fellermann
University of Newcastle, GB
- Jérôme Feret
ENS – Paris, FR
- Christoph Flamm
Universität Wien, AT
- Walter Fontana
Harvard Medical School – Boston, US
- David Gilbert
Brunel University – Uxbridge, GB
- Monika Heiner
BTU Cottbus, DE
- Marc Hellmuth
Universität Greifswald, DE
- Wim Hordijk
Konrad Lorenz Institut – Klosterneuburg, AT
- Rojin Kianian
University of Southern Denmark – Odense, DK
- Daniel Merkle
University of Southern Denmark – Odense, DK
- Markus E. Nebel
Universität Bielefeld, DE
- Nikolai Nøjgaard
Universität Greifswald, DE
- Marco Peressotti
University of Southern Denmark – Odense, DK
- Guillermo Restrepo
Universität Leipzig, DE
- Heinrich W. Schmidt
RMIT University – Melbourne, AU
- D. Eric Smith
Santa Fe Institute, US
- Peter F. Stadler
Universität Leipzig, DE
- Mirco Tribastone
IMT – Lucca, IT
- Klaus-Peter Zauner
University of Southampton, GB

Report from Dagstuhl Seminar 17461

Connecting Visualization and Data Management Research

Edited by

Remco Chang[1], Jean-Daniel Fekete[2], Juliana Freire[3], and Carlos E. Scheidegger[4]

1 **Tufts University – Medford, US,** remco@cs.tufts.edu
2 **INRIA Saclay – Orsay, FR,** jean-daniel.fekete@inria.fr
3 **New York University, US,** juliana.freire@nyu.edu
4 **University of Arizona – Tucson, US,** cscheid@email.arizona.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 17461 "Connecting Visualization and Data Management Research".

Seminar November 12–17, 2017 – http://www.dagstuhl.de/17461
1998 ACM Subject Classification D.1.7 Visual Programming, H.2 Database Management, H.5.2 User Interfaces, H.1.2 User/Machine Systems
Keywords and phrases Interactive data analysis, Data visualization, Visual analytics, Data management system, Systems for data science
Digital Object Identifier 10.4230/DagRep.7.11.46

1 Executive Summary

Remco Chang
Jean-Daniel Fekete
Juliana Freire
Carlos E. Scheidegger

What prevents analysts from acquiring wisdom from data sources? To use data, to better understand the world and act upon it, we need to understand both the computational and the human-centric aspects of data-intensive work. In this Dagstuhl Seminar, we sought to establish the foundations for the next generation of data management and visualization systems by bringing together these two largely independent communities. While exploratory data analysis (EDA) has been a pillar of data science for decades, maintaining interactivity during EDA has become difficult, as the data size and complexity continue to grow. Modern statistical systems often assume that all data need to fit into memory in order to support interactivity. However, when faced with a large amount of data, few techniques can support EDA fluidly. During this process, interactivity is critical: if each operation takes hours or even minutes to finish, analysts lose track of their thought process. Bad analyses cause bad interpretations, bad actions and bad policies.

As data scale and complexity increases, the novel solutions that will ultimately enable interactive, large-scale EDA will have to come from truly interdisciplinary and international work. Today, database systems can store and query massive amounts of data, including methods for distributed, streaming and approximate computation. Data mining techniques provide ways to discover unexpected patterns and to automate and scale well-defined analysis procedures. Recent systems research has looked at how to develop novel database systems architectures to support the iterative, optimization-oriented workloads of data-intensive algorithms. Of course, both the inputs and outputs of these systems are ultimately driven by people, in support of analysis tasks. The life-cycle of data involves an iterative, interactive process of determining which questions to ask, the data to analyze, appropriate features and models, and interpreting results. In order to achieve better analysis outcomes, data processing systems require improved interfaces that account for the strengths and limitations of human perception and cognition. Meanwhile, to keep up with the rising tide of data, interactive visualization tools need to integrate more techniques from databases and machine learning.

This Dagstuhl seminar brought together researchers from the two communities (visualization and databases) to establish a research agenda towards the development of next generation data management and interactive visualization systems. In a short amount of time, the two communities learned from each other, identified the strengths and weaknesses of the latest techniques from both fields, and together developed a "state of the art" report on the open challenges that require the collaboration of the two communities. This report documents the outcome of this collaborative effort by all the participants.

2 Table of Contents

Executive Summary

Remco Chang, Jean-Daniel Fekete, Juliana Freire, and Carlos E. Scheidegger . . . 46

Overview of Talks

A Holistic Approach to Human-centric Data Exploration and Analysis
Daniel Keim and Volker Markl 49

Towards making the database in*vis*ible
Tiziana Catarci, Surajit Chaudhuri, Danyel Fisher, Heike Hofmann, Tim Kraska, Zhicheng Liu, Dominik Moritz, Gerik Scheuermann, Richard Wesley, Eugene Wu, Yifan Wu 49

Understandability in Data Discoveries: an Integrated DB and Vis Perspective
Sihem Amer-Yahia, Harish Doraiswamy, Steven Drucker, Juliana Freire, Daniel Keim, Heike Leitte, Alexandra Meliou, Torsten Möller, Thibault Sellam, Juan Soto 52

Reducing Friction in Applying Data Management for Visualization
Michael Gleicher, Behrooz Omidvar-Tehrani, Hannes Mühleisen, Themis Palpanas 54

An Evaluation Methodology for Vis + DB
Leilani Battle, Carsten Binnig, Tiziana Catarci, Stephan Diehl, Jean-Daniel Fekete, Michael Sedlmair, Wesley Willett 55

Community Challenges 56

Conclusion 57

Participants 58

3 Overview of Talks

3.1 A Holistic Approach to Human-centric Data Exploration and Analysis

Daniel Keim (Universität Konstanz) and Volker Markl (TU Berlin)

The grand challenge of enabling human data interaction on today's fast, diverse, uncertain datasets is to support the entire iterative data exploration and analysis process, from data preparation to modeling and visualization in a holistic, end-to-end fashion. Making the process human-centric requires novel algorithms for detecting human interests and intents, deriving system-understandable tasks from them, and executing them under the consideration of interaction and cognitive latencies. The system may also proactively determine interesting properties of the data suggesting novel directions of exploration to the human, using the processing time available before starting the interactive analysis and during cognitive breaks. The human needs to be supported and guided in all steps of the process including the data preprocessing, data analytics and task specification process, requiring various algorithms with different constraints on precision and latency.

What is needed is a language that enables the specification of the entire process. Such a specification would allow for a joint understanding of the interdisciplinary problems across the process and facilitate building systems that address the needs of both authors and consumers of interactive visualizations with respect to productivity and performance. This requires us to design a specification and a corresponding grammar, language, and algebraic behavior; a "Grand Unified Algebra". This specification must be expressive enough to define desires and constraints on quality and performance, and to enable developers and users to iterate rapidly on design and analysis. At the same time it has to be compilable and processable by a data processing runtime engine, leveraging both well known concepts from database technology such as automatic query optimization and parallelization, transactions, and consistency, as well as addressing novel challenges such as progressive and approximative processing with performance guarantees as required by cognitive latencies (milliseconds vs seconds vs. minutes) in an iterative human data interaction process.

3.2 Towards making the database in*vis*ible

Tiziana Catarci (Sapienza University of Rome), Surajit Chaudhuri (Microsoft Research), Danyel Fisher (Microsoft Research), Heike Hofmann (Iowa State University), Tim Kraska (MIT), Zhicheng Liu (Adobe Systems Inc), Dominik Moritz (University of Washington), Gerik Scheuermann (Universität Leipzig), Richard Wesley (Tableau Software), Eugene Wu (Columbia University), Yifan Wu (University of California-Berkeley)

The rise of VisQL and its commercial counterpart Tableau highlight the value of a declarative visualization algebra to connect high level user task specification with the scalability and performance characteristics of data management engines. Arguably, establishing this language abstraction enabled the sub-field of OLAP-based visual analytics to take flight–visual analytics research could be grounded in a well-defined and important task definition with the promise that performance is a non-issue, while query execution and database optimizations could be targeted towards an important application domain.

For example, a visualization can provide hints to improve the performance of the database engine. Information such as the state of the visualization, the design of the user interface, and the past behaviors of the user can be used to provide hints to the database to preemptively execute subqueries, overcome the cold start problem, and improve caching strategies. Similarly, a visualization can be seen as a component of a federated data management system that automatically moves data and computation between multiple machines to improve performance. Data for a visualization can be modeled as a materialized view such that when updates occur because of arrival of new data (in a real-time system) or a user's interactions with the visualization, the federated system can automatically take care of data movement, streaming computation, and caching.

Data analysis systems designed with both database and visualization concerns in mind provide benefits beyond improvements in performance. For visualization designers and developers, an integrated system holds of promise of making the database *invisible*. That is, the visualization designer would prefer to not have to consider the manipulation of data in the database and in the visualization separately. Instead, with the proper language, API, and infrastructure, the visualization designer should be able to focus on design of the front-end interface without worrying about how the back-end system manages the data. Technologies that can make data management invisible to visualization designers can lead to a boom in the development of data systems in a similar way that D3 and other visualization languages have made data and visualization more accessible to the public.

There are exciting opportunities to advancing databases and visualization research in an integrated, synergistic way. However, there are also challenges and hurdles that prevent the two communities to immediate begin collaboration. These challenges stem from historical differences between the fields and gaps in needs and available technologies. In this section we explore and describe some of these challenges and possible ways to overcome them.

Challenge: Tooling

Current interactive visualizations are designed and constructed using low-level rendering libraries such as OpenGL, Java Swing, HTML Canvas or SVG, or high-level languages like D3, ggplot, Vega, VisQL, etc. These are either implemented as custom code by pulling data in memory (and thus restricted to smaller datasets), or dynamically translated to SQL in order to interoperate with existing data management systems.

How can we move from databases to data management systems for visualization? There is an impedance mismatch between languages for data visualization and data management. Databse APIs are not easy to use for visualization end-users. They do not clearly express visualization intent or state. Data queries have difficulty expressing queries that are core to visualization. And they require the visualization creator to manage issues that might be a database concern, such as where data is stored and how it is communicated. The goal is to develop a single language that allows a direct communication between visualizations and database systems. This might require a translation or even a replacement of traditional query languages such as SQL.

Challenge: Domain-Specific Operations

VizQL and OLAP offered remarkable abstractions for a specific subset of problems: data cubes, which map elegantly to core visualization types. The simple act of defining this subset of problems and drawing the connection to a subset of database queries had an immediate impact on progress in tooling, optimization, and performance. In the short-term, these observations serve as a template that may be applied to additional classes of task domains in order to draw a better understanding of the functionality and requirements that can be leveraged towards the Grand Unified Algebra. In fact a key challenge will be to uncover and identify the common analyses in order to aid the development of such an algebra.

The field of visualization is rich and diverse; different data types and computations that step away from those types are critical and important. In each of the following examples, the core challenge is to identify a crisp subset of operations that are "good enough" for a wide range of desired tasks and could possibly be implemented in an efficient manner. These operations include but are not limited to Complex Analyses (e.g. machine learning algorithms such as clustering, projection, etc.) and Time series, Sequential, Geographic, Hierarchical, Field (i.e. array-based) data manipulation. Identifying native database operations that can optimally support these common tasks in visualization is critical.

Challenge: Visualization Requirements

From the database perspective, one focus is on understanding novel and unique visualization-level requirements that current data management systems do not account for (e.g., progressiveness) as well as opportunities that are not currently exploited (e.g., user-thinking time). The current (imperfect) model is that a user-interaction in the visualization client issues a set of queries, along with visualization-level metadata that describes the result values as well as characteristics of how the results are returned.

In addition, there are immediate and longer term primitive operations that can be optimized towards these visualization-level requirements. Visualization-level requirements can be considered as constraints that are enforced by the data management system. Some possible data-related requirements may include: perceptual accuracy, performance (latency) constraint, progressive results overtime, consistency across views, etc.

Conversely, visualization-level information offers unique opportunities for data management right now. Such information can be assumed and exploited by the data management system to improve performance. For example, humans operate at a longer time-scale than the underlying system. This means there is ample time between human actions (either due to articulation time, or time to decide what to do next) to anticipate and prepare the system to answer the user's actions faster, more accurately, or with recommendations/explanations.

3.3 Understandability in Data Discoveries: an Integrated DB and Vis Perspective

Sihem Amer-Yahia (CNRS), Harish Doraiswamy (New York University), Steven Drucker (Microsoft Research), Juliana Freire (New York University), Daniel Keim (Universität Konstanz), Heike Leitte (TU Kaiserslautern), Alexandra Meliou (University of Massachusetts), Torsten Möller (Universität Wien), Thibault Sellam (Columbia University), Juan Soto (TU Berlin)

The overarching goal of human-data interaction systems is to simplify data analysis and make it accessible to a broad audience. These systems have evolved to provide access to a large range of statistical tools, interaction methods, and visualization facets, while automating many of the steps of the analysis process. The democratization of complex analysis tools through simple and accessible interfaces has enabled scientists and other experts to work more efficiently, and has allowed non-experts to partake in data discoveries.

Human-data interaction systems have simplified access to this array of powerful tools and would realize their full potential if they made those tools more understandable. Today, while users are able to interact with and analyze datasets faster and more easily, they often have a poor understanding of the steps involved in the analysis, and thus a poor understanding of the derived insights. As a result, human-data interaction systems face two important problems: (1) Users tend to rely on a very small subset of the analysis tools that these systems offer (and, typically, the less sophisticated ones, as those are more understandable). Thus, users fail to take advantage of the full power and capabilities of these systems. (2) The easy access to tools they understand poorly, can easily lead users to misuse, faulty workflows, and false discoveries. This problem is exacerbated with the automation of steps of the analysis process, as humans cannot easily process and critically reason about large numbers of derived steps and outcomes.

Understandability is thus a critical part of the data discovery process, and human-data interaction systems cannot push towards the goal of democratizing data analysis without support for understandability of the analysis steps. Supporting understandability in data discoveries involves several challenges, such as deriving the proper models and abstractions for explanation support in visual analytics, implementing “safety belts” to guard against tool misuse, and dealing with the computational and semantic explosion of the search space for explanations and hypotheses.

Challenge: Foundations for Understandability

This challenge addresses the essential question of what is understandability, how to capture it and how to convey it to different audiences, be they novice users, data scientists, or domain experts. Understandability can be approached from different angles and with different goals. It could serve the purpose of unveiling the relationship between input and output data, revealing underlying variables and models, explaining a decision process, or comparing different datasets and decisions. Understandability may be about exposing a statistical bias, a cognitive bias (e.g., confirmation bias), a perceptual bias, a method bias such as model selection, or a bias in the input data itself.

Understandability has been addressed extensively in the database and in the visualization communities with different goals and semantics.

For instance, in DB, perturbation analysis, outlier explanations, data cleaning, and provenance, are all forms of explanation. In the viz community, sensitivity analysis and looking for variables that cause bias. To understand anything, users need to be guided through a discovery process which itself requires to formalize different notions of understandability. The difficulty here is to devise a single, sufficiently expressive formalism to capture the different notions of understandability described above.

Challenge: Guiding and Guarding for (Genuine and False) Discoveries

Data analysis tools should help users from all backgrounds gain insights and draw conclusions from their data, by providing them with intuitive and systematic data exploration tool. This task is a challenge because many errors and biases can creep in the analysis. The statistics literature teaches us no dataset is unbiased, that correlation is not causation, that patterns can emerge out of sheer chance and that Simpson's paradox should be accounted for. And yet, no analyst is immune to mistakes and oversight. The challenge, as we see it, is: how do we provide data analysts of all levels of literacy more power while ensuring that the exploration is safe?

The problem can be addressed at different levels. One way is to design data analysis tools that are inherently safe given assumptions on the user's literacy. Another approach is to augment data analysis software with model checking primitives, to issue warnings when hypotheses are not fulfilled and highlight potential interpretation pitfalls. More generally, a pedagogy challenge is necessary to educate users for whom statistics is not a priority about common pitfalls and fallacies.

Challenge: Addressing Computational and Semantic Explosion

When it comes to enabling understandability and guiding and guarding for discovery, there is a combinatorial explosion that presents numerous challenges. Scalability only exacerbates this problem. For example, when understandability is about exposing relationships between inputs and outputs the explosion arises in the varying ways that data can be sliced and the varying semantics that reflect understandability. The exponential nature of this explosion raises both computational and semantic challenges.

Today, database and visualization research communities have developed pruning strategies and indexing mechanisms to reduce the complexity in exploring different data subsets. Human analysts have innate limitations when it comes to making sense of very large number of alternative semantics for understandability and the decoding of visualizations.

Database techniques for scalability can be used to provide different explanations/biases via guided interactions, faceted search, hypothesis testing and validation. From the visualization perspective, designers need to develop visualizations that can cope with the vast space of hypotheses and enable humans to best interpret what is observed.

3.4 Reducing Friction in Applying Data Management for Visualization

Michael Gleicher (University of Wisconsin-Madison), Behrooz Omidvar-Tehrani (LIG-Grenoble), Hannes Mühleisen (CWI), Themis Palpanas (Paris Descartes University)

Data visualization authors struggle to integrate data management systems, and therefore cannot reap the benefits that those provide. There are gaps between the real problems that visualization authors face in providing human-centered solutions, and suitable available data management solutions. Stakeholders of different kinds experience friction in the process, making moving across the gap difficult.

First, we consider "authors", engineers who design and implement visualization frameworks. "Consumers", on the other hand, who use those tools to explore data (e.g. domain scientists or even readers of data-interactive press reports). They are not necessarily experts in data management but instead have in-depth knowledge of a domain the visualization tools are supposed to be used in. Sometimes the author and consumer roles blur, such as in BI tools like Tableau.

We have identified two dimensions that make usage of readily available data management infrastructure difficult to the point where data management specialist knowledge is required: Rich Queries and Rich Data. Rich queries are a progression within the creative process of creating visualizations. The process often starts with a rough sense of what the ultimate results will be, so the visualization author has little idea how to describe them in terms of formalized queries. Here, a data management system that requires for example a fixed upfront schema definition and a set-theoretic approach to querying is of little help. As design ideas emerge, authors may have imprecise notions of what operations are needed to support them. As authors try to formalize these needs in order to move towards practical implementation, they often describe them using task-centric abstractions. That is, the abstractions that are convenient for describing the user needs. These abstractions may not map easily to data management abstractions, that is the things that are convenient for readily available data management systems, for example relational algebra.

Rich Data is the source of another dimension of complexity for the visualization author: Using traditional data management systems to process unstructured data (e.g. text data) is challenging as complex ETL pipelines using external tools would need to be built and complex statistical algorithms are required to make sense of the data. Specialized data management systems exist for individual forms of unstructured data but are then unlikely to easily support visualizations due to lack of higher-level analysis operators (e.g. inverted indices). A similar issue exists for structured data such as graphs or arrays. Again, specialized data management systems exist, but non-relational querying operators are often required. This creates additional adoption barriers for visualization authors as new systems and query languages need to be adopted for each of the various types of structured data.

3.5 An Evaluation Methodology for Vis + DB

Leilani Battle (University of Maryland), Carsten Binnig (TU Darmstadt), Tiziana Catarci (Sapienza University of Rome), Stephan Diehl (Universität Trier), Jean-Daniel Fekete (INRIA Saclay), Michael Sedlmair (Universität Wien), Wesley Willett (University of Calgary)

There exists a growing set of data-centric systems through which domain experts and data scientists of varying skill levels can visually analyze large data sets. Evaluating such systems, however, is non-trivial. To that end, we need a more principled methodology for evaluating specifically components that live at the intersection of visual interactive interfaces and the data management system behind it. While the long term goal is to have general user and task models that would allow to derive realistic benchmarks for different scenarios, this seems clearly impractical to solve in the short term. Therefore, we advocate to start with more empirical methods.

To address our challenges, we need to evaluate visual data analysis systems at different levels. On the work environment level, we need to better understand what data scientists/domain scientists are doing and how they are working, i.e. which are their goals, tasks, and constraints. On the system level we should collect and share concrete and systematic traces of real-world analysis and exploration activity (logs, video, annotations, etc.) that could drive more realistic evaluation of DM systems, and better inform Vis designs. On the component level we need to provide better benchmarks for making DB systems ready for EDA/Vis. We also need to support DB researchers in more user-centered evaluations. We suggest that the Vis and DB communities work together on a bottom-up approach, starting on the component level up to the work environment level, to developing a unified evaluation methodology that spans the different levels.

In the database community, there is a strong emphasis on using benchmarks to test the performance of different data management systems, where individual benchmarks are developed to simulate certain tasks. For example, transactional processing (TPC-C and TPC-E), data cleaning (TPC-DI), and online analytical processing (TPC-H and TPC-DS) are all areas of interest for data management benchmarks. However, these benchmarks focus on simulating large-scale enterprise systems (e.g., data warehouses), and generally ignore the case where database queries are being generated by a user interacting with a visualization interface.

Similarly, there is an extensive body of work in the visualization community on understanding, refining and improving the process of evaluating visualization tools. For example, by analyzing the large number of user studies conducted over the years, the visualization community has been able to: 1) layout the space of possible evaluations, 2) encourage the use of evaluation methodologies from other areas for visualization research, and 3) distill best practices for conducting future user study evaluations.

While evaluation methods exist for both communities, there are relatively few quantitative and comparative methods for evaluating the performance of large-scale visual exploration systems. In particular, user studies are designed and conducted to perform customized evaluations of individual systems, but the user interaction logs, system logs and metadata (e.g., datasets used, interfaces used) are rarely shared or published in a standardized way. There has been some initial work on creating a repository for benchmarking the efficacy of visualization interfaces, but there is still a need for recording data at the systems level to help standardize the evaluation process.

4 Community Challenges

Sihem Amer-Yahia (CNRS), Leilani Battle (University of Maryland), Carsten Binnig (TU Darmstadt), Tiziana Catarci (Sapienza University of Rome), Remco Chang (Tufts University), Surajit Chaudhuri (Microsoft Research), Harish Doraiswamy (New York University), Steven Drucker (Microsoft Research), Stephan Diehl (Universität Trier), Harish Doraiswamy (New York University), Jason Dykes (City University of London), Jean-Daniel Fekete (INRIA Saclay), Danyel Fisher (Microsoft Research), Juliana Freire (New York University), Michael Gleicher (University of Wisconsin-Madison), Hans Hagen (TU Kaiserslautern), Gerhard Heyer (Universität Leipzig), Heike Hofmann (Iowa State University), Daniel Keim (Universität Konstanz), Tim Kraska (MIT), Heike Leitte (TU Kaiserslautern), Zhicheng Liu (Adobe Systems Inc), Volker Markl (TU Berlin), Alexandra Meliou (University of Massachusetts), Torsten Moller (Universität Wien), Dominik Moritz (University of Washington), Hannes Mühleisen (CWI), Arnab Nandi (Ohio State University), Behrooz Omidvar-Tehrani (LIG-Grenoble), Themis Palpanas (Paris Descartes University), Carlos Scheidegger (University of Arizona), Gerik Scheuermann (Universität Leipzig), Michael Sedlmair (Universität Wien), Thibault Sellam (Columbia University), Juan Soto (TU Berlin), Richard Wesley (Tableau Software), Wesley Willett (University of Calgary), Eugene Wu (Columbia University), Yifan Wu (University of California-Berkeley)

The database community has developed solutions for many data management challenges that are relevant to the visualization community. Our goal should be to bring these solutions to a usable stage rather than a "research-solved" stage. In order to address the needs of the visualization community, it is important to have a common understanding and arrive at a common language. This may require the database community to abstract operations with richer semantics than relational algebra, and it will require the visualization community to specify common visualization operators jointly with the semantics.

Both communities understand these problems cannot be solved alone. The database community needs to think more about not just simple, commonly-used use cases, but also nuanced complex use cases faced by the visualization community and create software artifacts, publications, and research projects. The visualization community, at the same time, needs to better articulate needs at a lower-level of the stack, so as to allow for actionable areas of need. By incrementally working on smaller pieces on both sides, this allows for both communities to create layers of interaction, such as a shared algebra and grammar/language to reason about and build systems on.

Papers in this domain may face challenges of different publication cultures, as traditional evaluation metrics in the database community focus on systems behavior (latency, throughput, etc.). Evaluation visualizations, on the other hand, may require modeling user behavior and user studies, a concept not yet well established in database conference. This may require educating program committee members of both communities aware of the publication culture and importance of the respective metrics of the other field. In order to bring the communities together, it may be useful to establish workshops on this joint topic at the key conferences of both communities such as SIGMOD, ICDE as well as VIS. The HILDA workshop at SIGMOD and the DSIA workshop at VIS are two example of this. In order to achieve cross-fertilization, co-location of conferences of both communities and cross-selection of invited speakers will be useful instruments.

5 Conclusion

As data scale and complexity increases, the novel solutions that will ultimately enable interactive, large-scale exploratory data analysis will have to come from truly interdisciplinary and international work. Today, databases can store and query massive amounts of data, including methods for distributed, streaming and approximate computation. Data mining techniques provide ways to discover unexpected patterns and to automate and scale well-defined analysis procedures. Meanwhile, to keep up with the rising tide of data, visualization researchers develop interactive visualization tools to support human perception and cognition that help users reason about the data and gain insights from the data.

This seminar brought together researchers from the database and visualization communities who have clear overlaps in interest and goals, but seldom interact with each other. It became clear from the week-long discussions that the two communities need to collaborate and rely on each other to achieve the goal of supporting users in performing interactive data analysis on data that continue to increase in scale and complexity. In this report we summarized these discussions – we have identified areas of synergistic research activities, highlighted potential new exciting research opportunities, and suggested ways in each the two communities can work together.

We thank Dagstuhl for providing a wonderful environment for this meeting to take place. The opportunity for open dialog and formal (and informal) discussions was appreciated by all the participants. We hope that the outcome of our discussions can represent the start of a new research agenda that will lead to novel research ideas and useful software artifacts. Further, we hope that this research agenda will inspire researchers from both the database and the visualization communities to continue the discussions and form new collaborations towards the development of next generation data management and interactive visualization systems.

Participants

- Sihem Amer-Yahia
CNRS – St. Martin-d'Hères, FR
- Leilani Battle
University of Washington – Seattle, US
- Carsten Binnig
TU Darmstadt, DE
- Tiziana Catarci
Sapienza University of Rome, IT
- Remco Chang
Tufts University – Medford, US
- Surajit Chaudhuri
Microsoft Research – Redmond, US
- Stephan Diehl
Universität Trier, DE
- Harish Doraiswamy
New York University, US
- Steven M. Drucker
Microsoft Research – Redmond, US
- Jason Dykes
City – University of London, GB
- Jean-Daniel Fekete
INRIA Saclay – Orsay, FR
- Danyel Fisher
Microsoft Research – Redmond, US
- Juliana Freire
New York University, US
- Michael Gleicher
University of Wisconsin – Madison, US
- Hans Hagen
TU Kaiserslautern, DE
- Gerhard Heyer
Universität Leipzig, DE
- Heike Hofmann
Iowa State University – Ames, US
- Daniel A. Keim
Universität Konstanz, DE
- Tim Kraska
Brown University – Providence, US
- Heike Leitte
TU Kaiserslautern, DE
- Zhicheng Liu
Adobe Systems Inc. – Seattle, US
- Volker Markl
TU Berlin, DE
- Alexandra Meliou
University of Massachusetts – Amherst, US
- Torsten Möller
Universität Wien, AT
- Dominik Moritz
University of Washington – Seattle, US
- Hannes Mühleisen
CWI – Amsterdam, NL
- Arnab Nandi
Ohio State University – Columbus, US
- Behrooz Omidvar-Tehrani
LIG – Grenoble, FR
- Themis Palpanas
Paris Descartes University, FR
- Carlos E. Scheidegger
University of Arizona – Tucson, US
- Gerik Scheuermann
Universität Leipzig, DE
- Michael Sedlmair
Universität Wien, AT
- Thibault Sellam
Columbia University – New York, US
- Juan Soto
Technische Universität Berlin, DE
- Richard Wesley
Tableau Software – Seattle, US
- Wesley J. Willett
University of Calgary, CA
- Eugene Wu
Columbia University – New York, US
- Yifan Wu
University of California – Berkeley, US

Report from Dagstuhl Seminar 17462

A Shared Challenge in Behavioural Specification

Edited by

Klaus Havelund[1], Martin Leucker[2], Giles Reger[3], and Volker Stolz[4]

1 **Jet Propulsion Laboratory, US,** klaus.havelund@jpl.nasa.gov
2 **Universität Lübeck, DE,** leucker@isp.uni-luebeck.de
3 **University of Manchester, GB,** giles.reger@manchester.ac.uk
4 **West. Norway Univ. of Applied Sciences – Bergen, NO,** vsto@hvl.no

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 17462 "A Shared Challenge in Behavioural Specification". The seminar considered the issue of behavioral specification with a focus on its usage in Runtime Verification. The seminar was motivated by the observations that, whilst the field of Runtime Verification is becoming more mature, there is a lack of common specification language, in the main part due to the rich setting allowing for highly expressive languages. The aim of the Seminar was to shed light on the similarities and differences between the different existing languages, and specifically, suggest directions for future collaboration and research. The seminar consisted of two talk sessions, two working group sessions, and a feedback and reflection session. Working group topics were suggested and agreed in response to points raised in talks. One significant outcome was the proposal of a shared challenge project in which different Runtime Verification approaches can be compared, as outlined in one of the working group reports.

Seminar November 12–15, 2017 – http://www.dagstuhl.de/17462
1998 ACM Subject Classification D.2.4 Software/Program Verification, F.3.1 Specifying and Verifying and Reasoning about Programs, F.4.3 Formal Languages
Keywords and phrases behavioural specification, dynamic properties, runtime verification, temporal logic
Digital Object Identifier 10.4230/DagRep.7.11.59

1 Executive Summary

Giles Reger
Klaus Havelund
Martin Leucker
Volker Stolz

This seminar dealt with the issue of behavioural specification from the viewpoint of runtime verification. Runtime verification (RV) as a field is broadly defined as focusing on processing execution traces (output of an observed system) for verification and validation purposes. Of particular interest is the problem of verifying that a sequence of events, a trace, satisfies a temporal property, formulated in a suitable formalism. Examples of such formalisms include state machines, regular expressions, temporal logics, context-free grammars, variations of the mu-calculus, rule systems, stream processing systems, and process algebras. Of special

A Shared Challenge in Behavioural Specification, *Dagstuhl Reports*, Vol. 7, Issue 11, pp. 59–85
Editors: Klaus Havelund, Martin Leucker, Giles Reger, and Volker Stolz
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

interest is how to specify data-rich systems, where events themselves carry data. Applications cover such domains as security monitoring and safety monitoring.

Such techniques are characterised by highly expressive languages for specifying behaviour, enabled by the concreteness of dealing directly with single runtime traces, which makes the verification problem tractable. However, this permitted expressiveness has also led to a divergence in such languages. The aim of this Dagstuhl Seminar was to shed light on the similarities and differences between these different formalisms, and specifically, suggest directions for future collaboration and research. This effort can potentially lead to an attempt to standardize an RV formalism.

The seminar included a mixture of tool developers, theoreticians, and industry experts and the above aim was addressed by two main activities.

The first activity was that each tool developer was asked to produce a brief summary of their specification language in the form of a set of short examples. These were then presented as talks during the Seminar, alongside other general contributed talks on issues surrounding behavioural specification. The examples were uploaded to a shared repository (which will be available via runtime-verification.org) and eleven participants added their tool descriptions and examples to this repository, producing a lasting resource from the seminar.

The second activity was carried out through eight working groups formed during the Seminar to discuss topics raised by the talks. The results of this working groups are detailed in this report. We take this opportunity to detail the topics (in the form of questions) proposed during the seminar that were not chosen for discussion in working groups:

- *Where should we get specifications from?* This question addressed both the issue of designing specification languages that can be usable by engineers but also the trending topic of inferring specifications from various artifacts and how specification languages can support this.
- *How can we measure specification quality?* What is a good specification, or when is one specification better than another? This might be related to coverage of the system being specified, or might be about interpretability or some other measure of usability.
- *How do we ensure our specification language is not broken?* This question was inspired by the experience of one speaker with developing the industrial-strength PSL language and the issues surrounding getting it right.
- *How can we balance different levels of abstraction (e.g. local and global behaviour) in a specification?* It was noted that specification languages are often closely associated with specifications at a certain level of abstraction. Is this an inherent restriction or a positive feature? Should we build specification languages with a certain level of abstraction in mind?
- *How do we unify the different uses of a specification?* This was inspired by the observation that a specification may be used to explain behaviour, check behaviour, or synthesize behaviour, and different presentations may be preferred in these different contexts.

This seminar was the first time the runtime verification community has reflected on the broad issue of specification and has fed into further developments including new perspectives for the international runtime verification competition, a proposed shared challenge involving the NASA core flight system, and the first informal survey and categorisation of actively developed runtime verification tools.

2 Table of Contents

Executive Summary

Giles Reger, Klaus Havelund, Martin Leucker, and Volker Stolz 59

Overview of Talks

Trace Focussed and Data Focussed Specification: Complementary, Competing, Combined?
Wolfgang Ahrendt 63

Domain-Specific Languages with Scala, and model-based testing as an example
Cyrille Artho 64

The Tale of Dr Jekyll and Mr Hyde in Pattern-based Specification Languages
Domenico Bianculli 64

Asynchronous HyperLTL
Borzoo Bonakdarpour 65

PSL: The good, the bad and the ugly
Cindy Eisner 66

Two to Tango: A pair of specification languages for runtime monitoring
Adrian Francalanza 67

A “Do-It-Yourself” Specification Language With BeepBeep 3
Sylvain Hallé 67

Linking Heterogeneous Models for Intelligent-Cyber-Physical Systems
Zhiming Liu 68

Specification Languages for CPS
Dejan Nickovic 68

Automaton-Based Formalisms for Runtime Verification
Gordon Pace 69

What is parametric trace slicing good for?
Giles Reger 69

Specification: The Biggest Bottleneck in Formal Methods and Autonomy
Kristin Yvonne Rozier 70

TeSSLa: A Real-Time Specification Language for Runtime Verification of Non-synchronized Streams
Torben Scheffel 70

E-ACSL, an Executable Behavioural Interface Specification Language for C Programs
Julien Signoles 71

LOLA
Hazem Torfah 71

Metric First-Order Dynamic Logic and Beyond
Dmitriy Traytel 72

Behavioural Type-Based Static Verification Framework for Go
Nobuko Yoshida 72

Working groups

How do we integrate trace behaviour with state properties
Wolfgang Ahrendt, Stijn de Gouw, Adrian Francalanza, Zhiming Liu, and Julien Signoles . . . 73

Specifications that are like implementations
Cyrille Artho, Cindy Eisner, Keiko Nakata, Dejan Nickovic, and Volker Stolz . . . 73

Property Specification Patterns for Runtime Verification
Domenico Bianculli, Borzoo Bonakdarpour, Bernd Finkbeiner, Gordon Pace, Giles Reger, Kristin Yvonne Rozier, Gerardo Schneider, Dmitriy Traytel, and Nobuko Yoshida . . . 75

Exploring the tradeoffs between Declarative and Operational Specification
Adrian Francalanza, Wolfgang Ahrendt, Cindy Eisner, Zhiming Liu, and Gordon Pace 77

Event Stream Processing
Sylvain Hallé, Martin Leucker, Nicolas Rapin, César Sánchez, Torben Scheffel, and Hazem Torfah . . . 78

A shared challenge – NASA's Core Flight System
Volker Stolz, Borzoo Bonakdarpour, Martin Leucker, Nicolas Rapin, Kristin Yvonne Rozier, Julien Signoles, Hazem Torfah, and Nobuko Yoshida . . . 80

Data Quantification in Temporal Specification Languages
Dmitriy Traytel, Domenico Bianculli, and Giles Reger . . . 82

Participants . . . 85

3 Overview of Talks

3.1 Trace Focussed and Data Focussed Specification: Complementary, Competing, Combined?

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

5 Years ago, I participated in Dagstuhl seminar *Divide and Conquer: the Quest for Compositional Design and Analysis*. In effect, the seminar could have been named *Model Checking meets Deductive Verification*. It was a very interesting seminar, but we had some difficulties to identify the types of *properties*, and with them specification formalisms, which both communities are interested in, or can cope with using their respective technologies.

Property languages are often technology driven, and so are properties themselves. To analyse one system with *different methods*, we end up using *different fomalisms*, specifying *disconnected views*.

In Runtime Verification, as well as in Model Checking, there is a strong focus on traces, often traces of events of some kind. In Deductive Versification, as well as in Runtime Assertion Checking, the focus on properties of the data, at specific points in the execution. Is the difference really motivated by what either communities consider important system properties, or rather by what the respective technologies are good at checking? To which extent should specification formalisms make a pre-choice?

In my talk, I suggest the following community effort:

- Integrated/coordinated specification of trace *and* data focused aspects,
- Front-ends mapping divers aspects of the specification to tool/method-oriented formats,
- Delegation of sub-tasks to appropriate tools
- Delegation of sub-tasks to static or dynamic analysis
- Integration of the results from diverse analyses

I mentioned ppDATE [1] as one specific attempt to combine multiple aspects on the specification level. With that, I hope to trigger a more general discussion about the role and integration of trace focused and data focused specification.

References

1 Wolfgang Ahrendt, Jesús Mauricio Chimento, Gordon J. Pace, and Gerardo Schneider. Verifying data- and control-oriented properties combining static and runtime verification: theory and tools. *Formal Methods in System Design*, Apr 2017.

3.2 Domain-Specific Languages with Scala, and model-based testing as an example

Cyrille Artho (KTH Royal Institute of Technology – Stockholm, SE)

Joint work of Cyrille Artho, Klaus Havelund, Rahul Kumar, Yoriyuki Yamagata
Main reference Cyrille Artho, Klaus Havelund, Rahul Kumar, Yoriyuki Yamagata: "Domain-Specific Languages with Scala", in Proc. of the Formal Methods and Software Engineering - 17th International Conference on Formal Engineering Methods, ICFEM 2015, Paris, France, November 3-5, 2015, Proceedings, Lecture Notes in Computer Science, Vol. 9407, pp. 1–16, Springer, 2015.
URL https://doi.org/10.1007/978-3-319-25423-4_1

Domain-Specific Languages (DSLs) are often classified into external and internal DSLs. An external DSL is a stand-alone language with its own parser. An internal DSL is an extension of an existing programming language, the host language, offering the user of the DSL domain-specific constructs as well as the constructs of the host language.

This presentation gives a brief overview of the concepts and also looks at an internal DSL used for model-based testing with the tool "Modbat".

3.3 The Tale of Dr Jekyll and Mr Hyde in Pattern-based Specification Languages

Domenico Bianculli (University of Luxembourg, LU)

Joint work of Marcello Maria Bersani, Lionel Briand, Wei Dou, Carlo Ghezzi, Srdan Krstic, Pierluigi San Pietro
Main reference Domenico Bianculli, Carlo Ghezzi, Pierluigi San Pietro: "The Tale of SOLOIST: A Specification Language for Service Compositions Interactions", in Proc. of the Formal Aspects of Component Software, 9th International Symposium, FACS 2012, Mountain View, CA, USA, September 12-14, 2012. Revised Selected Papers, Lecture Notes in Computer Science, Vol. 7684, pp. 55–72, Springer, 2012.
URL http://dx.doi.org/10.1007/978-3-642-35861-6_4
Main reference Wei Dou, Domenico Bianculli, Lionel C. Briand: "A Model-Driven Approach to Trace Checking of Pattern-Based Temporal Properties", in Proc. of the 20th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2017, Austin, TX, USA, September 17-22, 2017, pp. 323–333, IEEE Computer Society, 2017.
URL http://dx.doi.org/10.1109/MODELS.2017.9

This talk presents two specification languages, SOLOIST [6] and TemPsy [8, 7]. Both are based on property specification patterns [5, 9] and have been defined in the context of an industrial collaboration.

SOLOIST (*SpecificatiOn Language fOr servIce compoSitions inTeractions*) is a metric temporal logic with new, additional temporal modalities that support aggregate operations on events occurring in a given time window; it can be used to specify both functional and quality-of-service requirements of the interactions of a composite service with its partner services. The trace checking algorithms proposed for SOLOIST rely on the use of SMT solvers [4, 1] and of distributed and parallel computing frameworks [2, 3].

TemPsy (Temporal Properties made easy) is a pattern-based, domain-specific specification language for temporal properties. Its goal is to make as easy as possible the specification of the temporal requirements of business processes, by supporting only the constructs needed to express temporal requirements commonly found in business process applications. TemPsy comes with an efficient trace checking algorithm [8] which relies on a mapping of temporal requirements written in TemPsy into Object Constraint Language (OCL) constraints on a conceptual model of execution traces.

References

1 Marcello Maria Bersani, Domenico Bianculli, Carlo Ghezzi, Srdan Krstić, and Pierluigi San Pietro. SMT-based checking of SOLOIST over sparse traces. In *Proceedings of the 17th International Conference on Fundamental Approaches to Software Engineering (FASE 2014), Grenoble, France*, volume 8411 of *Lecture Notes in Computer Science*, pages 276–290. Springer, April 2014.

2 Marcello Maria Bersani, Domenico Bianculli, Carlo Ghezzi, Srdan Krstić, and Pierluigi San Pietro. Efficient large-scale trace checking using MapReduce. In *Proceedings of the 38th International Conference on Software Engineering (ICSE 2016), Austin, TX, USA*, pages 888–898. ACM, May 2016.

3 Domenico Bianculli, Carlo Ghezzi, and Srdan Krstić. Trace checking of metric temporal logic with aggregating modalities using MapReduce. In *Proceedings of the 12th International Conference on Software Engineering and Formal Methods (SEFM 2014), Grenoble, France*, volume 8702 of *Lecture Notes in Computer Science*, pages 144–158. Springer, September 2014.

4 Domenico Bianculli, Carlo Ghezzi, Srdan Krstić, and Pierluigi San Pietro. Offline trace checking of quantitative properties of service-based applications. In *Proceedings of the 7h International Conference on Service Oriented Computing and Application (SOCA 2014), Matsue, Japan*, pages 9–16. IEEE, November 2014.

5 Domenico Bianculli, Carlo Ghezzi, Cesare Pautasso, and Patrick Senti. Specification patterns from research to industry: a case study in service-based applications. In *Proceedings of the 34th International Conference on Software Engineering (ICSE 2012), Zürich, Switzerland*, pages 968–976. IEEE Computer Society Press, June 2012.

6 Domenico Bianculli, Carlo Ghezzi, and Pierluigi San Pietro. The tale of SOLOIST: a specification language for service compositions interactions. In *Proceedings of the 9th International Symposium on Formal Aspects of Component Software (FACS'12), Mountain View, CA, USA*, volume 7684 of *Lecture Notes in Computer Science*, pages 55–72. Springer, September 2012.

7 Wei Dou, Domenico Bianculli, and Lionel Briand. OCLR: a more expressive, pattern-based temporal extension of OCL. In *Proceedings of the 2014 European Conference on Modelling Foundations and Applications (ECMFA 2014), York, United Kingdom*, volume 8569 of *Lecture Notes in Computer Science*, pages 51–66. Springer, July 2014.

8 Wei Dou, Domenico Bianculli, and Lionel Briand. A model-driven approach to trace checking of pattern-based temporal properties. In *Proceedings of the 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS 2017), Austin, TX, USA*. IEEE, September 2017.

9 Matthew B Dwyer, George S Avrunin, and James C Corbett. Patterns in property specifications for finite-state verification. In *Proc. ICSE 1999*, pages 411–420, New York, NY, USA, 1999. ACM.

3.4 Asynchronous HyperLTL

Borzoo Bonakdarpour (McMaster University – Hamilton, CA)

HyperLTL is a temporal logic for expressing a subclass of hyperproperties. It allows explicit quantification over traces and inter-trace Boolean relations among traces. The current semantics of HyperLTL evaluate formula by progressing a set of traces in a lock-step synchronous manner. In this talk, we will present our recent work on relaxing the semantics of HyperLTL to allow traces to advance with different speeds. While this relaxation makes the verification problem undecidable, the decidable fragment is expressive enough to express most commonly used security policies. Our new semantics has also application in model-based runtime monitoring.

3.5 PSL: The good, the bad and the ugly

Cindy Eisner (IBM – Haifa, IL)

Joint work of Cindy Eisner, Dana Fisman, John Havlicek, Yoad Lustig, Anthony McIsaac, Johan Mårtensson, David Van Campenhout

For a specification language to be suitable for formal verification by model checking, it must have sufficient expressive power, its semantics must be formally defined in a rigorous manner, and the complexity of model checking it must be well understood and reasonable. In order to allow widespread adoption in the industry, there is an additional requirement: behavioral specification must be made easy, allowing common properties to be expressed intuitively and succinctly. But while adding syntax is simple, defining semantics without breaking important properties of the existing semantics is surprisingly difficult. In this talk I will discuss various extensions to temporal logic incorporated by PSL, their motivation, and the subtle semantic issues encountered in their definition. I will emphasize where we succeeded, where we were less successful, and point out some features that are still missing.

References

1 C. Eisner and D. Fisman. Augmenting a regular expression-based temporal logic with local variables. In A. Cimatti and R. B. Jones, editors, Formal Methods in Computer Aided Design (FMCAD), pages 1–8. IEEE, 2008.

2 C. Eisner and D. Fisman. Structural contradictions. In H. Chockler and A. J. Hu, editors, Intl. Haifa Verification Conference (HVC), volume 5394 of Lecture Notes in Computer Science, pages 164–178. Springer, 2008.

3 C. Eisner, D. Fisman, and J. Havlicek. A topological characterization of weakness. In M. K. Aguilera and J. Aspnes, editors, Symp. on Principles of Distributed Computing (PODC), pages 1–8. ACM, 2005.

4 C. Eisner, D. Fisman, and J. Havlicek. Safety and liveness, weakness and strength, and the underlying topological relations. ACM Trans. Comput. Log., 15(2), 2014.

5 C. Eisner, D. Fisman, J. Havlicek, Y. Lustig, A. McIsaac, and D. Van Campenhout. Reasoning with temporal logic on truncated paths. In W. A. H. Jr. and F. Somenzi, editors, Intl. Conf. on Computer-Aided Verification (CAV), volume 2725 of Lecture Notes in Computer Science, pages 27–39. Springer, 2003.

6 C. Eisner, D. Fisman, J. Havlicek, and J. Mårtensson. The top,bot approach for truncated semantics.Technical Report 2006.01, Accellera, May 2006.

7 C. Eisner, D. Fisman, J. Havlicek, A. McIsaac, and D. Van Campenhout. The definition of a temporal clock operator. In J. C. M. Baeten, J. K. Lenstra, J. Parrow, and G. J.Woeginger, editors, Intl. Coll. on Automata, Languages and Programming (ICALP), volume 2719 of Lecture Notes in Computer Science, pages 857–870. Springer, 2003.

3.6 Two to Tango: A pair of specification languages for runtime monitoring

Adrian Francalanza (University of Malta – Msida, MT)

The choice of a specification language is ultimately determined by its intended use. In this talk we motivate the need to employ two specification languages to be able to study the problem of monitorability. In particular, we will outline why we chose a variant of the modal-mu calculus on the one hand, and a process calculus on the other to understand formally what can and cannot be monitored for at runtime. We will also overview how the choice of two formalisms can be used to asses the correctness of a monitor that is entrusted with checking the execution of a system against a specification.

3.7 A "Do-It-Yourself" Specification Language With BeepBeep 3

Sylvain Hallé (University of Quebec at Chicoutimi, CA)

Main reference Sylvain Hallé: "When RV Meets CEP", in Proc. of the Runtime Verification - 16th International Conference, RV 2016, Madrid, Spain, September 23-30, 2016, Proceedings, Lecture Notes in Computer Science, Vol. 10012, pp. 68–91, Springer, 2016.
URL https://doi.org/10.1007/978-3-319-46982-9_6

BeepBeep is an event stream processing engine [1]. In BeepBeep, streams of events (of any kind) are piped into computing units called *processors*. The output of processors can be used as the input of other processors, forming potentially complex processor chains.

BeepBeep is organized along a modular architecture. The main part of BeepBeep is called the *engine*, which provides the basic classes for creating processors and functions, and contains a handful of general-purpose processors for manipulating traces. The rest of BeepBeep's functionalities is dispersed across a number of optional *palettes*.

BeepBeep provides multiple ways to create processor pipes and to fetch their results. A first obvious way is programmatically, using BeepBeep as a library and Java as the glue code for creating the processors and connecting them. In addition to directly instantiating and connecting processors, BeepBeep also offers the possibility to create domain-specific languages with subsets of all the available processors. To this end, BeepBeep provides a parser, which can be given a BNF grammar at runtime, and create a parse tree from any string. With the help of a grammar file and a custom-made "expression parser", one can hence create, in a few lines of code, a domain-specific language with an arbitary syntax, and a parser that converts an expression of this language into a BeepBeep processor chain.

References

1 S. Hallé. "When RV Meets CEP". Proc. RV 2016, Springer Lecture Notes in Computer Science 10012, 68-91, 2016.

3.8 Linking Heterogeneous Models for Intelligent-Cyber-Physical Systems

Zhiming Liu (Southwest University – Chongqing, CN)

Compared to the challenges in traditional ICT applications that engineers used to face, CPS systems and their software development are to, based on the infrastructures of existing systems, develop new components or subsystems, new applications and front end services and to integrate them onto the existing systems. Such development and integration have to deal with the complexity of ever evolving architectures digital components, physical components, together with sensors and smart devices controlled and coordinated by software. The architectural components are designed with different technologies, run on different platforms and interact through different communication technologies. Software components run in these systems for data processing and analytics, computation, and intelligent control. The requirements and environment of a CPS components keep changing with significant uncertainty. Thus, a CPS must contain dynamic monitors and adapters. In this talk we intend to discuss the need of a foundation for the combination of traditional software engineering and AI (or knowledge-based engineering) and the emerging Big Data technologies. We propose research problems including monitoring AI (including learning systems) components, end-to-end specification of composition learning and non-learning components, and a unifying modeling theory to link the different modeling paradigms of non-learning software components and learning software components. We believe that the unified modeling framework need to combine models data functionality, interaction protocols, and timing in both declarative and operational languages, but yet it has to support separation of different design and verification concerns.

3.9 Specification Languages for CPS

Dejan Nickovic (AIT Austrian Institute of Technology – Wien, AT)

Continuous and hybrid behaviors naturally arise from cyber-physical systems (CPS). In this talk, we will present a brief overview of the specification languages that were designed to tackle CPS-specific properties. We will mainly focus on Signal Temporal Logic (STL) and Timed Regular Expressions (TRE), but will also present their syntactic and semantic extensions. We will discuss what are the strength and weaknesses of these languages and in which situations they should or should not be used.

3.10 Automaton-Based Formalisms for Runtime Verification

Gordon Pace (University of Malta – Msida, MT)

An open question is the appropriateness of logic-based vs. visual, graph-based formalisms to be used as specification languages. Clearly there are different ways in which one may measure appropriateness, ranging from ease of writing, ease of comprehension, maintainability of specifications, efficiency of verification and conciseness to mention but a few. Over the past years, we have used graph-based formalisms in various projects and domains, with a particular focus on their use in runtime verification. The formalisms used range from DATEs (the formalism used by the Larva runtime verification tool), ppDATEs (an extension of DATEs used by StaRVOOrs static+dynamic analysis tool), contract automata (used to formalise contracts, including obligations, prohibitions and permissions) and policy automata (used to formalise social network privacy policies). The primary drive towards using these formalisms was the ease of adoption from industrial partners, and correspondence to models typically documented (or not) of the lifecycles of entities in such systems. We present these formalisms and issues arising from their use, and go on to discuss formalisms lying above and below this automaton-based level of abstraction – outlining our experience with controlled natural languages (above) and guarded-commands (below).

3.11 What is parametric trace slicing good for?

Giles Reger (University of Manchester, GB)

Parametric trace slicing [2, 6] is an approach for parametric runtime monitoring that was introduced by tracematches and JavaMOP [1] and later extended by QEA [3, 4, 5]. In this talk I will briefly discuss what it is good for and, perhaps more interesting, what it is not good for. I argue that this approach is very efficient where we have a few quantified variables and care about the cross-product of their domains (such situations arise reasonably often when reasoning about API usage). I argue that language based on this approach tend to be less intuitive and 'non-local' (i.e. you always need to consider the whole specification when considering each part). Additionally, specifications tend not to be composable.

References

1 Meredith, P.O., Jin, D., Griffith, D. et al. Int J Softw Tools Technol Transfer (2012) 14: 249.

2 Chen F., Roşu G. (2009) Parametric Trace Slicing and Monitoring. In: Kowalewski S., Philippou A. (eds) Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2009. Lecture Notes in Computer Science, vol 5505. Springer, Berlin, Heidelberg

3 Barringer H., Falcone Y., Havelund K., Reger G., Rydeheard D. (2012) Quantified Event Automata: Towards Expressive and Efficient Runtime Monitors. In: Giannakopoulou D., Méry D. (eds) FM 2012: Formal Methods. FM 2012. Lecture Notes in Computer Science, vol 7436. Springer, Berlin, Heidelberg

4 Havelund K., Reger G. (2015) Specification of Parametric Monitors. In: Drechsler R., Kühne U. (eds) Formal Modeling and Verification of Cyber-Physical Systems. Springer Vieweg, Wiesbaden

5 Reger G., Cruz H.C., Rydeheard D. (2015) MarQ: Monitoring at Runtime with QEA. In: Baier C., Tinelli C. (eds) Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2015. Lecture Notes in Computer Science, vol 9035. Springer, Berlin, Heidelberg

6 K. Havelund and G. Reger and E. Zalinescu and D. Thoma Monitoring Events that Carry Data, In E. Bartocci and Y. Falcone, editors, Lectures on Runtime Verification – Introductory and Advanced Topics, volume 10457 of *LNCS*, pages 60–97, Springer, 2018

3.12 Specification: The Biggest Bottleneck in Formal Methods and Autonomy

Kristin Yvonne Rozier (Iowa State University, US)

Advancement of increasingly AI-enhanced control in autonomous systems stands on the shoulders of formal methods, which make possible the rigorous safety analysis autonomous systems require. Formal methods are highly dependent on the specifications over which they reason; not only are specifications required for analysis, but there is no escaping the “garbage in, garbage out” reality. Correct, covering, and formal specifications are thus an essential element for enabling autonomy. However, specification is difficult, unglamorous, and arguably the biggest bottleneck facing verification and validation of aerospace, and other, autonomous systems. This talk will examine the outlook for the practice of formal specification, and highlight the on-going challenges of specification, from design-time to runtime system health management. We will pose challenge questions for specification that will shape both the future of formal methods, and our ability to more automatically verify and validate autonomous systems of greater variety and scale.

3.13 TeSSLa: A Real-Time Specification Language for Runtime Verification of Non-synchronized Streams

Torben Scheffel (Universität Lübeck, DE)

Joint work of Torben Scheffel, Sebastian Hungerecker, Martin Leucker, Malte Schmitz, Daniel Thoma

The Temporal Stream-based Specification Language (TeSSLa) operates on non-synchronized real-time streams. It was first created for specifying properties about programs running on multi core systems and it is currently used and developed in the COEMS project.

TeSSLa can express a lot of different properties like real time properties, reasoning about sequential executions orders, calculating and comparing statistics and more. From the beginning on, TeSSLa was built in a way that the generated monitors can be synthesized and executed in hardware (more concrete: FPGAs) such that we are able to still process even huge amounts of data online by exploiting the high parallelism of the hardware.

Furthermore, the goal was that industry software engineers are able to understand and write TeSSLa specifications easily. Hence, TeSSLa is equipped with a strong macro system such that we are able to define a huge standard library on top of the basic functions so software engineers can use TeSSLa. It is also easy to define new functions based on existing ones if needed. Besides using TeSSLa for hardware supported monitoring, it is also feasible to use TeSSLa for creating software monitors, which might reason about objects.

This talk shows the basic motivation for TeSSLa, the basic operators of TeSSLa, application areas and examples.

3.14 E-ACSL, an Executable Behavioural Interface Specification Language for C Programs

Julien Signoles (CEA LIST – Gif-sur-Yvette, FR)

This talk introduces E-ACSL, a behavioral specification language for C programs. It is based on a typed first order logic whose terms are pure C expressions extended with a few specific keywords. Every construct may be executed at runtime. Among others, it provides assertions, contracts, invariants, data dependencies and ghost code. It is powerful enough to express most functional properties of C programs and encode other properties such as LTL properties and information flow policies.

References

1 E-ACSL manual available at http://www.frama-c.com/download/e-acsl/e-acsl.pdf

2 Mickaël Delahaye and Nikolaï Kosmatov and Julien Signoles. Common Specification Language for Static and Dynamic Analysis of C Programs. In Symposium on Applied Computing SAC'13, pages 1230–1235, 2013, ACM.

3.15 LOLA

Hazem Torfah (Universität des Saarlandes, DE)

LOLA is a specification language and stream processing engine for monitoring temporal properties and computing complex statistical measurements. Lola combines the ease-of-use of rule-based specification languages with the expressive power of heavy-weight scripting languages or temporal logics previously needed for the description of complex stateful dependencies. The language comes with two key features: template stream expressions, which allow parameterization with data, and dynamic stream generation, where new properties can be monitored on their own time scale. We give an overview on the development and the current state of our tool in addition to a series of applications.

3.16 Metric First-Order Dynamic Logic and Beyond

Dmitriy Traytel (ETH Zürich, CH)

I present Metric First-Order Dynamic Logic (MFODL), the "supremum" of the specification languages Metric First-Order Temporal Logic (MFOTL) [4] and Metric Dynamic Logic (MDL) [2] used in the MonPoly [1] and Aerial [3] monitoring tools. Moreover, I discuss a few missing features of MFODL, which can be useful in applications: context-free or context-sensitive temporal dependencies, aggregations, and absolute time references.

References

1 D. Basin, F. Klaedtke and E. Zalinescu. The MonPoly monitoring tool. In International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools, 2017.

2 D. Basin, S. Krstić, and D. Traytel. Almost event-rate indepedent monitoring of metric dynamic logic. In S. Lahiri and G. Reger, editors, *RV 2017*, volume 10548 of *LNCS*, pages 85–102. Springer, 2017.

3 D. Basin, S. Krstić, and D. Traytel. Aerial: Almost event-rate indepedent algorithms for monitoring metric regular properties. In International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools, 2017.

4 J. Chomicki. Efficient checking of temporal integrity constraints using bounded history encoding. *ACM Trans. Database Syst.*, 20(2):149–186, 1995.

3.17 Behavioural Type-Based Static Verification Framework for Go

Nobuko Yoshida (Imperial College London, GB)

Main reference Julien Lange , Nicholas Ng , Bernardo Toninho , Nobuko Yoshida: "A Static Verification Framework for Message Passing in Go using Behavioural Types", in Proc. of the 40th Int'l Conf. on Software Engineering (ICSE 2018), to appear, ACM 2018.

I first give an introduction of our group working on session types.

Go is a production-level statically typed programming language whose design features explicit message-passing primitives and lightweight threads, enabling (and encouraging) programmers to develop concurrent systems where components interact through communication more so than by lock-based shared memory concurrency. Go can detect global deadlocks at runtime, but does not provide any compile-time protection against all too common communication mismatches and partial deadlocks.

In this work we present a static verification framework for liveness and safety in Go programs, able to detect communication errors and deadlocks by model checking. Our toolchain infers from a Go program a faithful representation of its communication patterns as behavioural types, where the types are model checked for liveness and safety.

4 Working groups

4.1 How do we integrate trace behaviour with state properties

Wolfgang Ahrendt (Chalmers University of Technology – Göteborg, SE), Stijn de Gouw (Open University – Heerlen, NL), Adrian Francalanza (University of Malta – Msida, MT), Zhiming Liu (Southwest University – Chongqing, CN), and Julien Signoles (CEA LIST – Gif-sur-Yvette, FR)

As a follow-up of the talk *Trace Focused and Data Focused Specification: Complementary, Competing, Combined?*, this group discussed how the community could move towards integrated/coordinated specification and analysis (static or dynamic) of trace *and* data focused aspects. We first made a distinction of specifications and models. The former describe properties a system is supposed to have, while the latter are a vehicle for system design, and finally, development. Following the theme of the seminar, we focused on specifications. Further, we soon converged on the view that the tasks of combining, relating, or even integrating divers specification styles and aspects requires a common semantic foundation.

An archetypal semantic base for both trace and data focused properties could, among others, be sequences $\langle(e_0, v_0), (e_1, v_1), (e_2, v_2), \ldots\rangle$ of pairs (e_i, v_i), where e_i are relevant events, and v_i are valuations, assigning variables (or locations of a heap, or alike) to values. It is obvious how to interpret trace oriented properties on such traces. But also data oriented properties can be interpreted on that basis. For instance, a Hoare triple $\{\phi\}\texttt{m()}\{\psi\}$ could be defined by using method entry and exit events, here $\texttt{m}^{\downarrow}$ and $\texttt{m}^{\uparrow}$, stating that $\texttt{m}^{\downarrow} = e_i$, $\texttt{m}^{\uparrow} = e_j$, and $v_i \models \phi$ implies $v_j \models \psi$.

We arrived at the following (sketch of) a method for relating and combining different formalisms, trace and data oriented.

1. Unify the events relevant for the formalisms.
2. Unify the data (valuations) relevant for the formalisms.
3. Design a semantic domain representing the above, suited to naturally give meaning to the divers properties of interest.

4.2 Specifications that are like implementations

Cyrille Artho (KTH Royal Institute of Technology – Stockholm, SE), Cindy Eisner (IBM – Haifa, IL), Keiko Nakata (SAP Innovation Center – Potsdam, DE), Dejan Nickovic (AIT Austrian Institute of Technology – Wien, AT), and Volker Stolz (West. Norway Univ. of Applied Sciences – Bergen, NO)

To be flexible and expressive with a specification, it is helpful to combine declarative and imperative modes. If the model is rich, this avoids the need for a "bridge specification" that refines an existing model, for example, if parameters are very complex and beyond the expressiveness of the initial modelling language.

4.2.1 Why is this the case?

Many formalisms are too restricted. Temporal logics often feel unnatural to use. Furthermore, logics and finite-state machines cannot count; to cope with that, either formulas become unnecessarily complex, or the formalisms have to be extended with counters. (This is the case, for example, with extended finite-state machines, which are much closer to a possible implementation of a system than what a basic finite-state machine can express.) To be flexible and expressive with a specification, it is helpful to combine declarative and imperative modes. If the model is rich, this avoids the need for a "bridge specification" that refines an existing model, for example, if parameters are very complex and beyond the expressiveness of the initial modelling language

4.2.2 Trade-offs

The table below shows some of the trade-offs with the choice of the specification language:

	Simple, declarative specification	**Complex (imperative?) specification**
Specific burden (human)	High if specification becomes complex due to limitations	May be lower; feels "natural" to developers
Level of abstraction	High	Low
Semantics	Well-defined but perhaps difficult for a human	Loosely-defined but perhaps clearer to developers
Analysis burden (tool)	Low	High
Link to implementation	Remote	Close
Refining specification to implementation	Difficult	More direct
Integration with existing tools	Difficult	More straightforward

A specification that is very low-level may make it difficult to express complex data types or properties. Conversely, though, the high level of abstraction required by typical specification languages is also a desirable trait. The specification should be significantly simpler than the implementation and not just mirror it. Otherwise, the level of abstraction leads to the same type of thinking, and hence the same bugs in both the specification and the implementation; the distinction of "what" (specification) vs. "how" (implementation) becomes blurred. In many cases, a "rich" specification language may be enticing because it offers the same features like a programming language. Conversely, certain features are not available in a programming language. For example, it is not possible (without much extra code) to quantify over all instances on the heap in a programming language, but some specification languages may allow that.

On the practical side, using the implementation language (or a subset of it) for specifications eliminates the need to learn about a new language or platform. This choice makes tool chain integration straightforward and allows the same tools to be used for static and dynamic analysis. It also allows properties to refer directly to the implementation. Refinements are easier to reason about if the language is the same. The semantics of the language may be clearer to developers, making this a more "practical" choice.

4.2.3 Solutions

Domain-specific languages seem to be a great way to hide low-level aspects of “classical” specification languages while restricting the complexity of the specification. For example, PSL has been very successful because it is on a higher level of abstraction than LTL, allows for different styles of specification, and has several ways to make specifications more succinct. However, building better specification languages is difficult; it takes experience to extract common patterns and “package” them in a nice way. Visual specification languages are popular for software engineering, but their semantics are often not well-defined and not executable. A mathematical or textual specification language should still be reasonably close in appearance to a programming language. This explains why very mathematical notations like LTL and Z are not so popular with engineers.

The choice of the language depends on the problem to be solved: If problems to be specified are close to the implementation platform, such as in run-time verification, programming languages can be suitable. For something that should be platform-agnostic and abstract, other choices may be better.

If a programming language is used for verification, the language features used should be limited, and side effects in statements must be avoided. It is good to forgo completeness (being able to express any property): some abstraction is good and should be enforced at the level of the specification language. In the extreme case, a reference implementation is an executable specification, but properties may not be explicit. In this case, the implementation part should be separate from the monitor that checks results.

A good specification toolkit includes validation tools (vacuity, consistency, realizability checking; simulation; visualization). High-level languages may be easier to validate in principle, but programming languages often have good tools for this purpose, too.

Most importantly, we need to support a good V and V process, not just to provide tools. The given specification language should be used in the spirit of proving a formal model, not an implementation.

4.3 Property Specification Patterns for Runtime Verification

Domenico Bianculli (University of Luxembourg, LU), Borzoo Bonakdarpour (McMaster University – Hamilton, CA), Bernd Finkbeiner (Universität des Saarlandes, DE), Gordon Pace (University of Malta – Msida, MT), Giles Reger (University of Manchester, GB), Kristin Yvonne Rozier (Iowa State University, US), Gerardo Schneider (Chalmers University of Technology – Göteborg, SE), Dmitriy Traytel (ETH Zürich, CH), and Nobuko Yoshida (Imperial College London, GB)

Property specification patterns have been proposed as a means to express recurring properties in a generalized form, to aid engineers writing system requirements precisely and map them to formalisms like temporal logics. Starting from the seminal work by Dwyer et al. [4], several systems (i.e., catalogues) of property specification patterns have been proposed in the literature [1, 3, 2, 6, 5, 7]. This working group discussed the use of property specification patterns for writing specifications in the context of Runtime Verification (RV).

4.3.1 Why Specification Patterns in RV?

A first discussion point focused on the identification of the main reasons for using property specification patterns in RV:

abstraction: to raise the level of abstraction when writing specifications;

conciseness: to express complex specifications in a concise way;

reusability: to reuse blocks of specifications in different systems and programming languages;

compositionality: to be able to compose and intertwine different aspects of a specification;

extensibility: to provide a system that can be extended in a coherent way for different application domains;

automation: to enable the automated transformation of a high-level, abstract specification into a low-level, concrete specification, by supporting different formalisms, technologies, and instrumentation strategies.

4.3.2 Towards a System of Property Specification Patterns for RV.

The second part of the discussion pinpointed the main attributes that would characterize a system of property specification patterns tailored for RV applications. Such a system shall:

- support the three main "planes" of RV (i.e., time, data, and algorithms/behaviors) as first-class entities;
- enable the definition of multi-level patterns;
- provide pattern combinators;
- be built based on existing pattern systems.

Furthermore, several complementary extensions could be envisioned for this system:

- anti-patterns, to give guidance about which specifications are better avoided;
- design patterns for building observers, verifiers, and enforcers;
- an "interface language" to describe the capabilities of observers, verifiers, and enforcers with respect to certain patterns;
- transformational patterns, for transforming specifications into various target formalisms.

Finally, the working group touched upon the main design choices that the designers of such a system would face:

- How deep should the system be embedded in a target formalism?
- Should the system account only for functional specifications or also non-functional ones?

References

1 M. Autili, L. Grunske, M. Lumpe, P. Pelliccione, and A. Tang. Aligning qualitative, real-time, and probabilistic property specification patterns using a structured english grammar. *IEEE Transactions on Software Engineering*, 41(7):620–638, July 2015.

2 P. Bellini, P. Nesi, and D. Rogai. Expressing and organizing real-time specification patterns via temporal logics. *J. Syst. Softw.*, 82(2):183–196, February 2009.

3 Domenico Bianculli, Carlo Ghezzi, Cesare Pautasso, and Patrick Senti. Specification patterns from research to industry: a case study in service-based applications. In *ICSE 2012: Proceedings of the 34th international conference on Software engineering*, pages 968–976. IEEE Computer Society, 2012.

4 Matthew B. Dwyer, George S. Avrunin, and James C. Corbett. Patterns in property specifications for finite-state verification. In *ICSE '99: Proceedings of the 21st international conference on Software engineering*, pages 411–420. IEEE Computer Society Press, 1999.

5 Volker Gruhn and Ralf Laue. Patterns for timed property specifications. *Electron. Notes Theor. Comput. Sci.*, 153(2):117–133, 2006.

6 Lars Grunske. Specification patterns for probabilistic quality properties. In *ICSE 2008: Proceedings of the 30th international conference on Software engineering*, pages 31–40, New York, NY, USA, 2008. ACM.

7 Sascha Konrad and Betty H. C. Cheng. Real-time specification patterns. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, pages 372–381, New York, NY, USA, 2005. ACM.

4.4 Exploring the tradeoffs between Declarative and Operational Specification

Adrian Francalanza (University of Malta – Msida, MT), Wolfgang Ahrendt (Chalmers University of Technology – Göteborg, SE), Cindy Eisner (IBM – Haifa, IL), Zhiming Liu (Southwest University – Chongqing, CN), and Gordon Pace (University of Malta – Msida, MT)

This working group explored the tradeoffs between declarative and operational/algorithm approaches to specification. We begin with defining these two terms:

- By "Declarative" (D) what we understand is "What" is expected to hold. A good example here would be regular expressions, defined (denotationally) as the sets of strings that they represent.
- By "Algorithmic" (A) what we understand is "How" we intend to check what is expected to hold. A good example here would be DFSAs: they may be used to denote the sets of strings that they accept, via the operational procedure of processing a string from the start state to the final state.

Another analogy could be the one between state-based vs action-based specification, though it is not clear that it fits exactly. In practice, this clear delineation is often blurred.

Next we consider what criteria one may want to consider when assessing/evaluating/deciding whether to prefer one style over the other.

Readability and Understandable. In general (D) tend to be more concise, hence potentially more readable. This, of course, need not be the case and is also very subjective. Certain (A) approaches have a graphical representation which improves substantially the understandability (e.g. automata). Various anecdotal evidence was brought forward. Whereas it was certainly the case for small specifications, it was unclear whether these advantages would scale for larger/real-world properties. Syntactic sugaring is also very useful in these cases, and (D) may be more amenable to this due to the compositional operators. These points were also linked to the need to have Domain-Specific language adaptations of these formalisms where, perhaps, only a subset of the expressive power may be needed.

Maintainability/adaptability. In general (D) are more algebraic by nature and come equipped with operations to compose specifications. This tends to affect maintainability/adaptability/compositionality/decompositionality.

Specification procurement. To give a bit of context, there seems to be a general aversion of engineers towards using specifications altogether. One criteria could thus be to lean towards the form of specification that facilitate best specification procurement. To this end, techniques such as visualisations and connections to controlled natural languages are definitely a plus. Neither form, (D) or (A), offered a particular advantage over the other. For instance the box modality $[\alpha]\psi$ can be expressed as *whenever α then ψ must hold.* A case can be made that logical operators are not as intuitive to the“general public” e.g. by *or* they often understand *exclusive or*, believe that *implication* is associated with causality etc. Another aspect that was briefly touched upon was that of integrating these things into the workflow of a company, having good error reporting etc.

4.5 Event Stream Processing

Sylvain Hallé (University of Quebec at Chicoutimi, CA), Martin Leucker (Universität Lübeck, DE), Nicolas Rapin (CEA – Gif sur Yvette, FR), César Sánchez (IMDEA Software – Madrid, ES), Torben Scheffel (Universität Lübeck, DE), and Hazem Torfah (Universität des Saarlandes, DE)

This working group concentrated on the topic of behavioural specification through Event Stream Processing (ESP). All the participants in this working group were involved in the design and development of one of several ESP tools: ARTiMon [5], BeepBeep [4], LOLA [1], and TeSSLa [3]. The discussion revolved around a few broad questions, which are summarized in the following.

4.5.1 Motivation

A first discussion topic focused on the reasons the various ESP tools have been developed in the first place. This point is relevant, given that most of us come from the Runtime Verification (RV) community, where specification languages are generally based on logic or automata. For most of us, the shift to ESP came “out of necessity” –that is, some use cases we were faced with were difficult or flatly impossible to model using traditional logic or automata.

Case in point, one participant recalled an experience with members of the industry, who were shown specifications expressed in temporal logic in a first meeting, and specifications expressed as stream equations during a subsequent meeting. The overall reception to the stream equations was much more positive than for the temporal logic formulæ (“why didn’t you show this in the first place?” was the reaction of one of the attendees). For some classes of problems, and for some audiences, the use of streams to model a specification may prove a better fit than existing logics and automata-based notations.

It shall be noted that, although the use of event streams in a verification or testing context is relatively recent, it can be seen as a generalization of much older concepts. For example, *temporal testers* have been introduced for the verification of Linear Temporal Logic specifications more than twenty years ago [2]. By using the same core principles (composition, incremental update upon each event), contemporary ESP tools extend temporal testers beyond Boolean streams and temporal logic, and generalize them to allow the computation of aggregations over numerical values and many other functions.

4.5.2 Declarative or Imperative?

A second point of discussion is the classification of event stream languages as *declarative* or *imperative*. For example, the main mode of operation of BeepBeep is to programmatically instantiate and connect "boxes" that each perform a simple computation, which is close to an imperative language. However, one can also see an event processing specification as a set of equations defining relationships between input and output streams; this is especially apparent in the input languages of ARTiMon, LOLA and TeSSLa. It can be argued that these equations define constraints on the inputs and the outputs that must be satisfied, but do not explicitly describe *how* to compute a satisfying output for a given input (or if such an output even exists). These traits would classify event stream specifications as declarative.

4.5.3 Relative Expressiveness

Although ESP tools share many similarities (the possibility to compute aggregation functions over time windows, for example), their expressiveness is not identical. One particular point where they differ is in their interpretation of time. Some systems assume synchronous streams where computation occurs in discrete steps, while others accept input streams that produce events at their own rate. Again, the particular implementation of time in each tool has been motivated by concrete use cases that the system had to handle.

4.5.4 Should there be fewer tools?

Another question that was discussed is whether it would be desirable to have fewer stream languages. The general consensus among panel members was that, since their expressiveness is not identical (see above), each tool fulfills a different need. Moreover, having multiple tools maintains a healthy level of competition and drives innovation.

It was also observed that, in the past fifteen years, we have witnessed the introduction of a large number of new programming languages. In many cases, new languages have been invented, because "starting from scratch" was easier than adapting an existing (and only partially appropriate) solution; a similar argument can be made about stream languages. To sum it up, if the existence of multiple programming languages is generally accepted and is not seen as abnormal, why would that be different for specification languages?

4.5.5 The future

As a next step for the short- and medium-term, it was suggested that authors of ESP tools should give themselves a standardized vocabulary to define the features of each specification language and each system. This vocabulary, in turn, would make it possible to compare the various solutions.

References

1 B. D'Angelo, S. Sankaranarayanan, C. Sánchez, W. Robinson, B. Finkbeiner, H.B. Sipma, S. Mehrotra, Z. Manna. (2005). LOLA: Runtime Monitoring of Synchronous Systems. In *12th International Symposium on Temporal Representation and Reasoning, TIME 2005*, IEEE Computer Society, 166–174.

2 Y. Kesten, A. Pnueli, and L. Raviv. (1998). Algorithmic verification of linear temporal logic specifications. In *Proc. 25th Int. Colloq. Aut. Lang. Prog.*. Springer: Lecture Notes in Computer Science, vol. 1443, 1–16.

3 M. Leucker, C. Sánchez, T. Scheffel, M. Schmitz, A. Schramm. (2018). TeSSLa: Runtime Verification of Non-synchronized Real-Time Streams. To appear on *ACM Symposium on Applied Computing, SAC 2018.*

4 S. Hallé. (2016). When RV Meets CEP. In Y. Falcone, C. Sánchez (eds), *Runtime Verification, RV 2016.* Springer: Lecture Notes in Computer Science, vol. 10012, 68–91.

5 N. Rapin. (2016) Reactive Property Monitoring of Hybrid Systems with Aggregation. In Y. Falcone, C. Sánchez (eds), *Runtime Verification, RV 2016.* Springer: Lecture Notes in Computer Science, vol. 10012, 447–453.

4.6 A shared challenge – NASA's Core Flight System

Volker Stolz (West. Norway Univ. of Applied Sciences – Bergen, NO), Borzoo Bonakdarpour (McMaster University – Hamilton, CA), Martin Leucker (Universität Lübeck, DE), Nicolas Rapin (CEA – Gif sur Yvette, FR), Kristin Yvonne Rozier (Iowa State University, US), Julien Signoles (CEA LIST – Gif-sur-Yvette, FR), Hazem Torfah (Universität des Saarlandes, DE), and Nobuko Yoshida (Imperial College London, GB)

4.6.1 Introduction

This working group focused on identifying a single, unified challenge which could be used by the RV community to show-case their tools and methods. In other communities, there have been case studies that are still around and referred to frequently, such as the Steam Boiler Control [3], the Mondex case study grand challenge [4], and the Common Component Modelling Example CoCoME [5]. We intend to kick-start a similar "future-proof" arena for joint activity through further discussion among the seminar participants. The following summarises the discussion within the working group.

In particular, we were interested in a topic that not only serves those goals (and has possibilities for most community members to contribute), but also is current and and engaging. Unmanned (aerial) vehicles (UAVs/drones) seem to fit this purpose very well: it is a topic that easily engages students, has many "moveable parts" that make it easy and interesting to instrument (sensor readings and engine data), and as a platform UAVs are easily extensible with additional sensors to tailor them to particular projects.

Although consumer-grade products usually come with their own control software, operating systems are easily interchangeable, and are frequently Android-based. To fulfil the requirement of a single unified system, NASA's Core Flight System (CFS, https://cfs.gsfc.nasa.gov/), can serve as a middleware tying together the various functionalities. Furthermore, prominent members of the community are, or have been, affiliated with NASA which developed and maintains the stack, and can thus serve as contact point for others seeking help or guidance with the system. In the following, we discuss some of the possibilities and challenges that those devices together with the CFS present as a platform for a shared RV challenge.

4.6.2 NASA's Core Flight System

The system covers the full range of a cyber-physical system: on top of a Linux kernel, a modular architecture provides a middleware for communication using publish/subscribe between the different components. Specific applications sit on top of this middleware. The entire system is open-source and available from NASA's open-source repositories. It also includes a simulator. The Linux-based open-source architecture should make it easy to interface with additional tools, such as the instrumentations required for applied runtime verification.

A wide range of hardware is already supported, and the software has already been used on off-the-shelf UAVs. For new hardware, some configuration may be required, and contacts with NASA confirm that they are willing to provide some remote support, and have already made steps to incorporate the latest updates into the publicly available repositories.

Even within constrained budgets, consumer-grade drones or a computationally more powerful octocopter can be purchased for below a 1000 USD, allowing even smaller groups to participate after a small up-front investment. Within this budget, a subgroup of the community could also create a reference architecture with a well-defined set of sensors and motors. For the lower-end hardware, CFS does not directly run on the hardware itself, but on a separate computer that uses a communications link (e.g. wifi) to remote-control the vehicle.

4.6.3 Requirements from the RV Perspective

For the RV community, it is important that as many of the diverse approaches can be served by this platform. Most importantly, this includes the division between online- and offline processing, runtime reflection, and solutions requiring specific software. In general, we remark that the open-source nature of the entire platform gives ample opportunity to customize it.

For online runtime verification, the publish/subscribe communications-style of the middleware makes it easy to receive events from subsystems. Runtime reflection, where the behaviour of the system is actively influenced, can be equally easily achieved if the component to be influenced reacts to messages on the middleware bus, but otherwise may of course require some modification, or the design of new components.

Offline runtime verification, where event data is analysed through RV techniques (evaluating for example temporal logic properties), can be easily achieved by recording event data from the system, e.g. by logging messages from the middleware. Even if no hardware is available, trace data can be obtained from the simulator. Even if a community member lacks the technical expertise to run the system themselves, traces can be generated and published (e.g. in the upcoming trace data repository of the EU COST Action IC1402 "ARVI – Runtime Verification Beyond Monitoring", see [1]) by other members.

Several seminar participants work with particular implementations of runtime verification technology, e.g. those that are implemented on top of the Java Virtual Machine. Naturally, those approaches do not easily carry over into the world of embedded systems, where often specific operating systems requirements do not allow running full user-land applications, or memory or CPU are too constrained to enable such approaches. The suggested platform in principle offers a full operating system, and is only constrained by the underlying hardware. This allows groups with higher computational requirements to move to more powerful hardware, or use the simulator.

4.6.4 Relevant Properties

This working group also took a precursory look at relevant runtime verification-related properties that may be investigated on such a system. In particular, the general nature of the middleware should offer plenty of opportunity to look for relevant events. Although of course one could potentially devise application specific properties, and existing set of specifications for the platform would be an interesting possibility of directly applying RV techniques in this domain.

We easily found existing specifications in particular for the LADEE mission such as “while in a flight mode, with active control disabled, a particular parameter shall also be disabled” (Karen Gundy-Burlet, SPIN 2014 keynote). The same mission has also already been utilised by Havelund as an application scenario for TraceContract [2].

Furthermore, we also found MatLab/Simulink models in the repositories which are of direct interest to runtime verification.

4.6.5 Conclusion

In conclusion, we are confident that such a system gives ample opportunity to either find, or come up with our own, properties that our current tools can then monitor, and that the majority of the community members hopefully have the resources and expertise to participate.

References

1 COEMS – Open traces from industry. S. Jaksic, M. Leucker, D. Li, and V. Stolz. In: RV-CuBES workshop, Kalpa, 2017.

2 Checking Flight Rules with TraceContract: Application of a Scala DSL for Trace Analysis. H. Barringer, K. Havelund, E. Kurklu, and R. Morris. In: Scala Days 2011

3 Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control. J.-R. Abrial, E. Börger, and H. Langmaack. In: LNCS 1165, Springer, 1996.

4 First steps in the verified software grand challenge. J. Woodcock. In: Computer, 39(10):57–64, October 2006.

5 The Common Component Modeling Example: Comparing Software Component Models. A. Rausch, R. H. Reussner, R. Mirandola, F. Plasil (eds.). In: LNCS 5153, Springer, 2008

4.7 Data Quantification in Temporal Specification Languages

Dmitriy Traytel (ETH Zürich, CH), Domenico Bianculli (University of Luxembourg, LU), and Giles Reger (University of Manchester, GB)

In this working group, we tried to collect and characterize the different kinds of quantification or parametrization that can be encountered in temporal specification languages used in the runtime verification community. Our selection, albeit far from being comprehensive, shows that the main semantic difference is the domain of quantification. We have identified five different groups.

4.7.1 Standard First-Order Quantification

An early approach taken by Emerson’s first-order linear temporal logic (FOLTL) [13] is to add standard first-order logic quantifiers to LTL. Thereby, the quantifiers range over a fixed domain, which is independent of the trace (sequence of structures). Chomicki’s real-time extension of FOLTL, called metric first-order temporal logic (MFOTL) [9] follows this approach. The MonPoly monitoring tool [4] demonstrates how such quantifiers can be handled algorithmically. The new Dejavu tool [16] (and logic) quantifies over fixed (infinite) domains.

4.7.2 Quantification over the Active Domain

A different approach, inspired by the database community, is to quantify over the active domain, i.e., values that occur in the trace, rather than a fixed, separately given domain. For certain classes of properties, e.g., where all quantifiers are bounded by atomic propositions as in $\forall x.p(x) \to \phi$ or $\exists x.p(x) \wedge \phi$, active domain quantification coincides with the standard first-order quantification.

Several specification languages favor the active domain quantification, as it appears to be algorithmically more tractable. They differ in their definition of what the *active domain* in a trace is. The traditional database definition as *all values contained in the trace*, used in LTL-FO [12], is hard to realize in the online setting, where the trace is an infinite stream of events. An adaptation of the active domain to *all previously seen values* would fit this setting better. However, the most widespread interpretation is to restrict the quantification to *values seen at the current time-point*. For example, the languages LTL^{FO} [7], LTL-FO^{+} [15], and Parametrized LTL [20] use this semantics. Additionally, DejaVu [16] in its current form can quantify over "seen" values in the past. There are in fact two sets of quantifiers: `forall` and `exists` for quantifying over seen values, and `Forall` and `Exists` for quantifying over all values in the fixed (possibly infinite) domain.

4.7.3 Freeze Quantification

Freeze quantification is a further refinement of the quantification over the current time-point approach. The usage of registers restricts the quantification to be a singleton: the only value that populates the register at a given time-point. Timed propositional temporal logic (TPTL) [1] uses such quantifiers to extend LTL with real-time constraints. Here, we are interested in quantification over the data dimension rather than the time dimension, as used in Freeze LTL [11] and its extensions [10]. A recent extension of MTL with freeze quantification over data $\text{MTL}^{\downarrow}$ [5] was used as the specification language when online monitoring our-of-order traces.

4.7.4 Templates and Parametric Trace Slicing

Some approaches avoid explicit quantification in their formalisms. Yet, they allow parametric specifications, which are handled by decomposing traces containing data into propositional ones. This approach is known as parametric trace slicing [8, 19, 21], which is at the core of the JavaMOP system [18] and in quantified event automata QEA [2].

More recently, the stream-based specification language LOLA [14] introduced parametrization in terms of template specifications. Semantically, templates behave similarly to parametric trace slicing, but the precise connections are yet to be explored.

4.7.5 Quantitative Quantifiers

Finally, some data quantifiers in addition to binding a variable also perform an arithmetic operation (be it filtering, grouping, or aggregation) on the quantified values (be them data or the number of satisfied instances). Example languages in this space are LTL^{FO} extended with counting quantifiers [6], LTL_4-C [17] with its probabilistic quantifiers, and the extension of MFOTL with aggregations [3].

References

1 R. Alur and T. A. Henzinger. A really temporal logic. *J. ACM*, 41(1):181–204, 1994.

2 H. Barringer, Y. Falcone, K. Havelund, G. Reger, and D. E. Rydeheard. Quantified event automata: Towards expressive and efficient runtime monitors. In D. Giannakopoulou and D. Méry, editors, *FM 2012*, volume 7436 of *LNCS*, pages 68–84. Springer, 2012.

3 D. A. Basin, F. Klaedtke, S. Marinovic, and E. Zalinescu. Monitoring of temporal first-order properties with aggregations. *Formal Methods in System Design*, 46(3):262–285, 2015.

4 D. A. Basin, F. Klaedtke, S. Müller, and E. Zalinescu. Monitoring metric first-order temporal properties. *J. ACM*, 62(2):15:1–15:45, 2015.

5 D. A. Basin, F. Klaedtke, and E. Zalinescu. Runtime verification of temporal properties over out-of-order data streams. In R. Majumdar and V. Kuncak, editors, *CAV 2017*, volume 10426 of *LNCS*, pages 356–376. Springer, 2017.

6 A. Bauer, R. Goré, and A. Tiu. A first-order policy language for history-based transaction monitoring. In M. Leucker and C. Morgan, editors, *ICTAC 2009*, volume 5684 of *LNCS*, pages 96–111. Springer, 2009.

7 A. Bauer, J. Küster, and G. Vegliach. The ins and outs of first-order runtime verification. *Formal Methods in System Design*, 46(3):286–316, 2015.

8 F. Chen and G. Rosu. Parametric trace slicing and monitoring. In S. Kowalewski and A. Philippou, editors, *TACAS 2009*, volume 5505 of *LNCS*, pages 246–261. Springer, 2009.

9 J. Chomicki. Efficient checking of temporal integrity constraints using bounded history encoding. *ACM Trans. Database Syst.*, 20(2):149–186, 1995.

10 N. Decker and D. Thoma. On freeze LTL with ordered attributes. In B. Jacobs and C. Löding, editors, *FoSSaCS 2016*, volume 9634 of *LNCS*, pages 269–284. Springer, 2016.

11 S. Demri, R. Lazic, and D. Nowak. On the freeze quantifier in constraint LTL: decidability and complexity. *Inf. Comput.*, 205(1):2–24, 2007.

12 A. Deutsch, L. Sui, V. Vianu, and D. Zhou. Verification of communicating data-driven web services. In S. Vansummeren, editor, *PODS 2006*, pages 90–99. ACM, 2006.

13 E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 995–1072. 1990.

14 P. Faymonville, B. Finkbeiner, S. Schirmer, and H. Torfah. A stream-based specification language for network monitoring. In Y. Falcone and C. Sánchez, editors, *RV 2016*, volume 10012 of *LNCS*, pages 152–168. Springer, 2016.

15 S. Hallé and R. Villemaire. Runtime monitoring of message-based workflows with data. In *EDOC 2008*, pages 63–72. IEEE Computer Society, 2008.

16 First Order Temporal Logic Monitoring with BDDs. K. Havelund, D. Peled, and D. Ulus 17th Conference on Formal Methods in Computer-Aided Design (FMCAD 2017), 2-6 October, 2017, Vienna, Austria. IEEE.

17 R. Medhat, B. Bonakdarpour, S. Fischmeister, and Y. Joshi. Accelerated runtime verification of LTL specifications with counting semantics. In Y. Falcone and C. Sánchez, editors, *RV 2016*, volume 10012 of *LNCS*, pages 251–267. Springer, 2016.

18 P. O. Meredith, D. Jin, D. Griffith, F. Chen, and G. Rosu. An overview of the MOP runtime verification framework. *STTT*, 14(3):249–289, 2012.

19 G. Reger and D. E. Rydeheard. From first-order temporal logic to parametric trace slicing. In E. Bartocci and R. Majumdar, editors, *RV 2015*, volume 9333 of *LNCS*, pages 216–232. Springer, 2015.

20 V. Stolz. Temporal assertions with parametrized propositions. *J. Log. Comput.*, 20(3):743–757, 2010.

21 K. Havelund and G. Reger and E. Zalinescu and D. Thoma Monitoring Events that Carry Data, In E. Bartocci and Y. Falcone, editors, Lectures on Runtime Verification – Introductory and Advanced Topics, volume 10457 of *LNCS*, pages 60–97, Springer, 2018

Participants

- Wolfgang Ahrendt
Chalmers University of Technology – Göteborg, SE
- Cyrille Artho
KTH Royal Institute of Technology – Stockholm, SE
- Domenico Bianculli
University of Luxembourg, LU
- Borzoo Bonakdarpour
McMaster University – Hamilton, CA
- Stijn de Gouw
Open University – Heerlen, NL
- Cindy Eisner
IBM – Haifa, IL
- Bernd Finkbeiner
Universität des Saarlandes, DE
- Adrian Francalanza
University of Malta – Msida, MT
- Sylvain Hallé
University of Quebec at Chicoutimi, CA
- Martin Leucker
Universität Lübeck, DE
- Zhiming Liu
Southwest University – Chongqing, CN
- Keiko Nakata
SAP Innovation Center – Potsdam, DE
- Dejan Nickovic
AIT Austrian Institute of Technology – Wien, AT
- Gordon Pace
University of Malta – Msida, MT
- Nicolas Rapin
CEA – Gif sur Yvette, FR
- Giles Reger
University of Manchester, GB
- Kristin Yvonne Rozier
Iowa State University, US
- César Sánchez
IMDEA Software – Madrid, ES
- Torben Scheffel
Universität Lübeck, DE
- Gerardo Schneider
Chalmers University of Technology – Göteborg, SE
- Julien Signoles
CEA LIST – Gif-sur-Yvette, FR
- Volker Stolz
West. Norway Univ. of Applied Sciences – Bergen, NO
- Hazem Torfah
Universität des Saarlandes, DE
- Dmitriy Traytel
ETH Zürich, CH
- Nobuko Yoshida
Imperial College London, GB

Report from Dagstuhl Seminar 17471

Artificial and Computational Intelligence in Games: AI-Driven Game Design

Edited by

Pieter Spronck[1], Elisabeth André[2], Michael Cook[3], and Mike Preuß[4]

1 **Tilburg University, NL,** p.spronck@gmail.com
2 **Universität Augsburg, DE,** andre@informatik.uni-augsburg.de
3 **University of London, GB,** mike@gamesbyangelina.org
4 **Universität Münster, DE,** mike.preuss@wi.uni-muenster.de

Abstract

With the dramatic growth of the game industry over the past decade, its rapid inclusion in many sectors of today's society, and the increased complexity of games, game development has reached a point where it is no longer humanly possible to use only manual techniques to create games. Large parts of games need to be designed, built, and tested automatically. In recent years, researchers have delved into artificial intelligence techniques to support, assist, and even drive game development. Such techniques include procedural content generation, automated narration, player modelling and adaptation, and automated game design. This research is still very young, but already the games industry is taking small steps to integrate some of these techniques in their approach to design.

The goal of this seminar was to bring together researchers and industry representatives who work at the forefront of artificial intelligence (AI) and computational intelligence (CI) in games, to (1) explore and extend the possibilities of AI-driven game design, (2) to identify the most viable applications of AI-driven game design in the game industry, and (3) to investigate new approaches to AI-driven game design. To this end, the seminar included a wide range of researchers and developers, including specialists in AI/CI for abstract games, commercial video games, and serious games. Thus, it fostered a better understanding of and unified vision on AI-driven game design, using input from both scientists as well as AI specialists from industry.

Seminar November 19–24, 2017 – http://www.dagstuhl.de/17471
1998 ACM Subject Classification I.2.1 Artificial Intelligence - Games
Keywords and phrases dynamical systems, entertainment modeling, game design, multi-agent systems, serious games
Digital Object Identifier 10.4230/DagRep.7.11.86

1 Executive Summary

Pieter Spronck
Elisabeth André
Michael Cook
Mike Preuß

The video game industry has been developing rapidly in the past decade. Whereas ten years ago video games were almost exclusively aimed at entertainment, nowadays they are used

Artificial and Computational Intelligence in Games: AI-Driven Game Design, *Dagstuhl Reports*, Vol. 7, Issue 11, pp. 86–129
Editors: Pieter Spronck, Elisabeth André, Michael Cook, and Mike Preuß
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

in a variety of places in everyday life. All kinds of organizations now use video games for simulation and training. Educational institutes use video games to enrich and replace parts of courses. Governmental and health care agencies use video games to educate people and stimulate them to lead more productive lives. On top of that, the entertainment-focused games industry continues to grow and is a major industry both culturally and financially.

Two parallel developments can be observed in the games industry. On the one hand, the high-profile entertainment games ("triple-A games") see a steady increase of time and financial resources invested in their development, to keep up with technological advances and to be able to compete in a tough market. On the other hand, the number of smaller, special-purpose games in development (including so-called "serious games") increases dramatically too, in particular in research, training, and education. Moreover, as the pervasiveness of video games increases, so does the the number of people involved in creating them. The job of creating games is no longer limited to specialist programmers and artists. Instead, those who need to use the games become heavily involved in their creation.

The serious-games domain poses additional challenges to game development beyond all the challenges already posed by games for entertainment, namely the need for a strong relation with the "real world". Serious games often have a purpose in training, which entails that the game worlds must be a realistic depiction of the actual environment in which user functions, in particular where "behaviors" are concerned. The big data revolution means that huge quantities of data about the real world are becoming available along with the means of processing them, which may offer the possibility to automatically construct games on the basis of such data. This is a particularly enticing notion, given the financial constraints for constructing serious games, which means that professional content producers might not be available.

Furthermore, new computer games are expected to much better incorporate the different needs of a wide variety of customers, to provide more alternative modes, solution paths, incentives, emotional states, and difficulty levels. Game design, and especially balancing, must take this into account. However, this increases the complexity of design and production considerably, such that AI-based tools that can assist the human developer or even partly automatize processes are more desired than ever.

Summarizing, we note the following four trends in modern game development:

1. Technological advances have lead to an increased challenge in developing modern video games, even for expert game developers
2. There is an increased need for non-experts to be able to design and develop games
3. There is an increased need for realism in the virtual world behaviors, in particular in the area of serious games
4. A greater variety of players and a better availability of data about players leads to the need for more variable and better customizable games, which require a more complex development process.

A solution for each of these issues can be found in the application of artificial-intelligence (AI) techniques to drive the design and development of games. From the perspective of AI-driven game design, AI supports or even takes over the role of the human game developer in creating particular elements of a game, and even complete new games.

While the game industry tends to use a small selection of well-known algorithms to generate elements of game worlds (in particular where graphics and animation are concerned), the use of AI to create new environments, new behaviors of virtual characters, new narratives, new game rules, or even new gameplay mechanisms is at present limited to a very small number of researchers. We see it, however, as the main direction in which innovation in game design and development can be found.

AI-driven game design sees applications in the design of virtual worlds, virtual characters, narratives, and game mechanics. Moreover, it can be used to assist in the human design process, and to adapt games automatically after publication. Finally, it can support the automated analysis of generated game elements. Each of these topics is a research domain by itself, which requires an interdisciplinary approach which borrows from computer science, psychology, cognitive science, and even the creative arts. A common ground is found in artificial intelligence techniques, in particular machine learning.

For this seminar, we brought together computer scientists and creative experts with the common goals of gaining a deeper understanding of various aspects of games, and of further improving games, in particular by using AI-techniques used to generate games and game elements. The goal was to look beyond what is currently possible and in use, and take steps towards the future of AI-driven game development. Besides theoretical discussions, part of the seminar was spent on trying to achieve first practical results.

Reports on the discussions and results achieved are found on the following pages. All in all, the organizers and participants deemed the seminar a great success, and are eager to continue into some of the directions that were focused on during the week at Schloss Dagstuhl.

2 Table of Contents

Executive Summary

Pieter Spronck, Elisabeth André, Michael Cook, and Mike Preuß 86

Overview of Talks

Some Observations on Automated Strategy Game Design
Cameron Browne 91

Working groups

Game Search Space Design and Representation
Dan Ashlock, Cameron Browne, Simon Colton, Amy K. Hoover, Jialin Liu, Simon M. Lucas, Mark J. Nelson, Diego Perez Liebana, Sebastian Risi, Jacob Schrum, Adam M. Smith, Julian Togelius, and Vanessa Volz 93

Blue Skies Research Questions in AI-driven Game Design
Peter I. Cowling, Elisabeth André, Rafael Bidarra, Pier Luca Lanzi, Mike Preuß, Spyridon Samothrakis, Pieter Spronck, Shoshannah Tekofsky, and Georgios N. Yannakakis 95

Some Industry Interests in AI-driven Game Design
Peter I. Cowling, Sander C.J. Bakkes, Pier Luca Lanzi, Adam M. Smith, Pieter Spronck, and Shoshannah Tekofsky 97

Machina Ex Machina (Part 1)
Reynald Francois, Matthew J. Guzdial, Marc Erich Latoschik, Antonios Liapis, Alexander Nareyek, Emily Short, and David Thue 100

Machina Ex Machina (Part 2)
Reynald Francois, Joshua Allen McCoy, Alexander Nareyek, and G. Michael Youngblood 100

What Is Machine Learning/Deep Learning?
Matthew J. Guzdial, Joshua Allen McCoy, Emily Short, Michael Treanor, R. Michael Young, and Jichen Zhu 101

AI-assisted Board Game Play
Antonios Liapis, Michael Cook, Steve Dahlskog, Mirjam P. Eladhari, Matthew J. Guzdial, Emily Short, Gillian Smith, Anne Sullivan, and Tommy Thompson 104

Mixed-media Game AI
Antonios Liapis, Elisabeth André, Sander C.J. Bakkes, Rafael Bidarra, Steve Dahlskog, Mirjam P. Eladhari, Ana Paiva, Mike Preuß, Gillian Smith, Anne Sullivan, Tommy Thompson, David Thue, Georgios N. Yannakakis, and R. Michael Young . 105

Emergence
Jacob Schrum, Cameron Browne, Mark J. Nelson, Ana Paiva, Julian Togelius, and Georgios N. Yannakakis 108

Human-assisted Creation of Content Within Games
Jacob Schrum, Pier Luca Lanzi, Alexander Nareyek, and Pieter Spronck 111

Play
Jacob Schrum, Cameron Browne, Amy K. Hoover, Sebastian Risi, David Thue, Julian Togelius, and Jichen Zhu 112

AI As Reflective Practice
Gillian Smith, Mirjam P. Eladhari, Matthew J. Guzdial, Emily Short, Adam M. Smith, Anne Sullivan, Tommy Thompson, and R. Michael Young 113

The Happiness Game
Pieter Spronck, Alex J. Champandard, Peter I. Cowling, and Pier Luca Lanzi . . . 115

Social Network Modelling in Video Games (Part 1)
Shoshannah Tekofsky, Dan Ashlock, Matthew J. Guzdial, Emily Short, David Thue, Michael Treanor, and Jichen Zhu 116

Social Network Modelling in Video Games (Part 2)
Shoshannah Tekofsky, Dan Ashlock, and G. Michael Youngblood 117

Backstory Generation
David Thue, Mirjam P. Eladhari, Joshua Allen McCoy, Mike Preuß, Spyridon Samothrakis, Emily Short, Anne Sullivan, Michael Treanor, and R. Michael Young 118

A General Language for Matching Tile Games
Julian Togelius, Cameron Browne, Simon Colton, Mark J. Nelson, and Diego Perez Liebana 119

Gameplay Evaluation Measures
Vanessa Volz, Dan Ashlock, Simon Colton, Steve Dahlskog, Jialin Liu, Simon M. Lucas, Diego Perez Liebana, and Tommy Thompson 122

Explainable AI for Designers
Jichen Zhu, Rafael Bidarra, Alex J. Champandard, Simon Colton, Reynald Francois, Matthew J. Guzdial, Amy K. Hoover, Antonios Liapis, Sebastian Risi, Gillian Smith, Anne Sullivan, and G. Michael Youngblood 125

Panel discussions

Evaluation
Pieter Spronck 127

Participants 129

3 Overview of Talks

3.1 Some Observations on Automated Strategy Game Design

Cameron Browne (RIKEN - Tokyo, JP)

The program LUDI, developed for my PhD studies [1], evolved combinatorial board games as symbolic expressions representing structured rule trees. The generation of games proved easy – the system could evolve thousands of candidates per second using standard genetic programming operators – but their evaluation proved much more difficult. Evaluation required approximately 30 minutes of self-play analysis per game, and the derivation of appropriate aesthetic indicators made up the bulk of the research effort.

The quality of a game, i.e. its potential to interest human players, cannot be evaluated by its rules alone. Much of a game's appeal lies in the unexpected emergent behaviour that can occur as it is played, which may not be deducible from the rules. For example, the most popular game evolved by LUDI, which it named Yavalath [2], contains two apparently contradictory rules: players win by making 4-in-a-row of their colour but lose by making 3-in-a-row of their colour beforehand. This combination produces an interesting forced move mechanism when played, adding tension and drama to the game and making it interesting for players.

Such emergent strategies can be not just entertaining for players, but crucial for their comprehension of the game. For example, in the recently invented game Omega [3], players' scores are based on the product of group sizes, requiring a degree of calculation that made the game mentally exhausting and difficult to plan ahead. Players found the game opaque and typically made uniformed moves until forced to count the score at the end, removing any tension and making it initially unpopular. It was not until Omega was implemented in the Axiom game system [4] that an emergent strategy was observed. The system's Monte Carlo tree search (MCTS) [5] move planner made moves consistent with a strategy of forming its own pieces into groups as close to size 3 as possible, while forming enemy pieces into groups as far from size 3 as possible. This was later proven to be an optimal strategy for the game [3].

This optimal strategy of "form groups of size 3" provided a convenient meme for players that made the game more comprehensible and outlined a concrete plan of action. It also revealed the game to fundamentally be both a connection game and an anti-connection game [6], as players sought to connect enemy groups into larger configurations while blocking their own groups from being so extended, which imported a whole slew of implicit sub-strategies. Players immediately found Omega more accessible and enjoyable through a simple change of perception, and the discovery of this simple strategy saved the game.

Lantz et al. introduce the notion of the strategy ladder [7], in which players learn increasingly complex strategies relevant to a game as they play it, that build upon each other. They posit that the most interesting games are those with a constant and linearly increasing strategy ladder. This makes good sense as such games would give players both something to play towards (the strategies they know) and something to learn (the strategies they don't know). Games in which winning strategies are trivially learnt and applied would be too simple to be of interest to players, while those in which even the simplest strategies are excessively difficult to learn would be too intractable to be of interest to players. This resonates with the observation by Allis et al. that games which survive do so because they provide an intellectual challenge at a level which is neither too simple to be solvable, nor too complex to be incomprehensible [8].

The question then arises: when automatically evaluating games for their potential to interest human players through AI self-play, what level of playing strength will best capture an authentic experience of the game as played between human players? Random play will obviously not simulate the experience of the game as played between intelligent players. Conversely, superhuman AI play could go too far to the other extreme, and give an equally unrepresentative experience of the game as played by human players. For example, Draughts is drawish when played at even the human champion level – international tournaments have ended with a whimper when finalists drew all 20 games in the final [9] – but remains an engaging and hugely popular game for the average player worldwide.

There is constant and understandable pressure in the game AI research community to strive for superhuman results in all cases. However, I argue that capping the playing strength of AI agents at a lower (strong human) level for the purposes of game evaluation, is more likely to capture a realistic "human" experience of the game. But even estimating what constitutes "strong human level" for a given game remains an open question.

References

1 C. Browne. *Automatic Generation and Evaluation of Recombination Games*. Evergreen Terrace PhD thesis, QUT, Brisbane, 2008

2 C. Browne. *Evolutionary Game Design*. Springer, 2011

3 N. Romeral Andres. From Mathematical Insight to Strategy. In *Game & Puzzle Design*, 1(2):53–54, 2015

4 G. Schmidt. The Axiom General Purpose Game Playing System. In *IEEE Transactions on Computational Intelligence and AI in Games* (T-CIAIG), 6(4):332–342, 2014

5 C. Browne et al. A Survey of Monte Carlo Tree Search Methods. In *IEEE Transactions on Computational Intelligence and AI in Games* (T-CIAIG), 4(1):1–43, 2012

6 C. Browne. *Connection Games: Variations on a Theme*. AK Peters, 2005

7 F. Lantz, A. Isaksen, A. Jaffe, A. Nealen, and J. Togelius. Depth in Strategic Games. In *The AAAI-17 Workshop on What's Next for AI in Games?* (WS-17-15), 967–974, 2017

8 L. V. Allis, H. J. van den Herik, and I. S. Herschberg. Which Games Will Survive? In *Heuristic Programming in Artificial Intelligence 2: The Second Computer Olympiad*, Ellis Horwood, 232–243, 1991

9 C. Freeling. Dameo: A New Step in the Evolution of Draughts? In *Game & Puzzle Design*, 1(2):28–37, 2015

4 Working groups

4.1 Game Search Space Design and Representation

Dan Ashlock (University of Guelph, CA), Cameron Browne (RIKEN - Tokyo, JP), Simon Colton (Falmouth University, GB), Amy K. Hoover (NJIT - Newark, US), Jialin Liu (Queen Mary University of London, GB), Simon M. Lucas (Queen Mary University of London, GB), Mark J. Nelson (Falmouth University, GB), Diego Perez Liebana (University of Essex - Colchester, GB), Sebastian Risi (IT University of Copenhagen, DK), Jacob Schrum (Southwestern University - Georgetown, US), Adam M. Smith (University of California - Santa Cruz, US), Julian Togelius (New York University, US), and Vanessa Volz (TU Dortmund, DE)

Automatically generating games, including rules, environments, and boards, is a difficult type of automatic content generation [2, 5, 6, 8]. There are many possible levels on which Automatic Game Design (AGD) can happen. One of the simpler types of AGD is to take an existing game with a large number of parameters and set those parameters so that the resulting game is fun or interesting [4]. This group discussed the possibilities for making the problem of automatic game design easier by designing the search space, while assuming that only playable games are of interest.

The manifesto the group decided on was to locate search spaces in which most points were good games, or at least search spaces that were enriched with good games. The simple example of parameter tuning to locate a good game shows that even the simplest case of automatic game design may need to assume some degree of automatic tuning. This means that a highly enriched search space may be one that assumes that the fitness measure is *Baldwinian*, it uses the object being evolved as the starting point for a local optimization and returns the fitness of the optimized object.

Tools for search space design and enrichment

An approach that has been effective in the past is to carefully design the representation for search. A simple example of this would be to create a collection of level fragments that include a matching rule for how they may be linked, and then evolve the choices of which fragments to use. This amounts to decomposing the level design problem into a fragment design problem and a fragment assembly problem, both of which are easier. This approach is synergistic with automatic parameter tuning.

A more general approach that includes decomposition is that of search space transformation. Abstract the essential details that define multiple versions of a puzzle and then search the abstraction. The principle of transforming the search space can be realized in a number of ways and might make a good target for a special session at a conference.

Another potentially important element is to create a search space for games that embeds common user experience measures within the design or which leaves hooks or entry points for measuring or sampling user behavior. This speaks not so much to the space of games defined, but to effective design in which the game designed, whatever its nature, is relatively easy to evaluate from a user experience perspective. This sort of design criteria can include creating games that are easy to crowd-source the gathering of user experience data.

Many game design spaces are *ad hoc* with the parameters to be tuned arising from code that plays an initial version of the game. While being able to tune whatever game someone sends you source code for is a valuable skill, designing search spaces of games may be made easier by employing formal semantics. Evolutionary computation can search the realizations of a grammar without difficulty. Incorporating formal semantics into the design space for a category of games employs a natural and well developed set of tools within computer science and is a method for both search space decomposition and transformation. It also would help to formalize the process of including user experience data gathering hooks.

Another possible approach is to discover structures that are highly evolvable. If we have a game specification for a game that is evolutionarilly close to other good games, then at least the part of the search space where these games lie is an enriched search space. This idea echos the idea of exploitation in evolutionary computation: having found a good area, examine it more closely. This, in turn, raises the issue of representation. Turning the problem on its head, an effective representation is one in which good games are often close to one another. In this case the metric used to define "close" is evolutionary time needed to discover one object when the other is available in the evolving population.

The *mutual fertility* of two members of an evolving population is the expected fitness of their offspring, excluding offspring that are in effect clones of the parents. Searching for high fertility parents and then using these as population seeds is an automatic way of locating a rich search space. If the final phases of a design process are human-in-the-loop evaluation, locating high fertility starting points may be a way of increasing the efficiency of human-in-the-loop systems by avoiding user fatigue. it is worth noting that the fertility of parents is an algorithm-dependent notion.

Anchoring a rich system

The group proposed the following method of anchoring and evaluating a rich search space. It should be possible to express existing games in the representation used for search and use such games as anchors and as genetic seeds. The definitive quality measure for such a game search space is the rate at which good, novel games are discovered. Requiring that existing games be possible is a way of placing the search space in what is already known to be good territory. There may be a price paid in degree of novelty and the group thought this might be an excellent area for additional research.

A subgroup continued to investigate this thought further, following the idea of using a Generative adversarial network (GAN) to encode the anchor. Generative adversarial networks are a popular algorithm in machine learning usually applied to image generation, where impressive results were obtained in terms of the imitation of the style of the training image set [3]. The group therefore investigates how GANs can be applied to PCG using the MarioAI framework as an example. The developed approach is detailed in the following.

We use a GAN to generate Mario levels represented as a matrix of integers that encode the different tiles and enemies in the level. The training corpus are the levels for the original Super Mario Bros. game taken from the *Video Game Level Corpus* [7]. After the GAN has concluded training, we have obtained a model that is able to generate different Mario levels based on a random input vector. Next, we optimise the input vector (based on [1]) in order to obtain playable Mario levels using a CMA-ES. Our evaluation function for the playability of a level is based on importing the generated levels into the MarioAI framework and running the (A* agent) on them.

We were able to observe very promising results with our first prototype in terms of the ability of the GAN to reproduce a basic level structure. However, almost none of the generated levels were playable due to the height of the required jumps. We therefore started experimenting with different versions of GANs as well as representations.

References

1 P. Bontrager, W. Lin, J. Togelius, and S. Risi. Deep interactive evolution. In *Proceedings of Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMusArt 2018)*, 2018.

2 Cameron Browne. *Evolutionary Game Design.* Springer, 2011.

3 Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 2014 Conference on Neural Information Processing Systems (NIPS 2014)*, pages 2672–2680, 2014.

4 Kamolwan Kunanusont, Raluca Gaina, Jialin Liu, Diego Perez-Liebana, and Simon Lucas. The n-tuple bandit evolutionary algorithm for game improvement. In *Proceedings of the Congress on Evolutionary Computation*, pages 2201–2208, 2017.

5 Mark J. Nelson, Julian Togelius, Cameron Browne, and Michael Cook. Rules and mechanics. In *Procedural Content Generation in Games*, pages 99–121. Springer, 2016.

6 Thorbjørn S. Nielsen, Gabriella A. B. Barros, Julian Togelius, and Mark J. Nelson. Towards generating arcade game rules with VGDL. In *Proceedings of the 2015 IEEE Conference on Computational Intelligence and Games*, 2015.

7 Adam James Summerville, Sam Snodgrass, Michael Mateas, and Santiago Ontañón. The VGLC: The video game level corpus. *Proceedings of the 7th Workshop on Procedural Content Generation*, 2016.

8 Julian Togelius and Jürgen Schmidhuber. An experiment in automatic game design. In *Proceedings of the 2008 IEEE Symposium On Computational Intelligence and Games*, pages 111–118, 2008.

4.2 Blue Skies Research Questions in AI-driven Game Design

Peter I. Cowling (University of York, GB), Elisabeth André (Universität Augsburg, DE), Rafael Bidarra (TU Delft, NL), Pier Luca Lanzi (Polytechnic University of Milan, IT), Mike Preuß (Universität Münster, DE), Spyridon Samothrakis (University of Essex - Colchester, GB), Pieter Spronck (Tilburg University, NL), Shoshannah Tekofsky (Spirit AI - London, GB), and Georgios N. Yannakakis (University of Malta - Msida, MT)

The aim of this group on the second day was to generate high-risk-high-return research directions to stimulate research in the games AI research community. The questions discussed and a brief commentary are given below. Some of these ideas were developed further in the workshop and will be discussed in other reports. The titles are deliberately short – since the groups initial task was to take 10 minutes to come up with a title of at most 6 words – this seemed to stimulate creativity.

Individual Turing Test (Shoshanna): Can we create an AI that learns from player data for both a group of players and an individual, that can pass the Turing test of having a behavioural profile in the game which is hard to distinguish from the individual player. Having effective AI Natural Language Processing would be a game changer for this – but there is much we can do without advances in NLP.

Paths of inevitability for persistent worlds (Spyros): Games and simulation environments tend to reach a point of stagnation where player actions, interactions and perceptions tend to repeat. It is desirable to provide a revolutionary shake-up of game worlds from time to time to maintain game and player interest. The question was to use AI to find the minimal change "meme" which would snowball into large-scale change without any stage of the change feeling unnatural to the player (consider the disruptive effect of mobile phones on society rather than that of wholesale population uprising).

Non-paradoxical entertaining tuned challenge (Pieter): An AI which acts as a director in a game, tuning the game world to an individual player. This might be seen in the context of an AI which chooses human opponents/team members with similar levels of skill to the player.

ML (Georgios): This project proposal involved extracting information about player interaction with a game – and used small amounts of data about an individual and transfer learning to learn what an individual player might like. Understanding and providing critique of what the ML is doing is a key.

Positive Computing (Elisabeth): There has been much interest in the HCI community as to value-centred design – for example designing games to make players happier, rather than simply to make them absorbed/addicted. The Wild Divine Game was cited as an interesting example in this area. Also replacing humans by AIs "just because we can" was not felt universally positive.

Science of Happy (Peter): Games, interactive and social media reach around 50% of planet earth, and through interaction have the potential to be a major positive influence on society. The research questions were about games with the purpose of making the player happy, through encouraging players to exercise, be mindful, connect with others, reflect, set goals etc. Habitica was cited as an interesting example where habits and action lists were combined into a game. We can conceive of AI mentors (possibly data-driven – Amazon recommenders know what you might want to buy – an AI mentor recommender might propose some action you could take to encourage positivity and positive self talk). Approaches such as Cognitive Behaviour Therapy might be embedded... There are psychology models that could help – such as the "big 5" dimensions of personality (Extraversion, Agreeableness, Conscientiousness, Neuroticism, Openness).

The questions above related to happiness and positivity were taken further by two groups (that later merged) to consider multi-agent economic models which also consider happiness, and whether policy or environment change could be investigated in such models – for example by the introduction of an effective meme. Work is needed to understand the work in the economic and sociology literature already undertaken in this area. It seems likely contributions using the expertise of the Game AI research community would be in embedding significant intelligence in the agents, and having an AI "government" which investigated the space of "political systems" for leading the system of agents. AI is also needed for analysis and understanding of the data from the multi-agent system.

While it is challenging to consider "AI for Games to change the world" it was useful to reflect and work together on this question of huge scale and potential impact.

4.3 Some Industry Interests in AI-driven Game Design

Peter I. Cowling (University of York, GB), Sander C.J. Bakkes (Tilburg University, NL), Pier Luca Lanzi (Polytechnic University of Milan, IT), Adam M. Smith (University of California - Santa Cruz, US), Pieter Spronck (Tilburg University, NL), and Shoshannah Tekofsky (Spirit AI - London, GB)

The group convened in the first session of the week to discuss the research questions collected by Peter Cowling before the workshop (given below). The goals of the group were to:

- Find low-hanging fruit – research questions from the games industry where approaches which are well known to the games AI research community might be used.
  - Need to talk to industry about potentially useful techniques from research – these may be used "deeply ingrained problems" such as bot behaviour where only a limited range of approaches is currently used in industry.
  - Given many examples of X, generate more examples of X (where X is art/music/levels/...) – felt to be a low-hanging fruit with the ready availability of ML and analytics tools.
  - A "button monkey" for games testing, but which more effectively explored the possibility space via MCTS/ML, was also felt to be low-hanging.
- Understand how things function in games companies, and how research ideas might be introduced, and how best to talk to games companies.
  - It was mentioned that amateur/hobbyist games developers are sometimes willing to give full source code for a game
  - Probably useful to both academic and industry communities if models and examples for academic-industry collaboration were made available
  - "Assist a designer" is more useful than "automatic/semi-automatic generation"
  - Academics with good industry contacts, and games industry folks who regularly attend academic research events, can provide an important brokerage service (as had been done here). Also attendance and talks at industry conferences such as the GDC AI summit and nucl.ai were felt promising. Attendance at research events at labs where there is significant industry input (e.g. DC Labs/IGGI at York) was also felt useful
  - Persuading larger games companies to sponsor prizes could be a useful way for them to get prototypes and recognise potential employees
  - Game jams also work well
  - PhD and masters students consultancies/placements provide good opportunities to embed technologies in games companies
- Understand our values – what does the games AI research community want from relationships with the games industry – where are the win-wins?
  - AI for back story was felt to be an interesting, difficult challenge
- Understand some of the "blockers" to getting research AI into industry
  - AI techniques such as ML might be perceived as "too risky" by the games industry, hence risk mitigation is likely to make research AI proposals more acceptable to industry
  - AI will be rated in terms of industry producers in terms of the enhanced experience for the player – there may be a perception of a small return for better AI compared to more levels or better art

Overall we had a positive and interesting discussion with much food for thought to take further later in the week...

The following question was asked by Peter Cowling to various companies which work in game development: **Research in AI-Driven Games Design should consider specific research questions and areas such as...** The following answers were given:

From a **variety of people at one of the world's largest games companies**:

- Machine Learning assisted art production:
  - Make tools that can create music using deep learning (see for example http://www.asimovinstitute.org/analyzing-deep-learning-tools-music/), and can generate the music on the fly adapting to the current situation the player is.
  - Develop a library that can learn the art style of a game to help generate more assets following that style, similar to deep learning demos that can transform a picture to match a well-known painter style (https://qz.com/495614/computers-can-now-paint-like-van-gogh-and-picasso/)
  - The previous two example were based on audio and 2d data, but can a similar system be created for 3D assets, allowing to help the creation of 3D models within the same art style of the rest of the game.
  - e.g. the game developer wants to give the ML assisted art production tool examples of things they want in their game and say 'make more of this please'.
- Machine Learning for QA:
  - Test which improves itself (in terms of coverage, scope, severity or performance, etc.) within a set of boundary rules.
- Interpreting inputs with Machine Learning:
  - Visual inputs for computer vision
  - Audio object recognition
  - User inputs - voice, face, eye, hand, gesture, movement, brain (e.g. gaze tracking, body tracking)
  - The basic problem is to 'create context from inputs'. (Is the system understanding what the user is doing? 'semantics', the meaning of the input from the system's pov.) (Is the system understanding what the user is feeling - 'emotion'.)
- Intelligent Agents:
  - Conversational interface – being done in Bots and Intelligent Assistants?
  - Intelligent agents that support multiple users (e.g. multiple users' voice input)

From a **CEO of Company making Games With a Purpose**: "Google just integrated AI-APIs into Android. Intel and Qualcomm both have AI chips being manufactured for mobile. What are researchers doing about these?"

From a **Head of Technology at a medium-sized games company**: "One interesting problem that has landed on my desk recently is the design of a system that can automatically generate UI based on a hierarchical tree of information. This information isn't limited to simple check boxes and input fields. It is more of a mind map of all the components that go to make something as complex as a car, from the valve specification right up to the colour and shape the bonnet. Each component has descriptive text along with links to child components that lie within that system or feature. Each component needs UI to allow fields to be changed and customised. Generating a good comprehensive UI for such a set of data is a huge task. An AI helper could be of use in this space.

UI design is an often overlooked feature of any game. It is very hard to get right and is often a lot more complex than it would appear at first glance.

It is also however the first experience that a player gets of any game. Like the lobby of a hotel, it needs to look and feel 'right'. An AI that can generate good UI look, feel and flow from loose descriptive data of the content and a style guide would be interesting.

Following on from this would be the first time user experience (FTUE). As developers we're often too close to the product to 'see' it as a first time user would. From game boot to 1st play session. Ideally focus testing helps in this space. This then has the problem that you need to find new users continually, and then assess how consistent their responses are from group to group. An AI that can give an assessment of FTUE in a consistent and measureable manner could be helpful."

From an **Academic Archaeologist with successful museum exhibitions (50K visitors) using VR**: "I am interested in how place based games can help people to engage with museums and cultural heritage sites (e.g. country houses). Nobody has any doubt that museums and old buildings have personalities, I want to know whether AI can help give these personalities a voice. How can people have conversations with old buildings? We can learn from the past, can the past learn from us?"

From a **Tech/programmer for a major games service provider**: "How reinforcement learning agents can be applied for testing across multiple games?"

From an **Academic researcher in game AI**: "Co-operative and / or partially observable games. See Ms. Pac-Man Vs Ghost Team as an example of taking a game and designing a PO version. Needs good AI for AI driven design which the competition is providing ready for the next step."

From a **CEO of a small games company**: "Procedural generation of back story for NPCs in RPGs – where do they come from and why does this cause them to behave in a particular way – also talked about memory for in-game agents so that even if they have limited responses they make the best response – not just the same response every time."

From a **CEO of a micro VR/games company**: "AI to test immersion in a VR game – immersion and realism are very different things. Using VR to visualise (and "play with") complex datasets."

From a **Programmer/researcher working in a (high tech non-games) SME and in a games lab**: "When games are used for serious purposes on human players, such as psychoanalysis, skill analysis, behaviour analysis, establishing baselines (such as the navigation abilities of the general population in Sea Hero Quest) or even treating medical/psychological conditions, then they need to be realistic to elicit realistic behaviour from the players under analysis. Hence, an obvious use case is employing AI to create realistic environments, realistic in-game characters, realistic metrics and realistic game play. Also, at the moment, such games can be quite prescribed so it would be good if they were more adaptive to the players – again for realism. However, this requires transparency of the algorithms – need to know what the algorithm is "doing" throughout (and preferably why it is doing it) to allow for thorough, systematic, transparent and effective analyses. I guess the question is how do you model and adapt the complexities of environments, characters and behaviour while maintaining transparency and authenticity, and achieving the desired outcome (serious games and analyses)?"

From a **CEO of a small games company** : "Superfast prototypes for game concepts, levels and terrains - to allow rich conversations with clients. Level design (my take – show it a bunch of levels and it makes new candidate levels)."

From a **CEO of a small games company**: "Level design – given some hand-crafted levels – come up with some similar editable candidates."

From a **CEO of a small games company**: "Convincing organic shapes that can move on their own – without the need for humans to generate pieces – as was done in Spore and No Man's Sky. Could focus primarily on locomotion or on form – or ideally on both. Believable worlds – using the back story and terrain to influence architecture, road layout, NPC occupants etc."

4.4 Machina Ex Machina (Part 1)

Reynald Francois (Ubisoft - Düsseldorf, DE), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Marc Erich Latoschik (Universität Würzburg, DE), Antonios Liapis (University of Malta - Msida, MT), Alexander Nareyek (Singapore, SG), Emily Short (Oxford, GB), and David Thue (Reykjavik University, IS)

The premises of the workshop were to lay the groundwork regarding whether AI systems could be used to profile players in order to create a model of their play-related needs. With this model, the director AI (dAI) could control the different aspects of a game in order to further satisfy the needs of the player and enhance the experience a given player has when playing the game.

The workshop was split in two topics:

Group 1 covered the definition of the player drives and the dAI as an "experience manager". By defining the boundaries of the field of research, the research group aimed at identifying multiple elements:

- The **field of action** the dAI would have on the experience (e.g. player motivation, direct manipulation of the game environment...
- The **tools** dAI needs to perform its task (verbs accessible to the dAI (create, remove, modify, etc...), player modeling, gamer & player interaction history...)
- The **experience metrics** (player satisfaction, temporality, game state VS optimal state. . .)

The ultimate deliverable from this part of the workshop was to provide a proposal for a dAI-to-game-to-player interaction model that would serve has a base for Group 2, explained next.

4.5 Machina Ex Machina (Part 2)

Reynald Francois (Ubisoft - Düsseldorf, DE), Joshua Allen McCoy (University of California - Davis, US), Alexander Nareyek (Singapore, SG), and G. Michael Youngblood (Xerox PARC - Palo Alto, US)

Group 2 continued where Group 1 left off (see above), and focused on the central tool to the dAI performance: **Player Modelling**, and its relevance or portability across multiple experiences.

The goal of the research was to demonstrate that properties inherited from the modelling of the player in one game can be carried over to have an impact on a chosen metric (in this case: player satisfaction) in another game.

The experimental protocol focused on creating a rudimentary player model containing in this case a single information: the risk-averse factor of a player, a property defining how much risks a player will take when playing games designed around high risk / high gains mechanics.

Two simple games based on the mechanic were designed (revolving around dice throwing); a human participant would take the role an hypothetical dAI would have and manipulate the game parameters to try to enhance the metric (satisfaction) and a protocol was drafted as following:

1. Have the player play game 1 and rate his experience
2. Create a player model mapping the risk-adverse factor of the player
3. Simulate the intervention of the dAI on game 1 by modifying the rules to be more risk-prone or adverse depending on the player profile
4. Have the player play game 1 again and rate his experience comparatively to the first playthrough
5. Have the player play game 2 and rate his experience
6. Simulate the intervention of the dAI on game 2 by modifying the rules to be more risk-prone or adverse depending on the player profile with the game 1 data
7. Have the player play game 2 again and rate his experience comparatively to the first playthrough
8. Assess whether the model created following game 1 data impacts favorably the selected metric (satisfaction) when the simulated dAI "enhances" game 2

4.6 What Is Machine Learning/Deep Learning?

Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Joshua Allen McCoy (University of California - Davis, US), Emily Short (Oxford, GB), Michael Treanor (American University - Washington, US), R. Michael Young (University of Utah - Salt Lake City, US), and Jichen Zhu (Drexel Univ. - Philadelphia, US)

Machine learning (ML) has advanced considerably in recent years, especially the subset of machine learning referred to as deep learning (DL) [2]. Public response to these advances has exceeded the advances themselves, impacting academic, economic, and journalistic fields. These factors put pressure on academic practitioners of non-ML artificial intelligence to apply ML to their research. However, there exists insufficient resources to train these academics, given that most educational materials target novices or experts. We identify the need for ML/DL educational resources aimed at this group: AI academics without existing ML/DL training. We focus on deriving introductions to ML/DL appropriate to this audience and identifying their common misconceptions and concerns.

To introduce ML we focus on the development of an accessible definition to separate what ML is and is not, and a classification of ML approaches. We settled on a definition of ML as function approximation. Given this definition, to apply machine learning to a problem one must have some desired input and output and must make the assumption that there exists some function mapping these two classes. These input/output pairs could be states to actions (as in reinforcement learning), elements to categories (as in clustering problems), or images to labels (as in image recognition problems). We divided machine learning approaches according to this definition into three camps: discrete classification (decision trees, reinforcement learning, etc.), continuous classification (linear regression, neural networks), and categorization (clustering). Given this definition we next identify the borders of what machine learning can and cannot do. We identify three five major boundaries of modern machine learning: (1) local vs global coherency, (2) completeness of data representation, (3) amount and nature of training data, (4) existence and handling of edge cases, and (5) explainability.

Local vs global coherency identifies the domains and problems to which modern ML can achieve reasonable success. ML performs well in domains where local coherency is sufficient over global coherency. As an example consider the success of ML approaches at image recognition and generation compared to its relative failure in textual domains. An image of a cat with one pixel altered is still an image of a cat, while a paragraph with a single word altered can have a vastly different meaning. This is also reflected in the second of our identified boundaries: "completeness of data representation", by which we indicate the extent to which the form of representation chosen for data encodes that data's meaning. For example again consider the differences between images and text, an image of a cat more closely aligns with a human understanding of a cat, than do the letters "c", "a", "t".

The next two identified boundaries are related: amount and nature of training data and the existence and handling of edge cases. By their nature, most modern ML approaches fit some distribution to the available training data. This means that the amount and nature of training data will impact the distribution learned, with extremely varied or small amounts of training data leading to worse performance.

Explainability in this context means the ability for an AI or ML system to explain its decisions or understanding to a non-expert human user, typically in natural language. Modern ML, particularly deep learning, has been described as a "black box" system, with decisions being difficult to explain outside of large sets of numbers. While there has been some work in using machine learned-features as a framework for explanation [3], in general the extent to which a domain calls for explanation will impact the success of applying ML.

We note that beyond a concrete definition and boundaries, many expert AI researchers without ML experience share concerns informed by this lack of experience. We identify three major concerns: (1) whether and how AI and ML can be integrated, (2) whether ML can be applied to cognitively-inspired CS research, and (3) whether ML can be applied to game systems.

In terms of the integration of AI and ML systems, we first note that most deployed ML systems make use of traditional AI approaches to fill in the gaps of modern ML. For example, using rules-based systems to translate a predicted distribution into discrete actions. We derive a three-part framework for describing integration of machine learning with traditional or classic artificial intelligence: AI before ML, AI during ML, or AI after ML. In AI before ML systems have input that is first parsed by a hand-authored AI system, such as tagging chat input with a rules-based system before constructing a reply with a ML system. In AI during ML, the AI and ML systems take turns or act concurrently at some task, perhaps a storytelling system with an AI planner handling global planning while an ML system handles sentence-to-sentence transitions. Lastly in AI after ML, AI is used to transform or clarify the output of an ML system, as in the earlier example of translating a distribution over potential actions to a single action.

Traditionally, AI has a strong link to cognitive science. In ML, including deep learning, these algorithms could be used to expose biases in datasets, an area for potential further research. We further identify a framework adapted from neural network research in cognitive science in which deep learning could represent a subconscious or fast processing cognitive process. Guzdial et al. [1] composed an abstract art system in which a convolutional neural net (a type of deep neural network) identifies emotional meaning from images, trained based on human tags of emotion names on flickr image uploads. A traditional AI search process utilizes this ML system as a heuristic or reward in trying to achieve certain visual emotional depictions (using the AI during ML framework as above). This abstract system represents a hypothesis of human abstract artist cognition, which could be experimentally interrogated.

In addition, we sketch out a few abstract frameworks to apply ML to game systems. For example, placing the player in a training role of a reinforcement learner, giving the player indirect control as that agent made its way through levels. Alternatively one could imagine placing the player in the role of an ML expert, tuning parameters to achieve certain model outputs for simplified puzzles. Lastly, we consider a game in which players alter training data for classifiers to seek certain performance.

We conclude by collecting some guidelines for best practice when applying machine learning to a problem. We include this as some AI experts consider ML tuning "black magic". We identify AB testing as a key practice, varying only a single parameter and checking performance. Further we identify a process for applying ML, and especially deep learning, to problems. First, cut down your data representation to only the information you personally would need to map input to output. Second, adapt existing architectures or start with the bare minimum (earn your complexity). Lastly, when tuning the architecture, create hypotheses for what could be going wrong, and vary appropriate parameters to test these hypotheses.

References

1 Matthew Guzdial, Duri Long, Christopher Cassion, and Abhishek Das. Visual procedural content generation with an artificial abstract artist. In *Proceedings of ICCC Computational Creativity and Games Workshop*, 2017.

2 Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

3 Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016.

4.7 AI-assisted Board Game Play

Antonios Liapis (University of Malta - Msida, MT), Michael Cook (University of London, GB), Steve Dahlskog (Malmö University, SE), Mirjam P. Eladhari (Södertörn University - Huddinge, SE), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Emily Short (Oxford, GB), Gillian Smith (Worcester Polytechnic Institute, US), Anne Sullivan (University of Central Florida - Orlando, US), and Tommy Thompson (Anglia Ruskin University - Cambridge, GB)

Over the last decade, an increasing amount of tabletop board games use digital media to replace or enhance elements of gameplay. As an example, the first edition of *Mansions of Madness* [6] required one of the players to take the role of a Keeper as an opponent to the remaining players, handling monster movement and revealing previously hidden information. The second edition of *Mansions of Madness* [8] "is now guided by a companion app that removes the need for a Keeper player, makes for a fully cooperative game, allows for single player gaming, and makes setup quick and easy"[1]. A plethora of other board games use primarily mobile phone applications for time management as in *One Night Ultimate Werewolf* [1], initializing and revealing hidden information when appropriate as in *Alchemists* [4], a combination of both as in *X-COM* [7] and *Space Alert* [3], and enhancing the atmosphere of the game e.g. via the sound effects in *Stop Thief!* [10].

Since board games are seeing a resurgence of popularity and a desire to innovate thematically, mechanically and technologically [5], it is the perfect time to analyze how digital game technologies and artificial intelligence can be applied to enhance analog play. Technology, especially if enhanced with AI, has many possible uses for board games in the near future. On the one end, simple enhancements such as saving or transmitting a game state can allow for games to be played in multiple sessions, or over the Internet. Assistive technology can be used to simulate the board game in a forward model digitally, allowing players to see the results of their actions one or more steps ahead: feedback can be provided to players in abstract form (e.g. whether this move is beneficial or not) or showing the full digitized board in a future state. In such a forward model, opposing players' moves can also be simulated using methods as simple as minimax [11] and as complex as a computational model of each player based on their previous decision-making strategies [9]. Beyond assistive technology for players, games in which a player acts as the game master can benefit from similar technology both for managing non-player characters (e.g. automating or semi-automating decision-making of individual NPCs digitally so that the game master confirms and moves the figures on the board) but also for providing inspiration and assistance during game preparation. On the more ambitious end of the spectrum, board game play mediated by technology can allow for customization of the game's rules: this can be done based on human choice (e.g. checking which optional rules to use from a list), or automatically based on past game data in this player group or a global player community. Automated rule adaptation can be done when the game starts (ensuring that all players understand and learn the rules of play which remain consistent throughout the session) or in real-time as the game progresses (considering play data in the current game for dynamic difficulty adjustment [12]).

[1] Official Frequently Asked Questions: https://www.fantasyflightgames.com/en/products/mansions-of-madness-second-edition/

In light of the plethora of options for integrating technology and AI into board game play, a set of 33 usage patterns were identified. The usage patterns largely focused around increasing "time to fun" (e.g. by simulating the effects of a player action as a tutorial), enhancing the atmosphere (e.g. by adding sound effects or by offering additional rendering options through projectors), increasing depth of play (e.g. by managing hidden information or logistics beyond what a human game master can do), or changing the ruleset (e.g. by creating new rules or adapting existing rules). Using the format for describing design patterns followed by Bjork and Holopainen [2], four example usage patterns were chosen and further detailed: (a) tech for creativity support, (b) tech for atmosphere, (c) tech provides player flavoring, (d) tech adapts/creates rules. With this initial set of usage patterns in place, all 33 usage patterns are planned to be mapped in future research on the topic.

References

1 Bézier Games. One Night Ultimate Werewolf, 2014.

2 Staffan Bjork and Jussi Holopainen. *Patterns in Game Design*. Charles River Media, 2004.

3 Czech Games. Space Alert, 2008.

4 Czech Games. Alchemists, 2014.

5 Owen Duffy. Board games' golden age: sociable, brilliant and driven by the internet. https://www.theguardian.com/technology/2014/nov/25/board-games-internet-playstation-xbox, 2014. The Guardian. Online. Accessed 9 January 2018.

6 Fantasy Flight Games. Mansions of Madness, 2011.

7 Fantasy Flight Games. X-COM, 2015.

8 Fantasy Flight Games. Mansions of Madness, 2nd ed., 2016.

9 Christoffer Holmgård, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis. Evolving models of player decision making: Personas versus clones. *Entertainment Computing*, 16:95–104, 2016.

10 Restoration Games. Stop Thief!, 2017.

11 Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd ed.)*. Upper Saddle River, 2003.

12 P. Spronck, I. Sprinkhuizen-Kuyper, and E. Postma. Difficulty scaling of game ai. In *Proceedings of the International Conference on Intelligent Games and Simulation*, 2004.

4.8 Mixed-media Game AI

Antonios Liapis (University of Malta - Msida, MT), Elisabeth André (Universität Augsburg, DE), Sander C.J. Bakkes (Tilburg University, NL), Rafael Bidarra (TU Delft, NL), Steve Dahlskog (Malmö University, SE), Mirjam P. Eladhari (Södertörn University - Huddinge, SE), Ana Paiva (INESC-ID - Porto Salvo, PT), Mike Preuß (Universität Münster, DE), Gillian Smith (Worcester Polytechnic Institute, US), Anne Sullivan (University of Central Florida - Orlando, US), Tommy Thompson (Anglia Ruskin University - Cambridge, GB), David Thue (Reykjavik University, IS), Georgios N. Yannakakis (University of Malta - Msida, MT), and R. Michael Young (University of Utah - Salt Lake City, US)

Over the last decades, digital technologies have moved away from the personal computer (PC) into cloud computing, ubiquitous computing, intelligent robots and smart devices.

From wearable technologies to remote-controlled household items and from sensors for crowd control to personal drones, a growing range of technologies provides an opportunity for games and artificial intelligence. While artificial intelligence (AI) is already a big part of the Internet of Things, raising concerns in terms of ethics and politics [14], research in game AI has been relatively confined to more traditional computer and video games. Relevant work on wearable technologies as game controllers [13], mixed- or virtual-reality rendering [8], technology-enhanced play in playgrounds [5] or board games [6], or social robots for games [3], has not taken full advantage of AI for controlling or mediating the experience. Using the term *mixed-media* to refer broadly to any media, digital or otherwise, aside from the game loop within a PC or a game-specific database, this working group attempted to map out the broad topic of mixed-media in terms of its applications for game AI.

Potentially, AI could assist games in taking advantage of mixed-media in two principal ways: use input from non-game sources within the game, and move elements of play or gameplay outcomes into non-PC outputs. The usefulness of non-game information as *input* to gameplay is primarily in the identification of *context*. Context can be personal (e.g. information relevant to this particular player) or broader and anonymous (e.g. number of people in the player's vicinity or trending topics on Twitter). Context can come from the actual environment of a player via sensors (e.g. temperature sensors on a player's mobile phone, or surveillance logs in public spaces such as museums), from a variant game controller (e.g. speech detection or face detection as an explicit part of gameplay), from social relationships (e.g. based on social media profiles, real life proximity, or a history of social interaction), from game histories (e.g. past gameplay habits in other games), from cultural histories (e.g. players' demographic data linking to cultural heritage databases), or from temporal context (e.g. the current time of day or date, or the time passed since the user interacted with the game).

On the other hand, the *output* of such applications (playful or not) can also be delivered in non-traditional media beyond digital screens of a PC or mobile device. Closer to traditional digital outputs, mixed- or virtual-reality devices could be considered, along with environmental projections (such as large screens or projections on different parts of a building, in the case of multiplayer games played in a shared environment). On a similar (and familiar) direction, output or intermediate states of a long-lasting game can be shared on social media, potentially soliciting other users' feedback as additional input (to provide even more context, as discussed above). A game's final output, such as a drawing in a collaborative drawing game [11] or an AI-assisted drawing tool [4], could be printed out in paper, fabric [1], or via 3D printing [9, 10], or sonified into a musical piece [12]. More ambitiously, the intermediate states of such a game could be used to control robots, wearable actuators (worn by players), or smart home devices. Obviously, the further the output is from traditional game output, the more challenging the design problem of making such an application a seamless, playful experience while ensuring that the game output is understood as such and –when moving into the real world– remains safe to use and respects players' privacy.

Following an initial mapping of the possibility space for mixed-media game AI, including possible audiences, purposes and challenges, the working group focused on three specific use cases. The first use case focused on textile input to physical output for MEND, a platform where people contribute to a communal art piece (projected on a wall) by scanning embroidered physical objects, one person at a time. The second use case focused on physical input to virtual output, for a game designed around the use of a sensor able to detect laughter in groups [7] as a mediator for when a game session is won and by whom. The final use case problematized the topic of input more broadly, trying to identify context within player's game data; the main question revolved around whether there are signals or log data which can be assumed to be independent of the game context [2] but able to capture the player's context in terms of experience. The broad nature of mixed-media game AI was thus mapped out, and a small part of the design and problem space it can offer was explored.

References

1 Lea Albaugh, April Grow, Chenxi Liu, James McCann, Jen Mankoff, and Gillian Smith. Threadsteading: Playful interaction for textile fabrication devices. In *Installation in Interactivity at ACM Conference on Computer-Human Interaction*, 2016.

2 Elizabeth Camilleri, Georgios N. Yannakakis, and Antonios Liapis. Towards general models of player affect. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, 2017.

3 Filipa Correia, Patrícia Alves-Oliveira, Nuno Maia, Tiago Ribeiro, Sofia Petisca, Francisco S Melo, and Ana Paiva. Just follow the suit! trust in human-robot interactions during card game playing. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 507–512. IEEE, 2016.

4 Nicholas Davis, Yanna Popova, Ivan Sysoev, Chih-Pin Hsiao, Dingtian Zhang, and Brian Magerko. Building artistic computer colleagues with an enactive model of creativity. In *Proceedings of International Conference on Computational Creativity*, 2014.

5 Die Gute Fabrik. Johann Sebastian Joust, 2014.

6 Fantasy Flight Games. Mansions of Madness, 2nd ed., 2016.

7 Simon Flutura, Johannes Wagner, Florian Lingenfelser, Andreas Seiderer, and Elisabeth André. Laughter detection in the wild: demonstrating a tool for mobile social signal processing and visualization. In *Proceedings of the ACM International Conference on Multimodal Interaction*, 2016.

8 Alex Hern. Will 2017 be the year virtual reality gets real? https://www.theguardian.com/technology/2016/dec/27/will-2017-be-the-year-virtual-reality-gets-real. The Guardian. Online. Last accessed 9 Jan 2018.

9 Britton Horn, Gillian Smith, Rania Masri, and Janos Stone. Visual information vases: Towards a framework for transmedia creative inspiration. In *Proceedings of the International Conference on Computational Creativity*, 2015.

10 Joel Lehman, Sebastian Risi, and Jeff Clune. Creative generation of 3D objects with deep learning and innovation engines. In *Proceedings of the International Conference on Computational Creativity*, 2016.

11 Antonios Liapis, Amy K. Hoover, Georgios N. Yannakakis, Constantine Alexopoulos, and Evangelia V. Dimaraki. Motivating visual interpretations in iconoscope: Designing a game for fostering creativity. In *Proceedings of the Foundations of Digital Games Conference*, 2015.

12 Marco Scirea, Gabriella A. B. Barros, Noor Shaker, and Julian Togelius. SMUG: Scientific Music Generator. In *Proceedings of International Conference on Computational Creativity*, 2015.

13 Joshua Tanenbaum, Karen Tanenbaum, Katherine Isbister, Kaho Abe, Anne Sullivan, and Luigi Anzivino. Costumes and wearables as game controllers. In *Proceedings of the International Conference on Tangible, Embedded, and Embodied Interaction*, 2015.

14 The Future of Life Institute. An open letter to the united nations convention on certain conventional weapons. https://futureoflife.org/autonomous-weapons-open-letter-2017/, 2017. Online. Accessed 9 January 2018.

4.9 Emergence

Jacob Schrum (Southwestern University - Georgetown, US), Cameron Browne (RIKEN - Tokyo, JP), Mark J. Nelson (Falmouth University, GB), Ana Paiva (INESC-ID - Porto Salvo, PT), Julian Togelius (New York University, US), and Georgios N. Yannakakis (University of Malta - Msida, MT)

Well designed games often produce emergent behaviour that makes them more than the sum of their parts. The aim of this workshop was to explore this phenomenon of emergence in game design. While it is not strictly possible to design for emergence – it is by definition the occurrence of the unexpected – the contribution of this workshop was to explore what emergence means in this context and ways it might be encouraged in future designs.

Emergence occurs within games in various ways. Higher-order constructs emerge from the interaction of lower-level entities, and such emergence is often a desirable outcome in games, because it allows for new player experiences, and can spare programmers from meticulously micromanaging every aspect of a game's design. In this session, various forms of emergence in video games were discussed, including the emergence of game dynamics, complex strategies, player goals, and social dynamics.

Game dynamics can emerge from low-level rules, such as how physics are simulated. If a game engine allows certain types of in-world objects to interact physically, then player actions can set off chain reactions that are ultimately the result of physical rules. Interesting examples include the gravity gun in Half-Life 2, and the physical simulation in World of Goo. Complex strategies can emerge from the low-level rules dictating how the game is won or how points and resources are acquired. For instance, in the abstract board game Omega, a player's score is the product of the sizes of each cluster of pieces in their color. This low-level definition of player score happens to lead to an emergent strategy, in which maximizing the number of groups of size three that one possesses will in turn maximize game score. A more complex example is the deck-building game Hearthstone, in which various rules about how different cards interact allow for a wide range of emergent player strategies.

Player goals can emerge when a game gives players the freedom to explore a world and perform a wide-variety of interesting actions. The primary example of such goal emergence that was discussed was Minecraft, because certain modes of play do not have any preset goal. Rather, it is up to players to simply survive, and do as they please. Players typically choose to build elaborate structures, but what they build and how is completely up to them. Player goals can also emerge in Massively Multiplayer Online Role-Playing Games (MMORPGs), in which individuals and groups decide where to go in the world and what goals they want to achieve.

Such games also allow for the emergence of social dynamics, due to the fact that human players are allowed to interact. The interactions that a game allows between players, be they combat interactions of economic transactions, can lead to the emergence of interesting social dynamics within the community. However, even in a game where someone primarily plays alone, social dynamics can emerge from the ability to share the fruits of one's labors, such as the ability to share one's creations in Minecraft, or the fact that anyone can share their experiences in any game via the video game streaming service Twitch.

Despite identifying these various forms of emergence, participants struggled to find a firm, and general definition that would allow for concrete measurement of the degree of emergence within a game. However, many proposals were considered. One prevalent notion was that a game with elegant rules could be seen to support emergence if it gave rise to complex and/or unexpected player strategies. Such emergence is certainly desired in certain types of games. However, examples of extremely complex games that allow emergent strategy also exist, such as the aforementioned Hearthstone, and Pokemon. These games are laden with many intricate details, and the details of how cards interact in Hearthstone and how different traits and abilities interact in Pokemon actually lend even more emergent complexity to these games.

Participants also realized that care should be taken when incorporating the concept of "unexpectedness" into the definition of emergence. For example, if a game's design allows for interesting behavior that is unexpected from the point of view of the players, then such an outcome might be an interesting example of emergence. However, the unexpected quickly becomes expected when it can be replicated, thus making such a definition highly subjective. Furthermore, a highly stochastic game could be argued to be full of unexpected outcomes, but such outcomes would likely not be considered interesting, and might even be frustrating to some. If we are considering emergence as a desirable property, then it would not make sense to include such outcomes under the umbrella of emergence.

However, examples of emergence that resulted in negative player experiences were also discussed. For example, the ability to kill other players in Diablo leads to the emergence of Player vs. Player (PVP) style gameplay, but such gameplay could be very undesirable to certain players. In order to foster an enjoyable player experience, many MMORPGs only allow PVP behavior in designated areas or on designated servers, to prevent this emergent behavior from harming the player experience. Unusual glitches, bugs, and exploits can also be considered to emerge from unexpected game rules and design choices, but game designers typically want to avoid this type of emergence.

A useful and actionable idea that was considered for measuring emergence was to tie it to the concept of strategic depth. Specifically, if a game could be designed in such a way that additional mental or computational effort led to a variety of different ways to succeed in the game, then that game could be argued to have emergent strategies. However, computational effort is strongly tied to the feature representation that is used to play a game, so it is unclear how general this idea would be in practice.

Note that computational effort and mental effort by the (human) player can be quite different things in practice. For example, consider the following hypothetical case, based on the game of Hex.

Figure 1 shows a game of Hex, in which Black has won by connecting the black sides of the board with a connected group of black pieces. Now consider two versions of this game:

1. Hex: Players win by connecting their sides of the board with a connected group of their pieces.
2. HexP: Players win by connecting their sides of the board with an even-sized connected group of their pieces, but lose by connecting their sides of the board with an odd-sized connected group of their pieces.

On a superficial reading of the rules, the games do not seem that different. From the computational resources viewpoint there is also little difference; the odd/even group size calculation will be of similar complexity to the connectivity test, and depending on implementation could handled by code that is nearly the same.

However, from the human player's perspective, there is a world of difference between these two games. The human brain is adept at spotting connections, and it is clear at a glance that Black has established a cross-board connection to win the game of Hex shown in

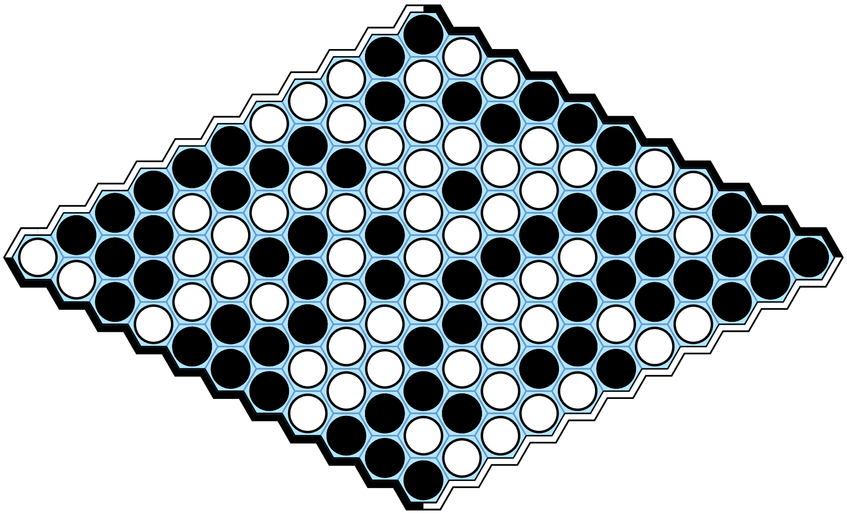

Figure 1 A game of Hex won by Black.

Figure 1. However, in order to determine the winner of the corresponding game of HexP for this same board state, players must count the size of the connecting group (33 pieces) to realise that Black has lost this game by connecting the black sides with an odd-sized group of black pieces. Hex is a brilliant game with clear rules and objectives that allow players to engage in deep strategic planning, whereas HexP would be a dreadful game with a confusing objective in which players must spend most of their mental effort simply counting and recounting group sizes, as groups of pieces grow and coalesce throughout the game. The additional group parity rule might introduce some new strategies to the game, but these are far outweighed by the added complexity that comes with them.

This example highlights that added complexity does not necessarily encourage the emergence of anything interesting. In fact, we could even go further to distinguish between "good emergence", i.e. the emergence of surprising, interesting and beneficial phenomena, and "bad emergence", i.e. the emergence of confusing, annoying and detrimental behaviour. It is not typically possible to predict which form of emergence a given set of conditions will produce. However, we can suggest a simple rule of thumb that might help identify potential discrepancies between computational cost and mental effort: Any rule that involves arithmetic of any sort will typically make a game harder for human players.

Another concrete idea proposed in the workshop for use in competitive games was to measure the size of a Pareto archive of player strategies. Many games are viewed to be interesting if there are multiple ways to win, or if the path to victory strongly depends on the play style of one's opponents. Thus, one could collect play strategies that are the only successful way to defeat other play strategies, even if those strategies were themselves susceptible to yet other strategies. A large archive of such strategies would, in some cases, indicate a game rich in strategic options. However, one could also imagine a degenerate example of a scaled up game of rock-paper-scissors (with more than three options), in which the Pareto archive would be very large, but few interesting player strategies would be available. Once again, it is difficult to distinguish between good emergence and bad emergence with a raw measurement of this sort.

Furthermore, for the proposed ways of measuring emergence, even in the best cases they would measure the strategic complexity of a game more than the actual emergence of that complexity. Short of looking at the code, it is not clear how one would distinguish between a game full of meticulously programmed, individualized responses to various situations, and one in which the same apparent complexity was present, but was the result of comparatively simple programming. However, finding more meaningful ways to measure complexity first could open the door to a more meaningful study of emergence in the future.

4.10 Human-assisted Creation of Content Within Games

Jacob Schrum (Southwestern University - Georgetown, US), Pier Luca Lanzi (Polytechnic University of Milan, IT), Alexander Nareyek (Singapore, SG), and Pieter Spronck (Tilburg University, NL)

Artificial Intelligence (AI) can be used in various ways to create content for video games. A commercial example is the racing game Forza Horizon 2, in which neural networks learn to emulate specific players to serve as opponents for humans. Procedural content generation is also currently used in many modern games, to create levels, among other things. Some prominent examples from academia are based on neuroevolution, such as the evolved weapon firing patterns in Galactic Arms Race, and the selectively bred flowers in the social game Petalz.

Outside of video games, many AI techniques have been developed for creating interesting artistic content. Some techniques that were discussed included basic search and optimization, case-based reasoning, generative/creative adversarial networks, and evolution of various representations, including Compositional Pattern Producing Networks (CPPNs). CPPNs have proven to be particularly flexible, in that they can produce 2D images, 3D shapes, animations, audio tones, soft body robots, and the flowers in the aforementioned Petalz game.

Because of the flexibility of CPPNs, a specific content-creation game was envisioned that could use evolved CPPNs in numerous ways: The Infinite Art Museum. This game concept would allow a player to explore a dynamically generated world filled with artistic artifacts, in which the locations the player chooses to explore would act as an implicit selection function influencing the creation of new art within the world. The geometry of the world would not need to adhere to real-world physical rules, which would allow a high degree of branching of hallways for players to explore. The textures on walls could be generated by CPPNs, allowing players to explore an interesting variety of abstract art. This basic concept could be further elaborated to allow the world to be populated by 3D objects generated by CPPNs, and even CPPN generated sounds (CPPNs can generate basic tones that can be used as "instruments" to play MIDI files) and animations (by querying CPPNs across time). It may even be interesting to allow for some degree of cross pollination between art mediums, so that users can hear CPPN generated sounds based on their 2D texture preferences, or see animations based on the 3D shapes they like the most. An element of action and conflict could even be introduced into the game by using CPPN generated soft robots as enemies.

If the basic design of the Infinite Art Museum described so far could be implemented, it would provide an interesting proof-of-concept demonstrating the appeal of such features in a video game.

4.11 Play

Jacob Schrum (Southwestern University - Georgetown, US), Cameron Browne (RIKEN - Tokyo, JP), Amy K. Hoover (NJIT - Newark, US), Sebastian Risi (IT University of Copenhagen, DK), David Thue (Reykjavik University, IS), Julian Togelius (New York University, US), and Jichen Zhu (Drexel Univ. - Philadelphia, US)

Humans and non-human animals play, and human enjoyment of games seems to be tied to our desire to play in various forms. Playful activities have some evolutionary benefits, such as aiding the development of motor and planning skills. Play also provides organisms opportunities to develop social skills such as the ability to cooperate and negotiate. Because of the benefits that play imparts to real-world organisms, it is natural to ask whether artificial agents in games can benefit from, or even exhibit, playful behavior.

Programming an artificial agent to exhibit playfulness first requires understanding playfulness. To an outside observer, an agent whose behavior and/or goals seem to shift randomly can appear to be playful, and some amount of randomness may indeed be an intrinsic aspect of playfulness. However, playful behavior can also be more focused, seemingly driven by curiosity. Therefore, searching for playful behavior can be facilitated by tools such as Novelty Search, which specifically optimizes for novel outcomes. Agents that imitate other agents are also viewed as playful in some contexts, though such imitation may be purposely imperfect or exaggerated, indicating that the imitating agent is in a sense transforming an observed sequence of actions into a slightly different sequence. Having thought about different forms of playfulness, several prototype programs were devised to make use of such notions.

The first prototype was a variant of Mario including an additional playful Mario agent, referred to here as Luigi. This game allows a human player to control Mario as usual, while an additional AI controller dictates the action of Luigi. In the prototype, Luigi is a variant of an A* planning agent that sets random goals, which results in it running back and forth on the screen in what can be viewed as a very simple form of playfulness. However, the introduction of such an agent opens up the possibility of several enhancements that could be pursued in the future. Specifically, Luigi could play tag with the human-controlled Mario, so that the goal of the game is to chase or escape the other agent rather than to beat the level. Another possible enhancement would be to attempt a competitive speed run with the artificial agent. The reason that this variant would encourage a playful mood within the game is that the two agents can interact, so that they are not merely performing the same actions, but frequently knocking each other out of the way and trying to control their actions despite the presence of competition. A final variant considered is one in which Luigi blithely pursues random goals, while it is the player's duty to protect Luigi from the hazards in the environment.

The second prototype presents a playful world to a human player. This prototype extends previous work in a 2D top-down car-racing simulator in which arbitrary game rules can be introduced. The human player starts by navigating with a car in a plain environment, but a disembodied agent begins introducing rules to alternately aid and thwart the player. Furthermore, another car-driving agent can interact with the human player within the world. This agent alternates between mimicking the human player and attempting to achieve its own goals, which are based on the shifting rules within the world.

The final prototype that was developed recognizes the seeming playfulness of inanimate objects, which is in-turn the product of the human tendency to ascribe agency to such objects. Simple interactions based purely on physical rules can result in the illusion of playfulness. Specifically, the movement of balloons in the wind might appear playful to an imaginative observer, and thus this prototype attempts to create an environment in which simple physical interactions of balloon-like objects appear playful to a human player. The human controls a pin point and tries to pop the balloons, but the opponent AI agent controls the wind, and can either help or hinder the player.

Though simple, these games show promise, and indicate that the concept of playfulness can be used to generate fun games. These various prototypes could be expanded upon to more thoroughly investigate the concept of play, and its usefulness in video games, in the future.

4.12 AI As Reflective Practice

Gillian Smith (Worcester Polytechnic Institute, US), Mirjam P. Eladhari (Södertörn University - Huddinge, SE), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Emily Short (Oxford, GB), Adam M. Smith (University of California - Santa Cruz, US), Anne Sullivan (University of Central Florida - Orlando, US), Tommy Thompson (Anglia Ruskin University - Cambridge, GB), and R. Michael Young (University of Utah - Salt Lake City, US)

This group formed out of a common interest in exploring why the act of creating AI systems is personally satisfying. In doing so, we discussed meaning, subjectivity, and bias in AI systems, and the implementation and adoption of AI systems for artistic expression. A common theme that emerged through the conversation was that of treating AI creation as a reflective practice.

Why Do We Do AI? Though scientific curiosity is a uniting factor in why each member of our group performs research in artificial intelligence, we share several other motivations of a more personal and subjective nature. Several group members commented that building AI systems involving virtual characters is empowering: it provides a means to experiment with different interpretations of human behavior, and to learn from the gaps between simulation and reality. Others found building AI systems to be satisfying because it is creative and expressive as an artistic medium [4], and because of the excitement for AI's potential to challenge pre-existing notions of what games can be [2, 9].

AI and Meaning. There was a commonly shared sense that creating AI systems is a deeply personal act, though as scientists we are taught and often strive to minimize subjectivity. AI systems are formalizations of theories pertaining to the domain the system operates on; while they may be based on theories from academic literature in relevant disciplines, they may also be used to encode and continually iterate upon personal perspectives and theories. A major theme that emerged in our discussion on meaning, subjectivity, and bias in AI systems was that of grappling with reductionism. In order to make theories concrete enough to be operationalized algorithmically, they must be stripped of nuance and interpretation. Such reductionism runs counter to humanistic goals; ironically, even in AI systems designed to simulate virtual humans, the algorithms can be dehumanizing in their reductionist formulation of social interactions. For example, The Sims [5] presents humans as manipulable entities with needs that are constantly eroding and must be satisfied by player intervention, reducing people and relationships to a small number of meters that must be managed.

A second theme in our discussion was that of treating AI as a means for reflection on the process of design. One group member described the act of creating a computational generative system as that of "explaining myself to an alien intelligence". The act of specifying every detail of a desirable space of artifacts to be generated to an entity that does not "understand" can be a reflective one, as each discovery of a forgotten rule prompts its specification.

AI for Artistic Expression. Several members of the group are practicing artists as well as artificial intelligence researchers. Thus, we spent some time discussing the relationship between art and artificial intelligence, and how we use and research AI in our own creative practices. A common theme that emerged was that of embracing error: in the arts, mistakes are a natural part of the creative process, and are often not problems to be fixed but rather treated as serendipitous. Currently, "mistakes" in AI systems are often seen as bugs to be squashed. This group discussed what it would mean for AI systems to be able to make mistakes and grow from them.

Fostering Reflection through Constructing AI Systems. We have argued thus far that there is value in treating the creation of AI systems as a personal and reflective process. A question then emerges: how, if at all, can we make that process accessible to those who do not have a background in artificial intelligence or even in computer science? In the context of AI-driven game design, we discussed the potential for mixed initiative design tools (e.g. [3, 8]) that support users in deeply reflecting on what they design. Inspired by Schon's framing of design as conversation (and, indeed, as reflective practice) [6], we discussed novel methods for engaging in that conversation, both in terms of the interface used to make design decisions, and the way in which feedback is provided to the designer. With regard to interaction design, we discussed ways in which tactile and tangible interfaces could permit a more organic and playful design exploration: pushing and pulling a system's expressive range [7] to include (or exclude) certain kinds of designed artifacts. Providing deeper feedback to designers was centered around providing rapid feedback to users in terms of impact on expressive range, inspired by Cook's Danesh [1], and we discussed several new potential metrics and visualizations to support reflective design.

References

1 Michael Cook, Jeremy Gow, and Simon Colton. Danesh: Helping bridge the gap between procedural generators and their output. In *Proceedings of the 2016 Workshop on Procedural Content Generation in Games*, co-located with FDG/DiGRA 2016.

2 Mirjam P. Eladhari, Anne Sullivan, Gillian Smith, and Josh McCoy. *AI-Based Game Design: Enabling New Playable Experiences*, UC Santa Cruz Baskin School of Engineering, Santa Cruz, CA, 2011

3 Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. Sentient sketchbook: Computer-aided game level authoring. In *Proceedings of ACM Conference on Foundations of Digital Games*, 2013

4 Michael Mateas. Expressive Intelligence: Artificial Intelligence, Games and New Media. In *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*. Springer Berlin Heidelberg, 2–2: 2007

5 Maxis. *The Sims*. PC game, 2000

6 D. A. Schon. *The reflective practitioner: How professionals think in action*. Basic Books, New York, NY, 1983

7 Gillian Smith and Jim Whitehead. Analyzing the Expressive Range of a Level Generator. In *Proceedings of the Workshop on Procedural Content Generation in Games*, co-located with FDG 2010

8 Gillian Smith, Jim Whitehead, and Michael Mateas. Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design. In *IEEE Transactions on Computational Intelligence in AI and Games* 3(3): 201–215, 2011

9 Mike Treanor, Alexander Zook, Mirjam P. Eladhari, Julian Togelius, Gillian Smith, Michael Cook, Tommy Thompson, Brian Magerko, John Levine, and Adam Smith. AI-based game design patterns. In *Proceedings of the 2015 Conference on the Foundations of Digital Games*, 2015

4.13 The Happiness Game

Pieter Spronck (Tilburg University, NL), Alex J. Champandard (AiGameDev.com KG - Wien, AT), Peter I. Cowling (University of York, GB), and Pier Luca Lanzi (Polytechnic University of Milan, IT)

Computer games are used to entertain people, train and educate them, support social interaction, and sometimes to learn about industrial and societal processes. While these are lofty goals, in an idealistic vision one can imagine that computer games are used in a much more world-changing manner. The question that we asked is "Can computer games be used to improve the overall happiness of the human race?"

Our answer to this question is the idea that a game can be developed that contains knowledge on what makes humans happy, and allows for experimentation with different societal constructs and different political systems to see what their effect is on human happiness. As such, the game would be a realistic model of happiness, and provide convincing arguments for particular social changes. We discussed and implemented an initial version of the so-called "Happiness Game."

The purpose of the Happiness Game is to emulate a world of agents which represent humans. The agents can interact in various ways. The agents can also be happy and unhappy to certain degrees. The main interaction between the agents is "trading," as that is the basis for human interaction. However, agents may also exhibit behaviors that are not materially beneficial, but may improve their "lives" by increasing their own happiness, or the happiness of other agents. Agents in the system may simply emulate certain natural behaviors, but may also be designed to try to exploit the system, or to learn to interact with the systems in order to maximize their effectiveness in acquiring material wealth or happiness. We implemented various agent strategies for proposing and accepting/rejecting trades.

The initial implementation of the game functioned correctly, but also quickly demonstrated that the concepts of "happiness" and "survival" are not easy to pin down. For instance, while the system would make agents which were too unhappy die off (which seems realistic), it also allowed agents to survive just by "being happy" (which is unrealistic). The reason is that in the initial system both food and happiness were both used to track an agent's survival state, but that either was sufficient for an agent to survive. Thus, our conclusion was that the concept of happiness and how it would propagate in the system needs a careful design, before a second generation of the system is built.

In a second generation, we look forward to experimenting with political systems, which may be imposed upon the system to affect the existence of the agents. For instance, a socialist system may take much of the agents' wealth to redistribute it over the population, while a communist system forces the agents into particular behaviors and a capitalist system leaves the agents free to behave in whichever way they want, including exploiting other agents. The system can then be used to study the effect of political systems on the quality of life of the agents, but also experiment with new, original political systems which may lead to an overall higher quality of life for all agents. This way, the effects of social change may be envisioned before the change is actually made.

The ultimate goal of such a system would be to increase the quality of life of all humans. Naturally, building a simulation of the necessary complexity is a huge undertaking, needing tens of millions of euros. However, small, limited prototypes can be developed cheaply to examine the workings of the system and as a guide for coming up with the necessary requirements of the system, and perhaps as a convincing argument to find grant money for building the systems that the world actually needs.

4.14 Social Network Modelling in Video Games (Part 1)

Shoshannah Tekofsky (Spirit AI - London, GB), Dan Ashlock (University of Guelph, CA), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Emily Short (Oxford, GB), David Thue (Reykjavik University, IS), Michael Treanor (American University - Washington, US), and Jichen Zhu (Drexel Univ. - Philadelphia, US)

The workshop started with a quick introduction of what social network modeling is and how it may map to video games. A social network model is a graph-based data structure where players are the nodes in the graph, and the edges between players are messages, relationships, or interactions between the players. A node contains (semi-)static information about the player, such as their personality, game demographics (profile), real life demographics, and session distribution. An edge can contain information about what characterizes the connection between two given players such as social style, weight (frequency/intensity), topic, cooperation/defection, and so forth. A network is the entire set of nodes and edges. It allows for calculations to be executed to approximate concepts such as popularity (in-degree) and social influence (centrality).

Having defined a social network, we continued on to brainstorm about applications of social networks in video games. There were four major threads identified. First, a game could be created where an AI attempts to disrupt or influence the social network of humans. The gameplay element could be to identify who is the AI and who is a regular human. This is in some ways reminiscent of Werewolves. Second, a game could be made that takes a human player's social network as input and generates a narrative game that happens in a similar social space. The game would thus have reflective properties for the player. Third, a game authoring tool may be developed that takes social network motifs as input for the creation of a story. The game could adapt to disruption of social motifs crucial for the story. For instance, if the story is based on a love triangle, but the player removes one of the characters in the love triangle, the game may recover by establishing a love triangle between different characters. Fourth, a game could be created where the human player is a "gardener" or deity that shapes a social network of fictional characters as they play out their daily lives. This is similar to the Sims franchise of games, but with a focus on social relationships instead of human needs.

4.15 Social Network Modelling in Video Games (Part 2)

Shoshannah Tekofsky (Spirit AI - London, GB), Dan Ashlock (University of Guelph, CA), and G. Michael Youngblood (Xerox PARC - Palo Alto, US)

This session started with a discussion of Bruce Tuckman's Stages of Group Development (1965) and how social network modeling and analysis could be used to represent, track, understand, and manipulate (for better or worse) this process—especially in the accelerated timescale of a game. This framing led to the discussion of other social phenomena that could also be represented with similar attributes and a progressive morphology.

The two days of conversations led to the identification of Social Network Modeling in Video Games constituting a novel field of video game studies. It is the synthesis of the methods and insights from the Social Network Modeling field to the application field of Video Games. The participants resolved to write a book as a primer and intro to the field. The topics and chapter structure were discussed.

The book will consist of two parts. Part 1 covers the technical application of Social Network Analysis (SNA) to video games, by introducing the relevant concepts from graph theory, applying the relevant data mapping to the graph for the video game space, and then working through the process of how to design a network representation for a given application. Part 2 of the book introduces four major application areas for SNA in games: group performance, norm/meme diffusion, social cohesion, and social modeling. First, group performance centers on the concept of matchmaking and development of productive teams in games. Second, norm/meme diffusion describes messages spread through a social network. Third, social cohesion details how players may increase or decrease the connectedness of their social network. Lastly, social modeling expands on the narrative-based applications for SNA that were discussed in part 1 of the workshop.

4.16 Backstory Generation

David Thue (Reykjavik University, IS), Mirjam P. Eladhari (Södertörn University - Huddinge, SE), Joshua Allen McCoy (University of California - Davis, US), Mike Preuß (Universität Münster, DE), Spyridon Samothrakis (University of Essex - Colchester, GB), Emily Short (Oxford, GB), Anne Sullivan (University of Central Florida - Orlando, US), Michael Treanor (American University - Washington, US), and R. Michael Young (University of Utah - Salt Lake City, US)

In the context of Artificial Intelligence (AI) and its applications in digital games, the ability to automatically generate narrative content offers many potential benefits. For authors of game narratives, it could provide additional leverage in the content-creation process, both multiplying the outputs of their work via parameterized generation and accelerating their workflow via AI-assisted exploration of a potential narrative space. For players of game narratives, it could support the real-time adaptation of a game's ongoing narrative, toward providing each player with a unique, personalized experience. While a significant amount of research has investigated story generation as the challenge of generating potential story futures, substantially less work has explored the challenge of generating potential story pasts – so called "backstories".

The need for further discussion of backstory generation became apparent on the first day of the Dagstuhl Seminar "Artificial and Computational Intelligence in Games: AI-Driven Game Design". Two preparatory talks (given by Peter I. Cowling and Emily Short, respectively) both highlighted backstory generation as a topic that creators of both digital games and interactive fiction were keen to see developed. Combined with the proposer's prior interest, these talks inspired the creation of a workshop on backstory generation that took place on the same day.

Given the diverse backgrounds of the workshop attendees, we began the workshop with a discussion of each others' motivations for attending. The result was an initial mapping of the space of potential applications for backstory generation. These took the form of questions that we might be able to answer if a backstory generator could be used, such as:

- How would authors use or interact with a backstory generator?
- What characterizes potential entry points for flexible, meaningful stories?
- How might we answer player questions about the past?
- Given a set of story characters, each with specific characteristics, can we generate plausible stories that explain how those characteristics came to be?
- More generally, given any desired next state of a narrative, can we generate a plausible chain of events that explains its occurrence?
- How might a drama/experience manager maximize a situation's narrative meaning?

Using this list as a basis, we began to build a set of insights and concerns that would be important to consider when developing technology for backstory generation. These included:

- the additional challenge of backstory generation as compared to future story generation (since some backstories could contradict existing story history, the problem is more constrained than future story generation);
- the notion that contradictions between backstories can sometimes be smoothed over by declaring that some part of the conflicting information was false (e.g., lies or faulty recollections);

- the need to develop heuristics to guide decisions during generation, as well as story metrics to use in those heuristics (e.g., how much narrative meaning will be added?);
- the need to respect the intent of authors and integrate it in the generation process (this could mean not filling in some parts of the past, because their ambiguity is intended);
- the value of using multiple levels of granularity/abstraction when representing content (while a scientific approach demands granularity, creative authoring benefits from pragmatic abstraction); and
- the realization that environmental storytelling (i.e., the way in which narrative is conveyed through the visual layout of a scene) is a form of backstory-telling, which makes environment design a rich target for backstory generation.

This work was aided by choosing an example of backstory being revealed in a popular movie (in which a character says "I am your father") and considering how and why such a revelation might be generated automatically.

Noting the importance of respecting the existing constraints of an ongoing story, we continued the workshop with a more detailed consideration of the narrative context in which a backstory generator might be used. Much of this discussion focused on the characters in a story's world and how they might best be modelled to support valuable backstory generation. We identified two parts of such a model as being particularly important: a way for characters to change or suspend their goals or personalities, and a way for their goals or personalities to determine what aspects of prior events they privilege when revealing those events as backstory.

To conclude the workshop, we returned to our initial discussion of what backstory generation is and why we should pursue it, toward forming a tentative synthesis. In our view, a backstory in a narrative game is part of a story that (i) the player experiences non-contemporaneously with its execution, and (ii) that demonstrates (ideally meaningful) connections between two or more elements of the larger story's history. Backstories should be generated automatically because they can add causal richness to a story's world in a way that offers authors additional leverage over the narrative space, and also because they can provide AI-driven story managers with additional flexibility when personalizing each player's narrative experience. The key challenges of backstory generation involve capturing and preserving authorial intent as well as ensuring sufficient narrative meaning in the results. Further work on this topic is planned, and we welcome discussions and suggestions for collaboration from all interested parties.

4.17 A General Language for Matching Tile Games

Julian Togelius (New York University, US), Cameron Browne (RIKEN - Tokyo, JP), Simon Colton (Falmouth University, GB), Mark J. Nelson (Falmouth University, GB), and Diego Perez Liebana (University of Essex - Colchester, GB)

Matching tile games are a genre of casual game based on, as the name implies, grids of tiles that need to be matched in various ways to clear them and/or score points. Popular examples include Bejeweled, Tetris, and Candy Crush Saga. This working group was inspired by two attempts to formalize the variation of game mechanics in the construction of this genre of games: a paper that examines the history of how these games as arose historically through

Listing 1 Bejeweled-like game in a JSON file format.

```
{
   "grid": {
      "type": "Square",
      "size": 4,
      "matchtilecell": "Square",
      "blocks": "single",
      "colour": "true"
   },

   "actions": { "rotate": {"size":2} },

   "rules":  {
      "spawning": {"from": "Top", "type": "random"},
      "gravity": {"direction":"down"},
      "match":
      {
         "effect":"Clear",
         "type":"minimum",
         "number": 3,
         "shapes": "OrthoLine",
         "pattern": "colour",
         "reward": {"type": "Incremental", "value":100}
      }
   },

   "terminations": {  "time": {"value":100} }
}
```

adaptations and variations on existing games [4], and a lengthy blog post cataloguing the various axes of variation needed to account for the games in this space [1].

This existing work that we consulted arose from a game-studies motivation, but is formal enough that it seemed promising to adapt it towards a generative model that would allow automatic discovery of new matching-tile games, an instance of rule-focused automated game design [5]. We investigated two approaches to modeling this space of games, both of which are currently in progress.

The first approach investigated adapting the Juul/Bailey style analysis as directly as possible, into a language along the lines of the Video Game Description Language (VGDL) designed at an earlier iteration of this Dagstuhl seminar series [2]. This approach takes each of the axes of mechanic variation they identified, and makes it explicit in a modeling language for matching-tile games. A specific game in this space is loadable from a JSON file by an interpreter. An example of a Bejeweled-like game described in this language can be seen in Listing 1.

The second approach we developed was a lower-level model inspired by interactive Cellular Automata (CA). Here, tiles are modeled as CA-style tiles, and mechanics are CA-style rules specifying how the following state is produced from the current state configuration. This approach does not explicitly capture Juul/Bailey style mechanic variations in the modeling language, instead using a smaller and lower-level set of representational primitives that make fewer assumptions, from which it would be expected that those higher-level mechanics

Table 1 Three in a row scoring mechanic for a Bejeweled-like game.

Purpose	Start / End	Time	Row	Column	Colour	State	Score
Three in a row scoring	Start pattern	t	x-1	y	c1	tiled	0
		t	x	y	c1	tiled	0
		t	x+1	y	c1	tiled	0
	End pattern	$t+\epsilon$	x-1	y	∅	exploded	+10
		$t+\epsilon$	x	y	∅	exploded	+10
		$t+\epsilon$	x+1	y	∅	exploded	+10

Table 2 Tile dropping down mechanic for a Bejeweled-like game.

Purpose	Start / End	Time	Row	Column	Colour	State	Score
Tile dropping down	Start pattern	t	x	y	∅	evacuated	0
		t	x	y+1	c1	tiled	0
	End pattern	$t+\epsilon$	x	y	c1	tiled	0
		$t+\epsilon$	x	y+1	∅	evacuated	0

identified in existing matching-tile games would emerge as particular types of designs. For a similar Bejeweled-like game, different game mechanics could be specified as primitives such as the ones shown in Table 1 (for scoring a match of three tiles with the same colour in a row) and Table 2 (for a tile to drop down when a gap is available beneath it).

Lines of future work include an analysis of the expressiveness of these two game representations, as well as the study of algorithms that can search in these spaces. Additionally, the development of general algorithms that can play this type of games without domain knowledge could be an interesting avenue for future research, following the trends of the General Game Playing [3] and General Video Game AI [6] competitions.

References

1 Match Game Mechanics: An exhaustive survey. https://www.gamasutra.com/blogs/JonathanBailey/20150227/237544/. Accessed: 2018-01-19.

2 Marc Ebner, John Levine, Simon M Lucas, Tom Schaul, Tommy Thompson, and Julian Togelius. Towards a Video Game Description Language. In *Dagstuhl Follow-Ups*, volume 6, pages 85–100, 2013.

3 Michael Genesereth, Nathaniel Love, and Barney Pell. General Game Playing: Overview of the AAAI Competition. *AI magazine*, 26(2):62, 2005.

4 Jesper Juul. Swap Adjacent Gems to Make Sets of Three: A History of Matching Tile Games. *Artifact*, 1(4):205–216, 2007.

5 Mark J Nelson, Julian Togelius, Cameron Browne, and Michael Cook. Rules and Mechanics. In *Procedural Content Generation in Games*, pages 99–121. Springer, 2016.

6 Diego Perez-Liebana, Spyridon Samothrakis, Julian Togelius, Simon M Lucas, and Tom Schaul. General Video Game AI: Competition, Challenges and Opportunities. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 4335–4337, 2016.

4.18 Gameplay Evaluation Measures

Vanessa Volz (TU Dortmund, DE), Dan Ashlock (University of Guelph, CA), Simon Colton (Falmouth University, GB), Steve Dahlskog (Malmö University, SE), Jialin Liu (Queen Mary University of London, GB), Simon M. Lucas (Queen Mary University of London, GB), Diego Perez Liebana (University of Essex - Colchester, GB), and Tommy Thompson (Anglia Ruskin University - Cambridge, GB)

The goal of this group is to develop a framework for logging information from games in a common format that captures common-currency metrics like win/loss, score as a function of time, entropy measures on games state, and others listed subsequently. The framework provides an implementation of a number of general measures that have previously been used to describe gameplay in an abstract form. The relevant information from the game has to be extracted with game-specific code implemented by the user and can then be processed by our framework. The framework is thus capable of logging full game states at each tick of the game, but also allows users to analyse specific characteristics of gameplay. The framework will be made available at [https://github.com/GAIGResearch/GameEvaluationMetricsAtDagstuhl] and is intended to be extensible to include more measures. During the seminar, we have implemented a number of features within the framework and applied it to the GVGAI software, which already resulted in interesting observations

Based on related publications, we have collected a list of measures / features that we envision to include or have already included in the framework. In this list, we have included a standard collection of core measures that are generic across (almost) all types of games and thus permit a broad classification of types of games. We have also included more specific features such as agent decisiveness and contextualised actions, thus enabling more sophisticated comparisons on restricted sets of games. Additionally, we have included some features that are an interpretation of a measure (e.g. drama), but are well-established in our research field along with their definition in the literature. Although we agree that cosmetic aspects of a game have a significant effect on the player, we focus here on features that describe the actual gameplay and exclude aesthetics-based features.

The features we wish to be able to log are listed in the following, those that are starred are captured in the prototype code. The annotation (D) indicates a directly loggable measure while (I) is an indirect one requiring additional processing.

Direct logging features

- Game duration* (in ticks / seconds) (D)
- Agent score*(D)
- Game outcome (win, loss, tie, fail, etc.)*(D)
- Agent actions log*(D)
- Game state log(D)
- Game events (count; frequency)* (D)
- Available actions log / Branching factor* (D)
- Hierarchical actions / Contextualised actions (D / I)
- Lead change (D)

General indirect features

- Spatial entropy (I)
- Action entropy* (I)
- State space entropy (I)
- Object density* (I)

Agent-based features

- Agent surroundings (I)
- Agent decisiveness* (I)
- Agreement of agent with player models(I)
- Agreement between multiple different agents (I)
- AI Critics (I)
- State space exploration percentage (I)
- Empowerment (I)

Interpreted features (with references)

- Outcome uncertainty (I) [3]
- Drama log * (I) [3]
- Engagement (I) [4]
- Skill depth (I) [5]

In the following, we define lesser known and complex terms in this context:

Game Events

A game event is an invocation of a rule of the game, as opposed to a low level action.

Hierarchical Actions / Contextualised Actions

Hierarchical actions are those that aggregate multiple primitive actions in a cluster made meaningful based on their context. Examples include waiting by a power pill until ghosts are close in pac man or the shuffle-shuffle-bend-hand-gesture combination that emits a fireball in Street Fighter. Detection of hierarchical features is part of the future intentions. The group suggested discovery of hierarchical features via the use of bounded time window Markov models or, in the case where the actions are known – like the two examples – can be defined within the code hook as loggable events.

Lead Change

In multiplayer games with an obvious measure of how well the players are performing in a game (such as victory points, score, etc.), the player performing best based on this measure is considered to be in the lead at a given point in time. A lead change occurs if the player who is in the lead changes. The fact that the obvious measure of which player is in the lead is often deceptive is a key feature of many games, for example Reversi/Othello.

Spatial Entropy

The nominal definition of spatial entropy is the Shannon entropy of the spatial position of the agent. This sounds simple, but there is a problem: some positions on the board may not be accessible or, worse, may be accessible based on a state conditioned event or situation. The group decided to record the spatial entropy as if all cells in the game space are accessible. This gives a common currency measure of spatial entropy that is useful for comparison within a game. The practical effect is that the normalization of game entropy will yield a maximum below one where there are inaccessible cells. If the entropy is used for comparison between players or agents this is not a problem at all; if the full, correct value if required for some purpose then calculation of the number of accessible squares can be performed as a post analysis and then the entropy value scaled accordingly.

Agent Surroundings

As agent surrounding, we define the subset of the game state observable by the agent at a given point in time. The intention of logging this feature is to facilitate reasoning about the agent's tactics and strategy. For a 3D game, this could be based on the agent's first-person view. For a 2D scrolling game, this could be the currently observable part of the game, but could also allow a restricting the view to the neighbourhood of the agent in order to facilitate easier reasoning / learning.

Agent Decisiveness

Measures that describe the decisiveness of an agent are of course dependent on the specific agent implemented, but a suitable general measure is an agent's convergence speed. Some agents such as MCTS also have an obvious uncertainty measure.

AI Critics [1]

AI critics are agents with a set of goal distinct from finishing the game that are supposed to simulate player preferences. Based on these goals / preferences, they can offer a critical evaluation of the gameplay. In [1], the goals are expressed as weights on a utility function.

Empowerment [2]

Empowerment is an information-theoretic measure that on the one hand expresses the influence an agent has on the game via its actions and on the other hand the extent to which this influence is observable by the agent.

References

1 A. Liapis, C. Holmgård, G. N. Yannakakis, and J. Togelius. Procedural personas as critics for dungeon generation. In *European Conference on the Application of Evolutionary Computation* (EvoApplication 2015), vol. 9028, 331–343, 2015

2 C. Guckelsberger, C. Salge, J. Gow, and P. Cairns. Predicting player experience without the player: An exploratory study. In *Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY). ACM, New York, NY, 305–315, 2017

3 C. Browne, and F. Maire. Evolutionary Game Design. In *IEEE Transactions on Computational Intelligence and AI in Games* 2(1):1–16, 2010

4 J. Brockmyer, C. Fox, K. Curtiss, E. McBroom, K. Burkhart, and J. Pidruzny. The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. In *Journal of Experimental Social Psychology* 45(4):624–634, 2009

5 J. Liu, J. Togelius, D. Pérez-Liébana, and S. Lucas. Evolving Game Skill-Depth using General Video Game AI Agents. In *IEEE Congress on Evolutionary Computation* (CEC'17), 2017.

4.19 Explainable AI for Designers

Jichen Zhu (Drexel Univ. - Philadelphia, US), Rafael Bidarra (TU Delft, NL), Alex J. Champandard (AiGameDev.com KG - Wien, AT), Simon Colton (Falmouth University, GB), Reynald Francois (Ubisoft - Düsseldorf, DE), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Amy K. Hoover (NJIT - Newark, US), Antonios Liapis (University of Malta - Msida, MT), Sebastian Risi (IT University of Copenhagen, DK), Gillian Smith (Worcester Polytechnic Institute, US), Anne Sullivan (University of Central Florida - Orlando, US), and G. Michael Youngblood (Xerox PARC - Palo Alto, US)

In response to the rapid technological success in AI, the emerging research area of Explainable AI aims to better communicate AI systems' decisions and actions to human users. The central goals of explainable AI are often to increase users' understanding, foster trust, and improve their ability to utilize the systems.

Explainable AI for designers (XAID), in particular, focuses on enhancing designers' capability to (co-)create user experiences with AI. Through the vantage point of computer games, we examine 1) the design space of explainable AI for game designers, 2) three case studies of XAIDs, and 3) design guidelines and open challenges in each case.

We identified different types of XAID techniques that can facilitate the game design and development process. In a broad stroke, they can be categorized into i) what to explain, ii) when to explain, and iii) how to explain. In terms of what to explain, XAIDs can be used to communicate many aspects of game AI. For instance, they can be used to explain the process of the chain of actions and reactions taken by game AI. Alternatively, they can simply explain the results of processing certain inputs. Regarding when to explain, the description may take place before, during, or after AI's operations, each scenario affording different types of explanations. As for how to explain, factors such as the form of explanations (e.g., as a tutorial or as justifications of specific AI actions at hand), levels of abstraction (e.g., concrete details or high-level abstraction), and the interaction model (e.g., AI as a tool for the human designers or as a co-creator with the designers) directly influences of how XAIDs should be designed.

To ground our survey of the XAID space, we conducted three case studies based on the type of AI systems (black-box or white-box) and the part of the game development process in which AI techniques are used. The case studies include XAIDs for 1) white-box procedural content generation (PCG) systems, 2) black-box PCG systems, and 3) black-box NPC behavior control.

In a white-box PCG system, we assume a system that has full knowledge of the underlying processes taking place; this allows XAID (e.g. in the form of text generation) to be embedded within the content generation code. White box PCG can hypothetically output a narrative (as a sequence of sentences), with each sentence produced following each command or function call. Ideal generative architectures for this approach are pipelines, where a number of generative processes are "chained", each producing an intermediate result which is taken by the next process as input and producing an enhanced result as its own output [1]. In such an architecture, generation of textual explanation need not be internalized within the codebase, instead assessing the intermediate results from each process along the pipeline. Explanations in a white box PCG system can be produced at any point in the generative process and in any degree of clarity (as the explanation subsystem can have full knowledge of the underlying logic or ways in which content quality is assessed). Due to these reasons, the challenge for XAID in white box PCG is how to handle the possibly vast volume of information that can be output by such a system. Presenting a compelling and intuitive narrative to the user regarding the choices taken by the system can be done:

- ***sequentially*** in the order that the system makes decisions. This explanation can follow some form of story structure which simulates e.g. the generative pipeline [2]. Work on story generation can be used to enhance how the connections are made between different time slices in the generation (e.g. via natural language processing [3]) so that the narrative is coherent and causal links are made obvious. This can be achieved by post-processing the generated sequence of sentences to introduce throwbacks to past generative decisions which affect future outcomes or to foreshadow how one decision early on affects the final outcome.
- as ***highlights*** of the generative process by filtering out and omitting less interesting points in the generated sentence structure. For this to happen, a number of evaluation mechanisms are needed to assess interestingness (will this be interesting to a human user?), clarity (will this be understandable by a human user?), or creativity (is this point a creative milestone [4] where the design shifts?) of the text or the underlying generative commands that prompted it. Therefore, it is necessary to have the whole narrative (sequence of sentences) before the most interesting points within it are chosen in a post-processing step.
- ***non-sequentially***, summarizing the explanation starting from the most important points regardless of whether those are performed first or last in the code. Indeed, it is possible to start by presenting a description (visual, textual, or otherwise) of the final artifact and backtracking some of its most interesting elements on points in the generative process where those happened. Moreover, tropes such as sports game summaries can be used as inspiration, presenting the main outcomes of the generative process first (as non-sequential highlights) followed by a longer form of the sequential narrative regarding how generation progressed from unformed to fully formed content.

An open challenge in providing useful XAID is how to fit the entire processes of the white-box system into something that is compact and yet sufficient for designers.

In a black-box PCG system, we specifically looked into what type of XAID would be useful for game designers at a AAA game studio. We determined an "AI as student" framework matched designers intuitions when it came to artificial intelligence and black box machine learning techniques. To fit this framework a potential agent would need to: (1) share a common language of design with the human designer, (2) communicate its current understanding of this language, and (3) update this understanding in response to designer feedback. We identified three areas of existing work that matched these requirements: zero and one shot learning, explainable AI, and active learning respectively. In practice this would look like a designer pre-defining content with certain tags and an AI training on these tags (zero and one shot learning), the designer interrogating the agent's output content and tags through checking the maximally activated filters (explainable AI), and giving feedback through single examples, which the AI could use to retrain (active learning). A key **open challenge** in this area is how to filter out common sense knowledge in the explanation.

Finally, in a *black-box NPC behavior* control system, we focused on agents controlled by deep neural networks (DNNs), especially deep Q-networks (DQNs). Two types of information stood out as of particular importance to be well communicated to game designers. First, given a particular gameplay context, what is the likely distribution of actions the NPC can take at this given moment. Second, given a particular NPC action, what are all the possible situations that can lead to this action. Given the large number of possible actions and/or situations, similar to highlights in white-box PCG systems, a good design guideline for XAID is to highlight the unexpected and reduce the visibility of the common. A key open challenge to provide both types of information to a human designer is how to design the reward function.

In conclusion, explainable AI for designers is crucial for advancements in AI to be fully utilized in computer games and other types of interactive experiences. Results from this XAID workshop show that the current understanding of how to communicate the underlying AI algorithms to human designers is still rudimentary. Although different types of AI algorithms place varying challenges and opportunities for the corresponding XAIDs, an emergent key challenge shared in all three cases we investigated is **salience**. That is, among the various types of information that could be provided about the underlying AI algorithms, how do we define, identify, and communicate what is noteworthy. Future work includes deeper understandings of salience grounded in the specific needs of designers, algorithmic investigations of how to procedurally identify salient features, as well as design innovation of how to communicate them to human designers.

References

1 John Charnley, Simon Colton, Maria Teresa Llano Rodriguez and Joseph Corneli. The FloWr Online Platform Automated Programming and Computational Creativity as a Service. In *Proceedings of the International Conference on Computational Creativity*, 2016

2 Simon Colton, Jakob Halskov, Dan Ventura, Ian Gouldstone, Michael Cook and Blanca Perez-Ferrer. The Painting Fool Sees! New Projects with the Automated Painter. In *Proceedings of the International Conference on Computational Creativity*, 2015.

3 François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. Automatic generation of textual summaries from neonatal intensive care data. In *Artificial Intelligence* 173, 7–8 (May 2009), 789–816. 2009.

4 Georgios N. Yannakakis, Antonios Liapis, Constantine Alexopoulos. Mixed-Initiative Co-Creativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014.

5 Panel discussions

5.1 Evaluation

Pieter Spronck (Tilburg University, NL)

At the end of the Seminar, we had a one-hour session with all attendees still present to evaluate the seminar. While the general consensus was that the Seminar had been a lot of fun and a great success, a desire was expressed to be a slightly less free-form for a future Seminar. In particular, the following points were brought up:

- Several attendees felt that they should have been primed more before arriving on the purpose of the Seminar and what was expected of them.

- To get more off-the-wall, crazy ideas, attendees should be instructed to think about and discuss them before getting to the Seminar. As it was, the "crazy" ideas started flowing only on the second day.
- Working groups could already be set up before arriving at the Seminar.
- Several attendees expressed an interest in having someone give an instructive workshop, for instance on a tool.
- It would be valuable if all working groups would report back in a structured form using sheets.

Furthermore, a discussion was held on the value of having an explicit code-of-conduct. Several arguments for and against having a code-of-conduct were brought up, but ultimately no conclusion was reached, apart from the fact that some attendees would really appreciate one as several conferences in the past (not Dagstuhl meetings) would have benefited from having one. In this respect, Dagstuhl may consider to write an explicit code-of-conduct for Dagstuhl Seminars in general.

Participants

- Elisabeth André
Universität Augsburg, DE
- Dan Ashlock
University of Guelph, CA
- Sander C.J. Bakkes
Tilburg University, NL
- Rafael Bidarra
TU Delft, NL
- Cameron Browne
RIKEN – Tokyo, JP
- Alex J. Champandard
AiGameDev.com KG – Wien, AT
- Simon Colton
Falmouth University, GB
- Michael Cook
University of London, GB
- Peter I. Cowling
University of York, GB
- Steve Dahlskog
Malmö University, SE
- Mirjam P. Eladhari
Södertörn University –
Huddinge, SE
- Reynald Francois
Ubisoft – Düsseldorf, DE
- Matthew J. Guzdial
Georgia Institute of Technology –
Atlanta, US
- Amy K. Hoover
NJIT – Newark, US
- Pier Luca Lanzi
Polytechnic University of
Milan, IT
- Marc Erich Latoschik
Universität Würzburg, DE
- Antonios Liapis
University of Malta – Msida, MT
- Jialin Liu
Queen Mary University of
London, GB
- Simon M. Lucas
Queen Mary University of
London, GB
- Joshua Allen McCoy
University of California –
Davis, US
- Alexander Nareyek
Singapore, SG
- Mark J. Nelson
Falmouth University, GB
- Ana Paiva
INESC-ID – Porto Salvo, PT
- Diego Perez Liebana
University of Essex –
Colchester, GB
- Mike Preuß
Universität Münster, DE
- Sebastian Risi
IT University of
Copenhagen, DK
- Spyridon Samothrakis
University of Essex –
Colchester, GB
- Jacob Schrum
Southwestern University –
Georgetown, US
- Emily Short
Oxford, GB
- Adam M. Smith
University of California –
Santa Cruz, US
- Gillian Smith
Worcester Polytechnic
Institute, US
- Pieter Spronck
Tilburg University, NL
- Anne Sullivan
University of Central Florida –
Orlando, US
- Shoshannah Tekofsky
Spirit AI – London, GB
- Tommy Thompson
Anglia Ruskin University –
Cambridge, GB
- David Thue
Reykjavik University, IS
- Julian Togelius
New York University, US
- Michael Treanor
American University –
Washington, US
- Vanessa Volz
TU Dortmund, DE
- Georgios N. Yannakakis
University of Malta – Msida, MT
- R. Michael Young
University of Utah –
Salt Lake City, US
- G. Michael Youngblood
Xerox PARC – Palo Alto, US
- Jichen Zhu
Drexel Univ. – Philadelphia, US

Report from Dagstuhl Seminar 17472

Addressing the Computational Challenges of Personalized Medicine

Edited by

Niko Beerenwinkel[1], Holger Fröhlich[2], and Susan A. Murphy[3]

1 **ETH Zürich, Department of Biosystems Science and Engineering, Mattenstrasse 26, 4058 Basel, Switzerland**
`niko.beerenwinkel@bsse.ethz.ch`

2 **UCB Biosciences GmbH, Alfred-Nobel-Str. 10, 40789 Monheim, Germany and University of Bonn, Bonn-Aachen International Center for IT (b-it), Dahlmannstr. 2, 53127 Bonn, Germany**
`holger.froehlich@ucb.com; frohlich@bit.uni-bonn.de`

3 **Harvard University, Department of Statistics, Science Center 400 Suite, One Oxford Street, Cambridge, MA 02138-2901, USA**
`samurphy@fas.harvard.edu`

Abstract

This report provides an overview of the talks and the working group reports from the Dagstuhl Seminar 17472 "Addressing the Computational Challenges of Personalized Medicine". The seminar brought together leading computational scientists with different backgrounds and perspectives in order to allow for a cross-fertilizing and stimulating discussion. It thus joined expertise that is usually scattered in different research communities. In addition, selected medical researchers, pharmacogenomics researchers and behavioral scientists provided their input and established the link of the computational to the more medical aspects of personalized medicine (PM). The talks and corresponding discussion spanned mainly three areas: 1) how to enhance prediction performance of computational models for PM; 2) how to improve their interpretability; 3) how to validate and implement them in practice.

Seminar November 19–22, 2017 – http://www.dagstuhl.de/17472

1998 ACM Subject Classification I.2.6 Learning, I.6 Simulation and Modeling, J.3 Life and Medical Sciences

Keywords and phrases data science, machine learning, computational modeling, bioinformatics, systems biology

Digital Object Identifier 10.4230/DagRep.7.11.130

1 Executive Summary

Niko Beerenwinkel
Holger Fröhlich
Susan Murphy

Personalized medicine (PM) is understood as a non-traditional medical approach, in which patients are stratified based on their disease subtype, disease risk, disease prognosis or treatment response using specialized diagnostic tests. High promises for the whole health care sector are associated with PM, and correspondingly the topic has received a lot of

Addressing the Computational Challenges of Personalized Medicine, *Dagstuhl Reports*, Vol. 7, Issue 11, pp. 130–141
Editors: Niko Beerenwinkel, Holger Fröhlich, and Susan A. Murphy
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

attention during the last years. PM is tightly connected to and dependent on computational sciences (computer science, mathematical modeling, computational statistics, bioinformatics). Currently, shortcomings of computational methodology constitute an important bottleneck for PM, which hinders full realization.

The goal of the planned seminar was to bring together an international and interdisciplinary group of experts in different computational science disciplines in order to discuss, how some of the major existing computational challenges could be better addressed in the future, namely:

1. how to enhance prediction performance of computational models for PM
2. how to improve their interpretability
3. how to validate and implement them in practice

The seminar joined together expertise that is usually scattered across different disciplines. The seminar had a strict focus on computational methodology, but few selected non-computational scientists closed the gap to the application field.

2 Table of Contents

Executive Summary
Niko Beerenwinkel, Holger Fröhlich, and Susan Murphy 130

Overview of Talks

Bayesian matrix factorization with side information
Yves Moreau 133

Dynamic Patient Restratification Using Mobile Sensors
Kumar Santosh 133

Causality
Marloes Maathuis 134

Hybrid models – combining mechanistic and statistical modeling approaches
Andreas Schuppert 134

Visualizing and Integrating Biological Knowledge
Rudi Balling 135

Computational Analysis of Viral Drug Resistance
Thomas Lengauer 135

Enhanced translation of multi-modal stratification models, as a basis for Precision Medicine
Michael Rebhan 136

Discussed Challenges and Possible Approaches 137

Conclusions 140

Participants 141

3 Overview of Talks

3.1 Bayesian matrix factorization with side information

Yves Moreau (University of Leuven – Leuven, Belgium, moreau@esat.kuleuven.be)

Matrix factorization/completion methods provide an attractive framework to handle sparsely observed data, also called "scarce" data. A typical setting for scarce data are is clinical diagnosis in a real-world setting. Not all possible symptoms (phenotype/biomarker/etc.) will have been checked for every patient. Deciding which symptom to check based on the already available information is at the heart of the diagnostic process. If genetic information about the patient is also available, it can serve as side information (covariates) to predict symptoms (phenotypes) for this patient. While a classification/regression setting is appropriate for this problem, it will typically ignore the dependencies between different tasks (i.e., symptoms). We have recently focused on a problem sharing many similarities with the diagnostic task: the prediction of biological activity of chemical compounds against drug targets, where only 0.1% to 1% of all compound-target pairs are measured. Matrix factorization searches for latent representations of compounds and targets that allow an optimal reconstruction of the observed measurements. These methods can be further combined with linear regression models to create multitask prediction models. In our case, fingerprints of chemical compounds are used as "side information" to predict target activity. By contrast with classical Quantitative Structure-Activity Relationship (QSAR) models, matrix factorization with side information naturally accommodates the multitask character of compound-target activity prediction. This methodology can be further extended to a fully Bayesian setting to handle uncertainty optimally, which is of great value in this pharmaceutical setting where experiments are costly. We have developed a significant innovation in this setting, which consists in the reformulation of the Gibbs sampler for the Markov Chain Monte Carlo Bayesian inference of the multilinear model of matrix factorization with side information. This reformulation shows that executing the Gibbs sampler only requires performing a sequence of linear regressions with a specific noise injection scheme. This reformulation thus allows scaling up this MCMC scheme to millions of compounds, thousands of targets, and tens of millions of measurements, as demonstrated on a large industrial data set from a pharmaceutical company. We have implemented our method as an open source Python/C++ library, called Macau, which can be applied to many modeling tasks, well beyond our original pharmaceutical setting. https://github.com/jaak-s/macau/tree/master/python/macau.

3.2 Dynamic Patient Restratification Using Mobile Sensors

Kumar Santosh (University of Memphis – Memphis, USA, santosh.kumar@memphis.edu)

Recent advances in wearable sensing and mobile computing have opened up unprecedented opportunities to quantify dynamic changes in an individual's health state as well as key physical, biological, behavioral, social, and environmental factors that contribute to health and disease risk, anytime and anywhere. For example, smart watches can not only track physical activity, but they can also be used to monitor stress (from pulse rate), eating, brushing, driving, and smoking behaviors (from hand gestures). By simultaneous monitoring of changes in health status, exposures to surrounding geographical, environmental, visual, social, and digital worlds, and personal behaviors (both risky and healthy), mobile health (mHealth) can help discover new predictors of health outcomes. By monitoring the exposure to these health risk predictors, mobile health offers an opportunity to introduce temporal precision in precision medicine, especially when mHealth data is used together with traditional sources of biomedical data (e.g., genomics, clinical). Longitudinal nature of mHealth data and the fact that it comes from the natural free-living environment allows dynamic decision making such as adapting the treatments and interventions so as to maximize the efficacy and optimize the timing of delivery. Continuous monitoring of the context surrounding the individual and monitoring of the compliance and response to treatments and interventions offers additional opportunities for dynamic optimizations in a human-in-the-loop model. Realizing these potential presents a rich multi-disciplinary research agenda. It includes sensor design and mobile system design for optimizing data collection with minimum user burden, mobile sensor big data modeling to convert voluminous mobile sensor data into actionable information, sensor-triggered intervention modeling that leverages dynamic optimization opportunities to discover the most efficacious and temporally-precise treatments and interventions, and engaging visualizations to encourage health and wellness-supporting daily behaviors using new insights gained from mHealth data.

3.3 Causality

Marloes Maathuis (ETH Zürich – Zürich, Switzerland, maathuis@stat.math.ethz.ch)

Causal questions are fundamental in all parts of science. Answering such questions from non-experimental data is notoriously difficult, but there has been a lot of recent interest and progress in this field. I have discussed current approaches to this problem and have outlined their potential as well as their limitations.

3.4 Hybrid models – combining mechanistic and statistical modeling approaches

Andreas Schuppert (RWTH Aachen – Aachen, Germany, schuppert@aices.rwth-aachen.de)

Modeling for personalized medicine requires methods enabling to predict reliably the evolution of the diseases, the response on therapies as well as the therapeutic adverse side effects for individual patients. However, due to a lack of understanding of the broad range of mechanisms affecting diseases and therapies, pure mechanistic modeling rarely results in satisfactory precision. On the other side, pure machine learning – based modeling methods are hampered by their conceptually high data demand for model training and their lack of extrapolation. In patient populations, the intrinsic mutual control loops inside the system “patient” in combination with the high variety of optional covariates result in statistically poor, biased distributions of data in high dimensional data spaces, hampering machine learning even in large "real world evidence" data sets. Hence, a combination of mechanistic and machine learning in a hybrid model is required in order to achieve the necessary precision of the models. Hybrid modeling had been developed for chemical and biotechnological engineering in order to tackle the lack of process data, combined with common lack of quantitative understanding of the reaction kinetics . The mathematical basis of data representation by means of hybrid models goes back to Hilbert's famous 13th problem and has been intensively discussed by Kolmogoroff, Arnold and Vitushkin . Later it could be shown that the knowledge of the true system structure without any mechanistic knowledge is sufficient to break the curse of dimensionality, to reduce the data demand for model training and to enable extrapolability of the models . The inverse problem, namely the identification of model structures from data, has recently been discussed in the context of systems biology . These results apparently have a strong relationship to the current development of deep learning technologies. We expect that a future integrative technology might result in a modeling platform satisfying the requirements of personalized medicine.

3.5 Visualizing and Integrating Biological Knowledge

Rudi Balling (University of Luxemburg – Luxemburg, rudi.balling@uni.lu)

The LCSB is engaged in a number of community efforts to develop novel tools for the visualization, annotation and integration of network-encoded knowledge in biomedical research. In order to capture the rapidly increasing information and inter-relationships between different factors contributing to Parkinson's disease (PD), we have established a "PD-map". This map is a manually curated knowledge repository and serves as a computationally tractable representation of all known molecular interactions involved in the pathogenesis of Parkinson's disease. The disease map offers research-facilitating functionalities such as the overlay of experimental data and the identification of drug targets on the map. A major effort is also geared towards the development of genome-scale human and human gut metabolic reconstructions integrating the full spectrum of metabolic and transport reactions that can occur in a given organism. The goal is to develop a comprehensive knowledge base of human metabolism integrating pharmacogenetic associations, large-scale phenotypic data and structural information for proteins and metabolites.

3.6 Computational Analysis of Viral Drug Resistance

Thomas Lengauer (Max-Planck Institute for Informatics – Saarbrücken, Germany, lengauer@mpi-inf.mpg.de)

We present a concrete case of translational research in computational biology in 4 steps. The problem is to estimate the resistance to HIV to individual drug based on viral genotype and to combinatorial therapies with respect to their estimated effectiveness.

1. gene2pheno[resistance] estimates the level of resistance of HIV to individual inhibitors of viral protease and reverse transcriptase. The software has first been online 15 years ago. It is used in clinical practice, but has strong competition from expert rule systems. The software interprets its predictions in terms of effect of individual mutations in the HIV genome.
2. gene2pheno[coreceptor] Our "blockbuster" estimates viral coreceptor usage It provides a significance estimate. There is no competition from rule based systems.
3. gene2pheno[THEO] ranks combinations of drug therapies w.r.t. estimated effectiveness. In contrast to the previous tools this one has not entered clinical routine, partially because the predictions are not interpreted.
4. g2p2 is our new development that is aimed at bringing therapy predictions to clinical routine. It merges mathematical analysis with traditional schemes of therapy composition and is interactive.

3.7 Enhanced translation of multi-modal stratification models, as a basis for Precision Medicine

Michael Rebhan (Novartis AG – Basel, Switzerland, michael.rebhan@novartis.com)

Progress in Precision Medicine and Personalized Health is linked to our ability to translate increasingly complex 'multi-modal stratification' models from discovery to validation, and finally to real world healthcare settings where they can generate impact on patient outcomes. Such models need to be able to computationally deal with a diversity of signals from an increasing number of 'channels' that can influence stratification, including those derived from molecular biomarkers, imaging technology, and 'digital biomarkers', to name a few. Such models would, down the road, help us predict not only the best intervention for a particular patient, but also the best time and context for delivering it, considering disease progression knowledge, patient needs and priorities, and different healthcare settings. In this session, we will discuss the idea of co-designing an open innovation ecosystem for community-based learning on such models, 'on top' of the current health data silos. As there are many challenges on the translational path for these models, we will discuss potential solutions to explore as a community. How to best conduct high quality clinical validation studies that can help to bridge the gap between early research and responsible first use of multi-modal stratification models in clinical decision making? How could outcome-based feedback loops help with community-based learning, beyond the clinical institutions involved in patient care? How can open learning 'on top of the data silos' look like, in practice? As we discuss those challenges, we will try to consider the full complexity of the health innovation landscape with its many stakeholders (patients, physicians, payers, basic / applied researchers, regulators etc.), and real life challenges, as this will help us co-design meaningful translational paths. In addition, we will discuss guiding principles that can help with the community build, e.g. transparency (of data and algorithms), and their role in such an effort.

4 Discussed Challenges and Possible Approaches

Author: Niko Beerenwinkel, Holger Fröhlich, and Susan Murphy

The following comprises a summary of the problems and possible approaches that were discussed in different working groups and within the panel.

Enhancing Prediction Performance

Performance Metrics

It is necessary to consider performance metrics apart from the established area under ROC curve (AUC). The choice of performance metric should depend on the actual prediction problem at hand.

Data Quality and Systematic Biases

Data quality is one of the reasons behind low prediction performance. Data quality is a continuous concern, specifically with respect to omics data. Robust loss functions in machine learning methods should be considered.

Prediction performance is also affected, if the data represents unknown mixtures of different biological origin. For example, tumor biopsies often contain a mixture of actual tumor and stroma cells, which impacts measured gene expression. A possible approach is to de-convolute the original data via mixture (regression) models. At this point a Gaussian assumption for transcriptomics data seems feasible.

It is known that independently collected patient cohorts exhibit a systematic difference in their expression profiles to the original training cohort. The recently introduced zero-sum regression approach is a way to address this issue [1].

Feature Engineering and Extraction

Feature engineering remains a crucial topic for successful modeling, because it allows for using prior knowledge. Such prior knowledge could also come from similar data, which has been collected for different purposes.

In addition to feature engineering, extraction of (latent) meta-features is likely to be a successful strategy. Methods include matrix factorization techniques as well as auto-encoder networks.

Use of Multi-Modal Data

Multi-modal, longitudinal data is widely believed to provide a more detailed view on the complex relationship between biology and clinical outcome, which we try to capture with models in personalized medicine. Multi-modal patient trajectories are possibly embedded into a lower dimensional latent space, in which patterns become more obvious than in the original space. Matrix factorization approaches might be one way to identify a suitable latent space.

Despite of a multitude of available methods multi-modal data integration remains a challenge, specifically when fusion of static (e.g. genomic) and longitudinal data (e.g. clinical features) is desired. In the data science literature early, intermediate and late integration schemes are discussed, which have all their advantages and disadvantages. The optimal data fusion strategy is always data dependent and has thus to be found empirically.

In the future, data retrieved from web mining (patient blogs, social media) could play an increasing role. It might be possible to use these data within a Bayesian learning scheme, e.g. to define informative priors.

Improving Interpretability

Disease Maps

Disease maps describe cause-effect relationships between multi-modal molecular and clinical data entities. Disease maps are not computational models per se, but could be used in two different ways to obtain better interpretable prediction models:

- Post-hoc analysis of features in the model, e.g. via enrichment analysis and variants thereof.
- Embedding of network information into feature selection.

Both approaches are established in principle, but may require further adaptation to a specific problem, e.g. by defining subsets of the disease map, or extracting and engineering of appropriate features based on data.

Disease maps could also help informing causal network inference (see next paragraph).

Causal Models

Predictive models are often highly complex and not necessarily causal, which hinders the acceptance by physicians and limits scientific insights into the underlying pathophysiological mechanisms. Judea Pearl has established a widely accepted theory of causality in the context of probabilistic graphical models [2]. However, the graph structure of causal models can in general only be identified from observational data up to equivalence classes. Nonetheless, it is possible to predict bounds of intervention effects from purely observational data under certain conditions [3]. There is a need to better evaluate these methods in the context of personalized medicine. It has to be checked, how reproducible the results are and whether causal network models could inform data collection (when and what to collect) in the future. Moreover, the exact conditions under which for an individual prediction of causal intervention effects are possible should be clarified.

Hybrid Models

Causal models are not necessarily pointing towards detailed biological mechanisms. On the other hand fully mechanistic models are limited by the available background knowledge, which is often incomplete. Hybrid models combine partially available quantitative mechanistic and machine learning models into one unified framework. Within that framework machine learning “black-boxes” detect and correct errors in the mechanistic part. Black-box and mechanistic models can be integrated into a hybrid network and trained via mathematical optimization methods. Hybrid models currently have a theory gap. Nonetheless, the approach has been used successfully by Andreas Schuppert and colleagues for predicting drug response in diabetes I (unpublished work): This was possible, because there is a mechanistic model for diabetes I available. Behavioral aspects (eating, exercise, etc), which are also important for the disease, can be put into the black-box model part. There is much less known about diabetes II and consequently, there is no mechanistic model.

Critical Transitions

The development of many diseases may be interpreted from the perspective of a phase transition in a dynamical system. In physics this phenomenon is well known and appears in many models. Reliable detection of phase transitions in disease development could ultimately help to detect mechanistic biomarkers in the future and enable early disease diagnosis and prevention. There are hints in the literature that phase transitions might be detectable from data via simple statistical methods, such as variability and correlation. However, more evaluation is needed to demonstrate the actual utility in the context of personalized medicine.

Enhancing Transition to Clinical Practice

Better Transparency and Interpretability of Models

The lack of interpretability and transparency is one of the key obstacles that hinders acceptance of machine learning models by physicians and regulatory agencies. The pure focus on prediction performance is misleading. Additional measures such as stability, enrichment of existing knowledge and cross-study applicability should be considered. Moreover, there is the need to link model predictions with a "narrative" that can be understood by doctors and patients. Such a narrative may be generated in different ways:

- by visualizing molecular features that drive the prediction, possibly also with the help of disease maps
- by showing different decision alternatives together with their confidences whenever possible
- by showing and visualizing close patients from the training data (w.r.t. some metric)
- by generating a medical report for each patient in an automated fashion
- by linking latent model features back to biological knowledge

The latter point will require further research, but one possibility might be to look for evolutionary conserved disease modules. This could at the same time open the door to better utilize animal models.

Continuous Updating of Prediction Algorithms

There is a need for a clear and transparent process for continual iteration/updating and revalidation for precision medicine software tools. The notion of CLIA (Clinical Laboratory Improvement Amendments) labs provide a template for how health-related software tools (diagnosis, prediction, decision support) developed to inform precision medicine can be validated and re-validated in an clear, transparent manner as the tool is continually updated. CLIA labs are certified labs that go through a process of regular re-certification and monitoring by FDA and other regulatory agencies in the US. These labs follow a SOP, e.g. an approved, transparent and documentation process. Currently CLIA labs monitor medical devices, which can include software diagnosis tools. When a CLIA lab waives certification the tool can be used in practice. Most importantly the developer of the tool can update the software tool. However the CLIA labs are independent and decide when they will re-validate a software tool (so maybe not each time you upload a new version).

5 Conclusions

The current hype around AI and machine learning has to be contrasted with the reality, in which we are facing a number of challenges in the context of personalized medicine. These range from insufficient prediction performance over lack of model interpretability up to difficulties to engage people moving further on to clinical practice with a given model.

The current machine learning hype is dangerous, because it rises inappropriate expectations. In order to manage these expectations there is the strong need to better inform physicians about the current opportunities, challenges and future potential of data science in medicine. We therefore plan to publish a paper in a medical journal focusing on the key learnings from this Dagstuhl seminar.

The overall vision formulated in the seminar was to enable a causal treatment of patients with the right drug at the right time. We see a number of intermediate steps towards this grand vision:

- high dimensional causal graphical models
- hybrid models
- understanding of critical transitions
- better use of the principles of evolution, e.g. by looking for evolutionary conserved disease modules

All of these steps are computational. This underlines the crucial relevance of computational models for enabling personalized medicine.

References

1 M. Altenbuchinger, T. Rehberg, H. U. Zacharias, F. Stämmler, K. Dettmer, D. Weber, A. Hiergeist, A. Gessner, E. Holler, P. J. Oefner, R. Spang, Reference point insensitive molecular data analysis, Bioinformatics, 33(2): 219 – 226, 2017

2 J. Pearl, Causality: Models, Reasoning and Inference, Cambridge University Press, 2000

3 M. Maathuis, D. Colombo, M. Kalisch, P. Bühlmann, Predicting causal effects in large-scale systems from observational data, Nat. Methods, 7(4): 247 – 48, 2009

Participants

- Rudi Balling
University of Luxembourg, LU
- Niko Beerenwinkel
ETH Zürich – Basel, CH
- Holger Fröhlich
UCB Biosciences GmbH – Monheim, DE
- Oliver Kohlbacher
Universität Tübingen, DE
- Santosh Kumar
University of Memphis, US
- Thomas Lengauer
MPI für Informatik – Saarbrücken, DE
- Marloes Maathuis
ETH Zürich, CH
- Yves Moreau
KU Leuven, BE
- Susan A. Murphy
Harvard University – Cambridge, US
- Teresa Przytycka
National Center for Biotechnology – Bethesda, US
- Michael Rebhan
Novartis AG – Basel, CH
- Hannes Röst
University of Toronto, CA
- Andreas Schuppert
RWTH Aachen, DE
- Matthias Schwab
IKP – Stuttgart, DE
- Rainer Spang
Universität Regensburg, DE
- Daniel Stekhoven
ETH Zürich, CH
- Jimeng Sun
Georgia Institute of Technology – Atlanta, US
- Andreas Weber
Universität Bonn, DE
- Daniel Ziemek
Pfizer GmbH – Berlin, DE
- Blaz Zupan
University of Ljubljana, SI

Report from Dagstuhl Seminar 17481

Reliable Computation and Complexity on the Reals

Edited by

Norbert T. Müller[1], Siegfried M. Rump[2], Klaus Weihrauch[3], and Martin Ziegler[4]

1 **Universität Trier, DE,** mueller@uni-trier.de
2 **TU Hamburg-Harburg, DE,** rump@tuhh.de
3 **FernUniversität in Hagen, DE,** klaus.weihrauch@fernuni-hagen.de
4 **KAIST – Daejeon, KR,** ziegler@cs.kaist.ac.kr

Abstract

Naïve computations with real numbers on computers may cause serious errors. In traditional numerical computation these errors are often neglected or, more seriously, not identified. Two approaches attack this problem and investigate its background, Reliable Computing and Computable Analysis.

Methods in Reliable Computing are essentially mathematical theorems, the assumptions of which are verified on the computer. This verification is performed using the very efficient floating point arithmetic. If the verification succeeds, the assertions are true and correct error bounds have been computed; if not, a corresponding message is given. Thus the results are always mathematically correct. A specific advantage of Reliable Computing is that imprecise data are accepted; the challenge is to develop mathematical theorems the assumptions of which can be verified effectively in floating-point and to produce narrow bounds for the solution.

Computable Analysis extends the traditional theory of computability on countable sets to the real numbers and more general spaces by refining continuity to computability. Numerous even basic and simple problems are not computable since they cannot be solved continuously. In many cases computability can be refined to computational complexity which is the time or space a Turing machine needs to compute a result with given precision. By treating precision as a parameter, this goes far beyond the restrictions of double precision arithmetic used in Reliable computing. For practical purposes, however, the asymptotic results from complexity theory must be refined. Software libraries provide efficient implementations for exact real computations.

Both approaches are established theories with numerous important results. However, despite of their obvious close relations these two areas are developing almost independently. For exploring possibilities of closer contact we have invited experts from both areas to this seminar. For improving the mutual understanding some tutorial-like talks have been included in the program. As a result of the seminar it can be stated that interesting joint research is possible.

Seminar November 26–1, 2017 – http://www.dagstuhl.de/17481

1998 ACM Subject Classification F.2.1 Numerical Algorithms and Problems, F.1.3 Complexity Measures and Classes, G.4 Mathematical Software, F.4.1 Mathematical Logic

Keywords and phrases Computable Analysis, Verification Methods, Real Complexity Theory, Reliable Computing

Digital Object Identifier 10.4230/DagRep.7.11.142

Reliable Computation and Complexity on the Reals, *Dagstuhl Reports*, Vol. 7, Issue 11, pp. 142–167
Editors: Norbert T. Müller, Siegfried M. Rump, Klaus Weihrauch, and Martin Ziegler
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Norbert T. Müller
Siegfried M. Rump
Klaus Weihrauch
Martin Ziegler

The seminar was a meeting between two groups of researchers working in the related areas of reliable computing and of computational complexity on real numbers. While the first area originates in numerical analysis, the second area goes back to the roots of computer science and and computability.

Reliable computations aims to produce correct answers to numerical problems with mathematical rigor. This includes to prove that the problem is solvable and to compute mathematically correct error bounds for the solution. Reliable numerical computations solely use floating-point arithmetic to take advantage of the tremendous speed. Naturally that poses limits on the problems which can be solved, in particular the condition number. However, in contrast to purely numerical methods, no false answers are possible: Either a true error bound is computed or, a corresponding error message is given. There is a history of reliable numerical computations. In the early days, interval arithmetic was often used in a rather naive way. Still the computed results were correct, however, often wide or no bounds at all were computed. Meanwhile it is well understood how to derive effective methods for reliable numerical computations, avoiding wide bounds and pushing the set of solved problems to the limit of that of purely numerical algorithms. A number of interesting and hard mathematical problems have been solved using reliable numerical computations. This includes the famous Kepler conjecture, the existence of mutually distinct solutions to certain partial differential equations, and more. Needless to say that solving a mathematical problem requires rigorous solutions of all particular problems.

Computable analysis is a branch of computability theory studying those functions on the real numbers and related structures which can be computed by machines such as digital computers. The increasing demand for reliable software in scientific computation and engineering requires a sound and broad foundation not only of the analytical/ numerical but also of the computational aspects of real number computation. The branch of computable analysis based on the definition by Grzegorczyk and Lacombe of computable real functions (TTE, “Type 2 Theory of Effectivity”) has turned out to be particularly useful for investigating computability on uncountable sets. As a central concept computability appears as a specialization of continuity. Meanwhile computability of numerous analytic problems has been investigated (from basic analysis, functional analysis, ordinary and partial differential equations, analytic functions, measure theory, dynamical systems etc.). All these examples demonstrate the usefulness of the concept.

Once a problem has been shown computable, a natural next question asks for the computational efficiency of such a solution. This is where real analysis meets (discrete) complexity theory with notions of runtime and memory/space: asymptotically with respect to $n \to \infty$ for approximating the output up to absolute error 2^{-n}. The famous Bailey-Borwein-Plouffe method for instance permits to compute billions of digits of transcendental within minutes; while Bloch’s constant, although proven computable, is still not known up to error 2^{-5}. In fact the distinction between polynomial and exponential time, in the discrete realm gauged for instance by complexity classes P, NP, #P, and PSPACE, re-emerges in

the real case: The bit-cost of computing the maximum of an arbitrary fixed smooth (i.e. infinitely often differentiable) polynomial-time computable $f : [0; 1] \to [0; 1]$ has been shown to correspond to P-vs-NP; that of Riemann integration to #P; and that of solving an ordinary differential equation to PSPACE. On analytic functions on the other hand these operations map polynomial-time computable instances back to polynomial-time computable results.

For practical purposes and in the spirit of "algorithm engineering", the asymptotic results from complexity theory have to be refined by considering the efficiency of actual implementations. Corresponding software libraries are usually called "exact real arithmetic" (ERA) and implement real numbers in the sense of TTE. ERA implementations exist in many languages, like C, C++ JAVA, Haskell or OCaml. Internally, ERA has to perform operations on infinite data like $\{0, 1\}^\omega$. The user interface, however, hides the details and offers operations and functions on "exact" real numbers. In consequence, users do not need to care about aspects like rounding or truncation errors or the specification of precisions. Instead, they can concentrate on the mathematical part of the problem under consideration. As computable real functions have to be continuous, it is impossible to implement some widely used real functions (like testing on equality). In consequence, ERA cannot simply copy the double precision interface one-to-one, but needs to go its own ways. Additionally, for the reason of efficiency the representations used in TTE have to be carefully revised. The resulting speed is comparable to the use of multiple precision floating point numbers, but now without any need for manual precision control.

2 Table of Contents

Executive Summary

Norbert T. Müller, Siegfried M. Rump, Klaus Weihrauch, and Martin Ziegler . . . 143

Overview of Talks

Reliable Visual Analytics within a Verification and Validation Management
Ekaterina Auer and Wolfram Luther 148

Orbital stability investigations for travelling waves in a nonlinearly supported beam
M. Plum, B. Breuer, J. Horak, K. Nagatou, and P. J. McKenna 148

Command-like Expressions for Real Infinite-precision Calculations
Andrej Bauer, Sewon Park, and Simpson, Alex (University of Ljubljana, SI) 149

Bounds for eigenvalues of an eigenvalue problem with non-smooth coefficients
Henning Behnke 150

Computing positive invariant sets with intervals
Benoît Zerr, Luc Jaulin, and Thomas Le Mézo 150

On the Taylor model approach for solving ODEs
Florian Bünger 150

Uncertainty Quantification: Probabilistic Forecasts of Energy Demand
George F. Corliss 151

Nonlinear Symbolic Transformations for Simplifying Functions – Applied for Interval Based Global Optimization
Tibor Csendes and Elvira Dobjánné Antal 151

Daisy – a framework for sound accuracy analysis and optimization of finite-precision programs
Eva Darulova 152

Computability of geometric Lorenz attractors
Daniel Graça 152

Interval computations with compensated algorithms
Stef Graillat 153

On the Computational Complexity of the Range Computation Problem
Peter Hertling 153

Formal Verification of a Rigorous ODE Solver
Fabian Immler 153

Towards certified exact real computation
Sunyoung Kim 154

Functional exact real computation, towards verification of total correctness
Michal Konecny 154

Computable Numbers, Computable Sets, and Computable Functions and How It Is All Related to Interval Computations
Vladik Kreinovich 154

Need to Combine Interval and Probabilistic Uncertainty: What Needs to Be Computed, What Can Be Computed, What Can Be Feasibly Computed, and How Physics Can Help
Vladik Kreinovich 155

An Approach to Programming Configurable Computers for Numeric Applications
Fritz Mayer-Lindenberg 157

On implementing TTE
Norbert T. Müller 157

On the numerical norm estimation of the inverse operator in Hilbert space
Mitsuhiro T. Nakao 158

Parametrised complexity for the naive Cauchy representation
Eike Neumann and Florian Steinberg 158

Test Matrices for Numerical Linear Algebra
Katsuhisa Ozaki 159

A tutorial on reliable numerical computation
Paul Zimmermann 159

Computational complexity of solving polynomial differential equations over unbounded domains
Amaury Pouly and Daniel Graça 160

Numerical (arte-)facts and reliable computing
Siegfried M. Rump 160

Co-Polish spaces in Complexity Theory
Matthias Schröder 160

Bit complexity of Computing Solutions for Symmetric Hyperbolic Systems of PDEs with Guaranteed Precision
Svetlana Selivanova 161

Parametrised second-order complexity theory with applications to the study of interval computation
Florian Steinberg and Eike Neumann 162

Verified computations for solutions to 1-dimensional advection equations with variable coefficients
Akitoshi Takayasu 163

Average case complexity for Hamiltonian dynamical systems
Holger Thies, Akitoshi Kawamura, and Martin Ziegler 163

Small divisors and normal forms
Warwick Tucker 164

Soft Foundations for Geometric Computation
Chee K. Yap 164

On Formal Verification in Imperative Multivalued Programming over Continuous Data Types
Martin Ziegler, Gyesik Lee, Norbert T. Müller, Eike Neumann, Sewon Park, and Norbert Preining 165

Tutorial on Computational Complexity in Analysis
Martin Ziegler and Akitoshi Kawamura 166

Participants 167

3 Overview of Talks

3.1 Reliable Visual Analytics within a Verification and Validation Management

Ekaterina Auer and Wolfram Luther

A recently described four tier verification and validation management (VVM) defines requirements for categorization and classification of processes as a result of precise assessment procedures and addresses recommending techniques, user interaction and collaboration via adequate human machine interfaces. Huge data and program code require new visual analysis methods. Reliable visual analytics is paired with an assessment of (meta)data and code quality, adequate data types and methods to propagate and bound uncertainty. In our talk, we present an ontology-based architecture with a query engine and modern human machine interaction and requirements from the VVM for various use cases: GPS sensing and localization, spatial decision making, risk communication and perception, analysis of steel samples using SILENOS©, virtual museums and labs toolbox ViMEDEAS, biomechanics: e.g., femur prosthesis (PROREOP) as well as modeling and implementation of a microscopic traffic simulation (OLSIMv4).

References

1 E. Auer, W. Luther, Numerical Verification Assessment in Computational Biomechanics in: Proc. of Dagstuhl Seminar 08021: Numerical Validation in Current Hardware Architectures, Lecture Notes in Computer Science 5492. Springer (2009) 145–160

2 B. Weyers, W. Luther, Risk Communication and Perception in Low- and High-Immersion Virtual Environments, APSSRA6, 28-30 May 2016, Shanghai, China, H.W. Huang, J. Li, J. Zhang & J. B. Chen (editors) (2016)

3 G. Rebner, D. Sacher, B. Weyers, W. Luther, Verified stochastic methods in geographic information system applications with uncertainty, Structural Safety 52 (2015) 244–259

4 D. Sacher: A generative approach to virtual museums using a new metadata format. A curators', visitors' and software engineers' perspective. Logos, Berlin (2017) ISBN 978-3-8325-4627-4, 288p.

5 M. Thurau, Chr. Buck, and W. Luther, IPFViewer: Incremental, approximate analysis of steel samples, in Proceedings of SIGRAD 2014, Linköping University Electronic Press (2014) 1–8

6 Publications and projects SCG: http://www.scg.inf.uni-due.de/

3.2 Orbital stability investigations for travelling waves in a nonlinearly supported beam

M. Plum, B. Breuer, J. Horak, K. Nagatou, and P. J. McKenna

For a nonlinear beam equation on the whole real line with exponential nonlinearity, we prove existence of at least 36 travelling wave solutions for the specific wave speed c=1.3. Our proof makes heavy use of computer assistance: starting from numerical approximations, we use a fixed point argument to prove existence of solutions “close to” the approximate ones. Moreover we investigate the orbital stability of these solutions via computation of their Morse indices, using classical theoretical results by Grillakis, Shatah, and Strauss. Also for these stability investigations we make use of both analytical and computer-assisted techniques.

3.3 Command-like Expressions for Real Infinite-precision Calculations

Andrej Bauer (University of Ljubljana, SI), Sewon Park (KAIST – Daejeon, KR), and Simpson, Alex (University of Ljubljana, SI)

We present an idealized programming language, *Clerical*, for exact real-number computation. Clerical is an imperative language with conditional statements, `while` loops and local mutable variables. Its primitive datatypes are the Booleans, the integers and the reals. Basic arithmetical operations and comparison tests are provided. As is usual and necessary, comparison operators on reals may be non-terminating.

Clerical supports Dijkstra’s guarded non-deterministic choice

$$\texttt{case}\ b_1 \Rightarrow c_1 \mid b_2 \Rightarrow c_2\ \texttt{end}$$

The intended meaning is that c_1 may execute if b_1 is true, and c_2 may execute if b_2 is true, and one of the branches will execute as long as b_1 or b_2 is true. When b_1 *and* b_2 hold, either branch may execute, which leads to *multivalued* computations that may return one of several possible results. The `case` construct is used to circumvent non-decidability of $<$ on the reals. In a typical application we perform an *approximate* test $x < y$ with precision ϵ by running

$$\texttt{case}\ x < y + \epsilon \Rightarrow c_1 \mid y < x + \epsilon \Rightarrow c_2\ \texttt{end}.$$

One of the tests will always succeed and the corresponding branch will be executed. When both tests succeed, Clerical may choose either branch.

In Clerical real numbers are constructed with a limit constructor

$$\texttt{lim}(n : \texttt{int}, e(n))$$

where $e(n)$ is real-valued.

The result is defined to be the limit of the sequence $e(0), e(1), e(2), \ldots$, assuming the sequence converges with a required rate of convergence for all possible values of the terms $e(n)$. For the limit operator to be useful, we must allow $e(n)$ to be arbitrarily complex code. However, in order to make sense of the convergence of the sequence, $e(n)$ should be free of side-effects. We are led to the idea of a *command-like expression*, a value-returning command that may modify only its own local state.

The value of a Clerical program is a non-empty set of possible values, including non-termination $\bot$. Each datatype is interpreted naively: the booleans take values from the set $\{\texttt{false}, \texttt{true}\}$, the integers from $\mathbb{Z}$, and the reals from $\mathbb{R}$. In particular, we need no domain theory or space representations, a familiar set-theoretic model does the job. In a related talk we present Hoare-style proof rules for deriving correctness of Clerical programs. The rules are sound with respect to the set-theoretic semantics.

We will showcase an implementation of Clerical that executes programs in the style of Norbert Müller’s iRRAM. It approximates real numbers with intervals, using interval arithmetic. When the intervals get too inaccurate for the comparison tests to succeed, the entire computation is aborted and restarted anew with better initial precision. Such an execution strategy is correct with respect to the semantics of programs in the sense that is always computes one of the possible values of the program.

3.4 Bounds for eigenvalues of an eigenvalue problem with non-smooth coefficients

Henning Behnke (TU Clausthal, DE)

In the design of integrated optical chips an eigenvalue problem with piecewise continuous functions arises. Typical features of the technological problem are the possible occurence of eigenvalue clusters and the necessity of quite stringent relative error tolerances for the eigenvalues. For the computation of bounds we use finite elements based on the Rayleigh-Ritz and Temple-Lehmann-Goerisch methods for upper and lower eigenvalue bounds, respectively. Rounding errors are controlled with interval arithmetic.

3.5 Computing positive invariant sets with intervals

Benoît Zerr, Luc Jaulin, and Thomas Le Mézo

Main reference Thomas Le Mezo, Luc Jaulin, Benoît Zerr: “An Interval Approach to Compute Invariant Sets”, IEEE Trans. Automat. Contr., Vol. 62(8), pp. 4236–4242, 2017.
URL http://dx.doi.org/10.1109/TAC.2017.2685241

Given a set X, and a dynamical deterministic system S, the largest invariant set inside X is the set of all x_0 such that all trajectories going through x_0 stay inside S for all time instant t. In this presentation, I will show how interval methods can be used to compute an inner and an outer characterization of an invariant set. The procedure that will be presented is fast, guaranteed and does not require any interval integration of the dynamics. The main idea is to use the new notion of maze, which is a composition of graphs, pavings of boxes and polygons. The set of mazes forms a lattice with respect to the inclusion so that contractor-based methods could be used.

3.6 On the Taylor model approach for solving ODEs

Florian Bünger (TU Hamburg-Harburg, DE)

Taylor models have been used successfully to calculate verified inclusions of the solutions of initial value problems for ordinary differential equations (ODEs).

Especially Berz, Makino, and their group focused on that and invented several accompanying methods like “shrink wrapping”, “blunting”, and “preconditioning”.

We give a short description of Taylor models, their arithmetic, the algorithm for solving ODEs and the aforementioned accompanying methods.

3.7 Uncertainty Quantification: Probabilistic Forecasts of Energy Demand

George F. Corliss (Marquette University – Milwaukee, US)

Joint work of Mohammad Saber and Richard Povinelli
Main reference Mohammad Saber: "Quantifying Forecast Uncertainty in the Energy Domain", Ph.D. Dissertation, Department of Electrical and Computer Engineering, Marquette University, Milwaukee, Wisc., 2017.
URL http://epublications.marquette.edu/dissertations_mu/746/

We forecast daily natural gas demand for utilities at time horizons up to a week. Intervals could capture 100% confidence intervals, but our customers prefer a cumulative density distribution (CDF) to capture a richer expression of uncertainty. The CDF $f(x)$ expresses the probability that the actual demand d will be less than or equal to amount x. Our preferred probabilistic forecasting engine uses historical point-valued forecasts from a linear regression or artificial neural network model and bins empirical residuals, e.g., by temperature. In each bin, a Johnson transformation maps the empirical residuals to a (nearly) normal distribution. To generate a probabilistic forecast, we generate a point forecast, which we use to mean-shift the normal distribution appropriate to the forecast temperature. We assess the quality of our forecasts using Saber's Graphical Calibration Measure: If we make probabilistic forecasts for a year, we have 365 forecasts $P(d <= x_p) = p$, for example, $p = 0.95$. If our forecasting engine is good, we should be right 95% of the time and wrong 5% of the time.

3.8 Nonlinear Symbolic Transformations for Simplifying Functions – Applied for Interval Based Global Optimization

Tibor Csendes (University of Szeged, HU) and Elvira Dobjánné Antal

Main reference Elvira Antal, Tibor Csendes: "Nonlinear Symbolic Transformations for Simplifying Optimization Problems", Acta Cybern., Vol. 22(4), pp. 715–733, 2016.
URL http://dx.doi.org/10.14232/actacyb.22.4.2016.1

For interval arithmetic based reliable computation the expression of the function to be optimized can be critical. We have an automatic algorithm based on symbolic calculation to simplify nonlinear functions. The talk will give detailed results on the effect of this presolving technique on the efficiency of an interval arithmetic based branch-and-bound algorithm. Although we still have a long way to go, the first numbers are encouraging [1].

References

1 Tibor Csendes and Elvira Dobjánné Antal: Nemlineáris szimbolikus transzformációk optimalizálási feladatokra (In Hungarian: Nonlinear symbolic transformations for optimization problems). Szigma 48(2017) 33-46

3.9 Daisy – a framework for sound accuracy analysis and optimization of finite-precision programs

Eva Darulova (MPI-SWS – Saarbrücken, DE)

Main reference Daisy – Framework for Analysis and Optimization of Numerical Programs (Tool Paper), TACAS 2018 (to appear)

Floating-point or fixed-point computations are an integral part of many embedded and scientific computing applications, as are the roundoff errors they introduce. They expose an interesting tradeoff between efficiency and accuracy: the more precision we choose, the closer the results will be to the ideal real arithmetic, but the more costly the computation becomes. Unfortunately, the unintuitive and complex nature of finite-precision arithmetic makes manual op- timization infeasible such that automated tool support is indispensable. This talk presents an overview of Daisy, a framework for sound accuracy analysis and optimization of finite-precision programs. We will provide a high-level view of its main features: roundoff error analysis as well as rewriting and mixed-precision optimization.

3.10 Computability of geometric Lorenz attractors

Daniel Graça (University of Algarve, PT)

Joint work of Graça, Daniel; Rojas, Cristobal; Zhong, Ning
Main reference D. S. Graça, C. Rojas, and N. Zhong: "Computing geometric Lorenz attractors with arbitrary precision", Trans. Amer. Math. Soc. 370, pp. 2955–2970, 2018.
URL https://doi.org/10.1090/tran/7228

In many applications, ranging from biology to physics, one is interested in knowing about the long term behavior of a given system. However, many of those systems have complex dynamics, making their analysis quite difficult through the exclusive use of analytical methods. With the introduction of high speed and affordable computers, a common approach is to use numerical simulations to obtain new information about a system under study. This numerical approach led to new important insights. For example, evidence that "strange attractors" like the Lorenz attractor can occur came from numerical experiments. However, it is also commonplace to use floating point arithmetic in those simulations, where real numbers are substituted by approximations having fixed finite precision. This is a source of rounding and truncation errors which, especially in the case of chaotic systems like the Lorenz attractor, can be greatly amplified along time. Therefore the accuracy of the results obtained in this manner can be put into question. In response to one of the 18 unsolved problems that the Fields medalist S. Smale suggested for the 21st century, W. Tucker proved that the Lorenz attractor exists, using a combination of normal form theory and rigorous numerics. To reach that conclusion, it is shown that the Lorenz system behaves like a geometric Lorenz model. In this talk we will digress over these results and also show that geometric Lorenz attractors are computable.

3.11 Interval computations with compensated algorithms

Stef Graillat (UPMC – Paris, FR)

In this talk, we will present some resultats on interval computations with compensated algorithms. Compensated algorithms consist in computing the rounding errors of individual operations and then adding them later on to the computed result. This makes it possible to increase the accuracy of the computed result efficiently. Computing the rounding error of an individual operation is possible through the use of a so-called error-free transformation (EFT). The EFT need a rounding to the nearest to be exact. In this talk, we will show that EFT are still robust with directed rounding and and that they can be used to perform interval computations. This is a joint work with Fabienne Jézéquel.

3.12 On the Computational Complexity of the Range Computation Problem

Peter Hertling (Universität der Bundeswehr – München, DE)

The following problem is one of the basic problems of interval computations: given a function $f(x_1, ..., x_n)$ of n real variables, given n intervals, and given some desired output precision, compute the range of f over the box of these intervals at least with the desired precision. Gaganov (1981, 1985) considered the case where the input function f is a polynomial given by its coefficients and showed that this problem is at least as hard as any NP-problem. Kreinovich, Lakeyev, Rohn, and Kahl (1998) analysed the computational complexity of many further variants of this problem. First we show that the general problem is not harder than NP-problems. Then we consider some variants where a sequence of polynomials and a sequence of interval boxes are fixed and show that their complexity is closely connected to some other well-known open questions from structural complexity theory.

3.13 Formal Verification of a Rigorous ODE Solver

Fabian Immler (TU München, DE)

Main reference Fabian Immler: "A verified ODE solver and the Lorenz attractor", Journal of Automated Reasoning, Springer, 2018.
URL https://doi.org/10.1007/s10817-017-9448-y

This presents a formalization of ordinary differential equations (ODEs) and the verification of rigorous (with guaranteed error bounds) numerical algorithms in the interactive theorem prover Isabelle/HOL. The formalization comprises flow and Poincare map of dynamical systems. The verified algorithms are based on Runge-Kutta methods and affine arithmetic. They certify numerical bounds for the Lorenz attractor and thereby lift the numerical part of Tucker's proof of Smale's 14th problem onto a formal foundation.

3.14 Towards certified exact real computation

Sunyoung Kim (Ewha Womans University, KR)

When we use computers to do computing with real numbers, it is likely that we cannot guarantee the correctness of the results. In this talk, we give an overview of our recent project about certified exact real arithmetic. The main goal of our project is to develop and extend verified libraries for exact real arithmetic.

3.15 Functional exact real computation, towards verification of total correctness

Michal Konecny (Aston University – Birmingham, GB)

I have reported on the status of my attempts to develop formally verified numerical programs, taking advantage of the (relative) simplicity of the semantics of exact real computation and functional programming. In my implementation, I use a cut-down version of a functional version of the CIDR language which is currently being developed jointly with Brausse, Collins, Mueller, Neumann, Park and Ziegler in the CID project which is also similar to the Clerical language being developed by Bauer, Park and Simpson and the ERA language used in (Lee et al 2017). The programming language is deeply embedded in Isabelle and is given formal operational and non-deterministic denotations semantics. As the semantics works with a given resource limit, it is possible to formally specify program termination. The approach has been validated by formally verifying in Isabelle the functional correctness and termination of a program that computes the square root by Newton iteration.

References

1 Gyesik Lee, Norbert Müller, Eike Neumann, Sewon Park, Norbert Preining, Martin Ziegler *On Formal Verification in Imperative Multivalued Programming over Continuous Data Types*. arXiv:1608.05787 [cs.NA], 2017

3.16 Computable Numbers, Computable Sets, and Computable Functions and How It Is All Related to Interval Computations

Vladik Kreinovich (University of Texas – El Paso, US)

From the physical viewpoint, real numbers x describe values of different quantities. We get values of real numbers by measurements. Measurements are never 100% accurate, so after a measurement, we get an approximate value r_k of x. In principle, we can measure x with higher and higher accuracy. So, from the computational viewpoint, a real number is a sequence of rational numbers r_k for which, e.g., $|x - r_k| \leq 2^{-k}$. By an algorithm processing real numbers, we mean an algorithm using r_k as an "oracle" (subroutine). This is how computations with real numbers are defined in *computable analysis*.

Once we know the measurement result $\widetilde{x}$ and the upper bound Δ on the measurement error $\Delta x \overset{\text{def}}{=} \widetilde{x} - x$, we can conclude that the actual value x belongs to the interval $[\widetilde{x} - \Delta, \widetilde{x} + \Delta]$. In interval analysis, this is all we know: we performed measurements (or estimates), we get intervals, and we want to extract as much information as possible from these results. In particular, we want to know what can we conclude about $y = f(x_1, \ldots, x_n)$, where f is a known algorithm.

In computable (constructive) analysis, we take into account that eventually, we will be able to measure each x_i with higher and higher accuracy. In other words, for each quantity, instead of a *single* interval, we have a *sequence* of narrower and narrower intervals, a sequence that eventually converging to the actual value. From this viewpoint, *Interval analysis is applied constructive analysis* (Yuri Matiyasevich, of 10th Hilbert problem fame).

In this talk, we describe, from this viewpoint, what is a computable set, what is a computable function, and give examples of interval-related positive and negative results of computable analysis.

3.17 Need to Combine Interval and Probabilistic Uncertainty: What Needs to Be Computed, What Can Be Computed, What Can Be Feasibly Computed, and How Physics Can Help

Vladik Kreinovich (University of Texas – El Paso, US)

Need to Combine Interval and Probabilistic Uncertainty: What Needs to Be Computed, What Can Be Computed, What Can Be Feasibly Computed, and How Physics Can Help

In many practical situations, the quantity of interest y is difficult to measure directly. In such situations, to estimate y, we measure easier-to-measure quantities $x_1, \ldots, x_n$ which are related to y by a known relation $y = f(x_1, \ldots, x_n)$, and we use the results $X_1, \ldots, X_n$ of these measurement to estimatey as $Y = f(X_1, \ldots, X_n)$. How accurate is this estimate?

Traditional engineering approach assumes that we know the probability distributions of measurement errors $X_i - x_i$, however, in practice, we often only have partial information about these distributions. In some cases, we only know the upper bounds D_i; in such cases, the only thing we know about the actual value x_i is that it is somewhere in the interval $[X_i - D_i, X_i + D_i]$. Interval computation estimates the range of possible values of y under such interval uncertainty.

In other situations, in addition to the intervals, we also have partial information about the probabilities. In this talk, we describe how to solve this problem in the linearized case, what is computable and what is feasibly computable in the general case, and, somewhat surprisingly, how physics ideas – that initial conditions are not abnormal, that every theory is only approximate – can help with the corresponding computations.

References

1 Finkelstein, A. M., Kreinovich, V.: Impossibility of hardly possible events: physical consequences. Abstracts of the 8th International Congress on Logic, Methodology, and Philosophy of Science, Moscow, 1987, 5(2), 23–25 (1987)

2 Kosheleva, O., Zakharevich, M., Kreinovich, V.: If many physicists are right and no physical theory is perfect, then by using physical observations, we can feasibly solve almost all instances of each NP-complete problem, Mathematical Structures and Modeling 31, 4–17 (2014)

3 Kreinovich, V.: Toward formalizing non-monotonic reasoning in physics: the use of Kolmogorov complexity. Revista Iberoamericana de Inteligencia Artificial 41, 4–20 (2009)

4 Kreinovich, V.: Negative results of computable analysis disappear if we restrict ourselves to random (or, more generally, typical) inputs, Mathematical Structures and Modeling 25, 100–103 (2012)

5 Kreinovich, V., Finkelstein, A. M.: Towards applying computational complexity to foundations of physics. Notes of Mathematical Seminars of St. Petersburg Department of Steklov Institute of Mathematics 316, 63–110 (2004); reprinted in Journal of Mathematical Sciences 134(5), 2358–2382 (2006)

6 Kreinovich, V., Kunin, I. A.: Kolmogorov complexity and chaotic phenomena. International Journal of Engineering Science 41(3), 483–493 (2003)

7 Kreinovich, V., Kunin, I. A.: Kolmogorov complexity: how a paradigm motivated by foundations of physics can be applied in robust control. In: Fradkov, A.L., Churilov, A.N., eds. Proceedings of the International Conference "Physics and Control" PhysCon'2003, Saint-Petersburg, Russia, August 20–22, 2003, 88–93 (2003)

8 Kreinovich, V., Kunin, I. A.: Application of Kolmogorov complexity to advanced problems in mechanics. Proceedings of the Advanced Problems in Mechanics Conference APM'04, St. Petersburg, Russia, June 24–July 1, 2004, 241–245 (2004)

9 Kreinovich, V., Longpré, L., Koshelev, M.: Kolmogorov complexity, statistical regularization of inverse problems, and Birkhoff's formalization of beauty. In: Mohamad-Djafari, A., ed., Bayesian Inference for Inverse Problems, Proceedings of the SPIE/International Society for Optical Engineering, San Diego, California, 1998, 3459, 159–170 (1998)

10 Kreinovich, V., Pownuk, A., Kosheleva, O.: Combining interval and probabilistic uncertainty: what is computable?, In: Pardalos, P., Zhigljavsky, A., Zilinskas, J. (eds.): Advances in Stochastic and Deterministic Global Optimization, Springer Verlag, Cham, Switzerland, 2016, p. 13–32. http://www.cs.utep.edu/vladik/2015/tr15-66a.pdf

11 Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications, Springer (2008)

12 Pour-El, M. B., Richards, J. I.: Computability in Analysis and Physics, Springer, Berlin (1989)

13 Sun, L., Dbouk, H., Neumann, I., Schoen, S., Kreinovich, V.: Taking into account interval (and fuzzy) uncertainty can lead to more adequate statistical estimates, Proceedings of the 2017 Annual Conference of the North American Fuzzy Information Processing Society NAFIPS'2017, Cancun, Mexico, October 16–18, 2017. http://www.cs.utep.edu/vladik/2017/tr17-57a.pdf

14 Weihrauch, K.: Computable Analysis, Springer-Verlag, Berlin (2000)

3.18 An Approach to Programming Configurable Computers for Numeric Applications

Fritz Mayer-Lindenberg (TU Hamburg-Harburg, DE)

While conventional programming languages deal with a few types of numbers only, typically the ones implemented on their standard target processors, configurable computers, in particular FPGA based ones, can implement a large variety of number codes to make resource-aware application specific selections of the types to be used, and add an additional layer of programming this way. This motivated the design of a new programming language, besides the general need for simple programming tools for FPGA based systems. First, the required circuit design for the processors to be configured and for an infrastructure linking them to each other and to the memory resources is taken out of the task of programming numeric algorithms by building on a library of precompiled configurations. For the algorithmic programming, dealing with the various number codes as separate predefined types would result in a reduced level of abstraction. Instead, a single, abstract type of number is used as proposed in [1], at the same time rising abstraction to the level of mathematical algorithms on real numbers and clearly distinguishing numbers from their codes. The various number codes are represented by individual operations only, namely the corresponding roundings. By predefining tuples of numbers and operations on them, non-standard tuple codes such as block floating point codes can be supported as well to further reduce circuit complexity and memory requirements [2]. The language then proceeds to structures for the required control of parallel processing on heterogeneous sets of rather simple compute nodes for the different number codes [3]. In particular, it includes statements to identify the processor networks configured in an FPGA with sufficient detail to support native code generation for each individual processor, to select library components, and to simulate the operation of an entire parallel target.

References

1 F. Mayer-Lindenberg, A Management Scheme for the Basic Types in Programming Languages, SOFSEM 2005

2 F. Mayer-Lindenberg, A Modular Processor Architecture for HPC on FPGA, CDES'12

3 F. Mayer-Lindenberg, A programming language for embedded processor networks, ICSICT 2016

3.19 On implementing TTE

Norbert T. Müller (Universität Trier, DE)

Type-2-Theory of Effectivity (TTE) is the accepted model for computability and computational complexity on real numbers. One of its many advantages is that there do exist implementations being consistent with this theory.

The talk presents main common aspects of these implementations with several examples (like dynamical systems and ODE solving) and an outlook on future directions.

3.20 On the numerical norm estimation of the inverse operator in Hilbert space

Mitsuhiro T. Nakao (Waseda University – Tokyo, JP)

Joint work of Mitsuhiro T. Nakao, Yoshitaka Watanabe, Takehiko Kinoshita

We present some relations between the exact norm for the inverse operator in Hilbert space and its numerical estimation by some approximate methods. In order to verify numerically an exact solution of the nonlinear operator equations, particularly for partial differential equations, around some neighborhood of the approximate solution, it is important to calculate some approximate norm for an inverse of linearized operators. In this talk we clarify the conditions so that such approximate estimates by the finite element methods converge to the exact norm. This result enables us the appropriateness of the estimation by concerned numerical methods.

References

1 Y. Watanabe, T. Kinoshita, M. T. Nakao, A posteriori estimates of inverse operators for boundary value problems in linear elliptic partial differential equations, Mathematics of Computation, 82 (2013), 1543-1557. DOI:10.1090/S0025-5718-2013-02676-2

2 T. Kinoshita, Y. Watanabe, M. T. Nakao, Some remarks on the rigorous estimation of inverse linear elliptic operators, SCAN 2014, LNCS 9553, the proceedings of the conference '16th GAMM-IMACS International Symposium on Scientifc Computing, Computer Arithmetic and Validated Numerics' (2016), 225-235. DOI: 10.1007/978-3-319-31769-4-18

3.21 Parametrised complexity for the naive Cauchy representation

Eike Neumann (Aston University – Birmingham, GB) and Florian Steinberg (TU Darmstadt, DE)

Joint work of Eike Neumann, Florian Steinberg
Main reference Eike Neumann, Florian Steinberg: "Parametrised second-order complexity theory with applications to the study of interval computation", CoRR, Vol. abs/1711.10530, 2017.
URL http://arxiv.org/abs/1711.10530

The aim of this work is to provide a rigorous complexity framework for "non-rigorous" numerical algorithms, which come without explicit error bounds or convergence rates.

The standard representation of real numbers used in computable analysis, where a real number is encoded by a fast converging Cauchy sequence of dyadic rational numbers, leads to a very robust and realistic notion of computability and complexity on the reals which is closely related to rigorous numerical analysis.

In order to model non-rigorous computation one could attempt to simply drop the requirement of fast convergence. The resulting representation - the so-called naive Cauchy representation - is however well-known to be very ill-behaved computationally.

We show that the space of naive Cauchy reals can be enriched with a natural parameter, which essentially encodes a rate of convergence, to obtain a reasonable computability and complexity structure.

Although the resulting parametrised space of naive Cauchy reals is not even topologically equivalent to the space of real numbers with the standard representation, it has the same

(polytime) computable points. Moreover, a real function is (polytime) computable with respect to the parametrised naive Cauchy representation if and only if it is (polytime) computable in the usual sense.

We further show that the space of continuous real functions on the compact unit interval admits a minimal parametrised representation such that the evaluation functional

$$\mathrm{eval}\colon C([0,1]) \times [0,1] \to \mathbb{R}$$

becomes polytime computable when $[0,1]$ and $\mathbb{R}$ are given the parametrised naive Cauchy representation.

3.22 Test Matrices for Numerical Linear Algebra

Katsuhisa Ozaki (Shibaura Institute of Technology – Saitama, JP)

Main reference Katsuhisa Ozaki, Takeshi Ogita: "Generation of Linear Systems with Specified Solutions for Numerical Experiments", Reliable Computing, Vol. 25, pp. 148–167, 2017.

This talk concerns test matrices for numerical linear algebra, especially, linear systems and eigenvalue problems. If exact solution is known in advance, it is very useful for checking the accuracy and stability of numerical algorithms. Residual is often used for the check of the accuracy of numerical results. However, numerical solutions with small residual and big error can be obtained. We propose methods that produce problems with the exact solution based on error-free transformation of floating-point arithmetic. For linear systems, our methods generate a matrix, a solution and a right-hand side vector whose all elements are representable in floating-point numbers. We showed the exact stepwise errors for the BiCGSTAB method for several sparse matrices. For eigenvalue problems, our methods produce a matrix and its specified eigenvalues using the Hadamard matrix.

3.23 A tutorial on reliable numerical computation

Paul Zimmermann

This tutorial presents different software tools to perform reliable numerical computations: machine double precision using the IEEE 754 standard, fixed-precision interval arithmetic, arbitrary precision floating-point or interval arithmetic using MPFR, MPFI, or the Arb library. All these kinds of arithmetic can be used within the SageMath computer algebra system.

3.24 Computational complexity of solving polynomial differential equations over unbounded domains

Amaury Pouly (MPI-SWS – Saarbrücken, DE) and Daniel Graça (University of Algarve, PT)

Main reference Amaury Pouly, Daniel S. Graça: "Computational complexity of solving polynomial differential equations over unbounded domains", Theor. Comput. Sci., Vol. 626, pp. 67–82, 2016.
URL http://dx.doi.org/10.1016/j.tcs.2016.02.002

In this abstract we present a rigorous numerical algorithm which solves initial-value problems defined with polynomial differential equations (i.e. initial-value problems of the type $y' = p(t, y)$, $y(t_0) = y_0$, where p is a vector of polynomials) for any value of t. The inputs of the algorithm are the data defining the initial-value problem, the time T at which we want to compute the solution of the IVP, and the maximum allowable error epsilon>0. Using these inputs, the algorithm will output a value $\tilde{y}_T$ such that $|\tilde{y}_T - y(T)|$.

3.25 Numerical (arte-)facts and reliable computing

Siegfried M. Rump (TU Hamburg-Harburg, DE)

URL http://www.ti3.tuhh.de

We show examples of linear systems where the computed approximation has no correct digit, but nevertheless the residual (computed in floating-point) is exactly equal to zero. Similarly, the true inverse A^{-1} rounded to the nearest floating-point matrix $R := fl(A^{-1})$ may produce a residual $I - RA$ of norm larger than 1, but an approximate inverse $\tilde{R} = inv(A)$ computed by Matlab satisfies $\|I - \tilde{R}A\| < 1$. Despite, the entries of $\tilde{R}$ are wrong by more than a factor 2.

A remedy to incorrect approximations are verification methods. If the precision is not sufficient to compute correct error bounds for a solution, a corresponding message is given. Wrong results are not possible.

The power of verification methods is demonstrated by problems from quantum chemistry. Here floating-point algorithms fail completely, where verification methods provide tight inclusions of the solution. This is for problem sizes up to 30 million unknowns with tens of thousands of constraints.

3.26 Co-Polish spaces in Complexity Theory

Matthias Schröder (Universität der Bundeswehr – München, DE)

Co-Polish spaces play an important role in Type Two Complexity Theory. A Co-Polish space is defined to be a sequential topological space that is regular and for which the compact-open topology on the function space $C(X, IR)$ has a countable base. Co-Polish spaces turn out to be exactly those Hausdorff qcb-spaces X that admit a Simple Complexity Theory. Simple

Complexity Theory means that time complexity for functions on X can be measured in the desired output precision plus a *discrete* parameter on the input. For general spaces X, for example for non-locally-compact metric spaces X, an indiscrete parameter on the input is necessary.

3.27 Bit complexity of Computing Solutions for Symmetric Hyperbolic Systems of PDEs with Guaranteed Precision

Svetlana Selivanova

Joint work of Svetlana Selivanova, Victor Selivanov

The algorithms used in mathematics-oriented software can be divided into two big classes: symbolic algorithms which aim to find precise solutions, and approximate algorithms which aim to find "good enough" approximations to precise solutions. The symbolic algorithms are implemented e.g. in computer algebra systems while the approximate algorithms — in numerical mathematics packages. The both classes of algorithms are widely used in applications and in mathematical research. The symbolic algorithms correspond well to computations on discrete structures (with mathematical foundations in the classical computability and complexity theory) while the approximate algorithms — to computations on continuous structures (with mathematical foundations in the field of computability and complexity in analysis evolving under the slogan "Exact real computation").

An important idea relating the both classes of algorithms is to look for approximate solutions to a numerical problem with "guaranteed precision". The bit complexity of an algorithm is fundamental because it estimates the amount of computational resources needed to implement the algorithm on a computing device. Here we investigate the bit complexity of finding guaranteed precision solutions for Cauchy and boundary-value problems for symmetric hyperbolic systems of PDEs (see e.g. [1]) Such systems can be used to describe a wide variety of physical processes like those considered in the theories of elasticity, acoustics, electromagnetism etc. Accordingly, many people from theoretical and numerical mathematics worked on the existence and uniqueness theorems as well as on numerical methods of computing solution operators for problems related to such systems (the explicit solution formulas exist only in some simplest particular cases).

In [2] we developed an approach to the study of computability of the Cauchy and dissipative boundary-value problems for such systems based on finite-dimensional approximations (the so called difference schemes widely used in numerical analysis) and established the computability of solution operators in the rigorous sense of the TTE approach to computable analysis [3]. The main obstacle in proving the computable dependence of solutions on the input matrices is the fact that all known stable difference schemes for finding the approximate solutions use eigenvectors of some matrices and matrix pencils but these eigenvectors are known to be non-computable [4]. To overcome the obstacle, we considered in [2] restrictions of the solution operators to computably presentable real closed number fields and have shown that such restricted solution operators are computable. This fact together with close relationships of such fields to the field of computable reals (also established in [2]) imply that the solution operators are computable for any fixed computable input matrices.

We develop the approach from [2] to establish some reasonable upper bounds for some guaranteed-precision problems related to symmetric hyperbolic systems. A version of such a problem asks, given a fixed number of space variables, given algebraic real input matrices, rational polynomials as initial-value functions, and a precision p, to find an algebraic grid function such that the difference between the poly-linear interpolation of grid function and the precise solution is at most p. We establish the EXPTIME upper complexity bound for such problems and show that the estimate becomes polynomial under some additional restrictions often used in the practice of numerical methods. To our knowledge, these are the first such bounds in the literature.

Our approach makes a heavy use of some known and our own algorithms of computer algebra (exact computations with integers, rationals, algebraic reals and polynomials, polynomial-time computability of spectral decomposition of symmetric matrices and matrix pencils in the field of algebraic reals), together with some algorithms from numerical mathematics and computable analysis used in [2]. Altogether, our proofs demonstrate a fruitful mix of methods from symbolic and numerical computation. Although our methods do not seem to yield practically feasible algorithms for guaranteed precision, we hope that investigations in this direction are fruitful for both theoretical research and applications. In particular, on the implementation level it seems useful and rewarding to enhance the existing systems of "exact real computations" (like iRRAM) by packages based of highly developed algorithms of computer algebra. We are not aware of the existence of such "hybrid" systems built under the slogan of "guaranteed precision numerical computations".

References

1 Godunov S. K., ed.: Numerical Solution of Higher-dimensional Problems of Gas Dynamics (in Russian). Nauka, Moscow, 1976.

2 Selivanova S., Selivanov V.: Computing Solution Operators of Boundary-value Problems for Some Linear Hyperbolic Systems of PDEs. Logical Methods in Computer Science, 13 (4:13) 2017, pp. 1–31. Earlier version on arXiv:1305.2494 (2013).

3 Weihrauch K.: Computable Analysis. Berlin, Springer, 2000.

4 Ziegler M., Brattka V.: A computable spectral theorem. Proc. CCA-2001, Lecture Notes in Computer Science, v. 2064 (2001), 378–388.

3.28 Parametrised second-order complexity theory with applications to the study of interval computation

Florian Steinberg (TU Darmstadt, DE) and Eike Neumann (Aston University – Birmingham, GB)

Main reference Eike Neumann, Florian Steinberg: "Parametrised second-order complexity theory with applications to the study of interval computation", CoRR, Vol. abs/1711.10530, 2017.
URL http://arxiv.org/abs/1711.10530

We extend the framework for complexity of operators in analysis devised by Kawamura and Cook (2012) to allow for the treatment of a wider class of representations. The main novelty is to endow represented spaces of interest with an additional function on names, called a parameter, which measures the complexity of a given name. This parameter generalises the size function which is usually used in second-order complexity theory and therefore also central to the framework of Kawamura and Cook. The complexity of an algorithm is measured in terms of its running time as a second-order function in the parameter, as well as in terms of how much it increases the complexity of a given name, as measured by the parameters on the input and output side. As an application we develop a rigorous computational complexity theory for interval computation. In the framework of Kawamura and Cook the representation of real numbers based on nested interval enclosures does not yield a reasonable complexity theory. In our new framework this representation is polytime equivalent to the usual Cauchy representation based on dyadic rational approximation. By contrast, the representation of continuous real functions based on interval enclosures is strictly smaller in the polytime reducibility lattice than the usual representation, which encodes a modulus of continuity. Furthermore, the function space representation based on interval enclosures is optimal in the sense that it contains the minimal amount of information amongst those representations which render evaluation polytime computable.

This talk was based on the arXiv paper https://arxiv.org/abs/1711.10530 and a followup talk was given by Eike Neumann.

3.29 Verified computations for solutions to 1-dimensional advection equations with variable coefficients

Akitoshi Takayasu (University of Tsukuba, JP)

In this talk, we provide a methodology of verified numerical computations for solutions to 1-dimensional advection equations with variable coefficients. The advection equation is typical partial differential equations (PDE) of hyperbolic type. There are few results for verified computations to initial-boundary value problem of hyperbolic PDEs. Our methodology is based on the spectral method and semigroup theory. Numerical examples show that the rigorous error estimate showing the well-posedness of the exact solution is given with high accuracy and high speed.

3.30 Average case complexity for Hamiltonian dynamical systems

Holger Thies (University of Tokyo, JP), Akitoshi Kawamura (Kyushu University, JP), and Martin Ziegler (KAIST – Daejeon, KR)

Average case complexity in analysis was recently introduced by Schröder, Steinberg and Ziegler. We study the complexity of Hamiltonian systems like the famous n-body problem in this context. We use a simple parameterized worst-case complexity result for initial value problems with analytic right-hand side and the fact that Hamiltonian systems are volume preserving to relate the complexity of a system to the volume of singularities in phase space. As an application we show that the planar circular restricted three-body problem is computable in polynomial-time on average.

3.31 Small divisors and normal forms

Warwick Tucker (Uppsala University, SE)

Joint work of Warwick Tucker, Zbigniew Galias

In this talk, we will discuss the computational challenges of computing trajectories of a non-linear ODE in a region close to a fixed-point. By introducing a carefully selected close to identity change of variables, we can bring the non-linear ODE into an "almost" linear system. Determining the domain of existence for such a change of variables poses some interesting computational challenges.

3.32 Soft Foundations for Geometric Computation

Chee K. Yap (New York University, US)

Joint work of 1. Roots (V. Sharma, A. Eigenwillig, M. Sagraloff, R. Becker, J. Xu) 2. Surfaces (M. Burr, S. Choi, L. Lin, V. Sharma, G. Vegter) 3. Motion Planning (Y.-J. Chiang, C. Wang, J.-M. Lien, Z. Luo, C.-H. Hsu) 4. Voronoi Diagrams (E. Papadopoulou, H. Bennett, V. Sharma, J.-M
URL http://cs.nyu.edu/ exact/

For over two decades, Exact Geometric Computation (EGC) has provided a paradigm in Computational Geometry for the correct implementation of geometric algorithms. It is the most successful approach to numerical nonrobustness issues, leading to software libraries and practical algorithms. We review some reasons to extend this paradigm:

- EGC algorithms may not be Turing computable (e.g., transcendental functions)
- EGC may be too inefficient (e.g., shortest path problems)
- EGC entails numerous/difficult algebraic analysis (e.g., Vor diagram of polyhedra)
- Exact computation is inappropriate for the physical world (e.g., robot motion planning)

This talk describes a program to develop "soft" approaches for addressing these issues. "Soft" refers to numerical, certified approaches that nevertheless provide some modified notions of "hard" (topological/combinatorial) guarantees in the output. We illustrate these ideas by work in four areas:

- root isolation and clustering (ISSAC'09,'11,'12,'16, SNC'11, CiE'13, JSC'17)
- isotopic approximation of curves and surfaces (ISSAC'08, SoCG'09, SPM'12, ICMS'14)
- Voronoi diagrams (ISVD'13, SGP'16)
- robot motion planning (SoCG'13, WAFR'14, FAW'15, WAFR'16)

Common themes in this list include: we replace the Real RAM model by one based on numerical iteration on interval approximations. Algorithms are framed in the algorithmic paradigm of subdivision. We introduce an input resolution parameter (epsilon) but use it in novel "soft" ways. We design soft versions of classical hard geometric predicates in order to construct effective and practical algorithms. Some consequences of such a computational paradigm are:

- scope of computational geometry is vastly broadened to non-linear non-algebraic problems.
- unsolvable/hard problems in the Real RAM model becomes feasible
- soft algorithms are implementable and practical

One challenge is to revisit other classical problems of computational geometry with this view point. Another is to produce complexity analysis of such algorithms. Successes in complexity analysis for roots suggest that similar sharp "amortized" results can be obtained in higher dimensions.

References

1 Chee K. Yap, Vikram Sharma: Robust Geometric Computation. Encyclopedia of Algorithms 2016: 1860–1863

2 Ruben Becker, Michael Sagraloff, Vikram Sharma, Juan Xu, Chee Yap: Complexity Analysis of Root Clustering for a Complex Polynomial. ISSAC 2016: 71–78

3 Long Lin, Chee Yap, Jihun Yu: Non-local isotopic approximation of nonsingular surfaces. Computer-Aided Design 45(2): 451–462 (2013)

4 Huck Bennett, Evanthia Papadopoulou, Chee Yap: Planar Minimization Diagrams via Subdivision with Applications to Anisotropic Voronoi Diagrams. Comput. Graph. Forum 35(5): 229–247 (2016)

5 Cong Wang, Yi-Jen Chiang, Chee-Keng Yap: On soft predicates in subdivision motion planning. Comput. Geom. 48(8): 589–605 (2015)

3.33 On Formal Verification in Imperative Multivalued Programming over Continuous Data Types

Martin Ziegler (KAIST – Daejeon, KR), Gyesik Lee, Norbert T. Müller (Universität Trier, DE), Eike Neumann (Aston University – Birmingham, GB), Sewon Park (KAIST – Daejeon, KR), and Norbert Preining

Main reference Norbert Th. Müller, Sewon Park, Norbert Preining, Martin Ziegler: "On Formal Verification in Imperative Multivalued Programming over Continuous Data Types", CoRR, Vol. abs/1608.05787, 2016.
URL http://arxiv.org/abs/1608.05787v2

Inspired and guided by the iRRAM C++ library (Müller 2001), we formally specify a programming language for the paradigm of Exact Real Computation (ERC): reliably operating on encapsulated continuous data types such as (not necessarily algebraic) real numbers - imperatively and exactly (no rounding errors) with primitives computable in the sense of Recursive Analysis including a necessarily modified multivalued (=non-functional) semantics of tests. Three simple numerical problems demonstrate the elegance and convenience of writing programs handling real (and not just, say rational or algebraic) numbers: integer rounding, solving systems of linear equations, and continuous root finding. We establish Turing-completeness over the reals: a partial function is computable (in the sense of Recursive Analysis) iff it can be expressed in ERC. For rigorously specifying and arguing about such computations in Mathematical Logic, we then propose a decidable first-order theory over two sorts, integers and real numbers. We extend the rules of Hoare Logic to support the formal derivation of correctness proofs in ERC; and we have them, including their real quantification, verified in the Coq Proof Assistant.

3.34 Tutorial on Computational Complexity in Analysis

Martin Ziegler (KAIST – Daejeon, KR) and Akitoshi Kawamura (Kyushu University, JP)

URL http://theoryofcomputation.asia/survey3.pdf

Computability Theory in Analysis explores the ultimate capabilities and limitations of computing with real numbers, functions, and subsets. Complexity Theory in Analysis refines these investigations with respect to efficiency. It complements the classical Complexity Theory over discrete structures and adapts its generic goal to continuous data, based on two pillars: (i) Design and rigorous analysis of algorithms for approximating the solution up to guaranteed absolute error $1/2^{-n}$. The computational resources (runtime, memory) thus incurred constitute upper bounds on the computational complexity inherent to the problem; and (ii) establishing – preferably tight – lower bounds, that is, prove any algorithmic solution to require that many resources: This exhibits the algorithm from (i) as optimal, possibly subject to standard hypotheses such as P<>NP<>PSPACE<>EXP or by adapting adversary arguments from Information-Based Complexity to the bit-cost model.

Participants

- Andrej Bauer
University of Ljubljana, SI
- Henning Behnke
TU Clausthal, DE
- Jens Blanck
Swansea University, GB
- Franz Brauße
Universität Trier, DE
- Florian Bünger
TU Hamburg-Harburg, DE
- Pieter Collins
Maastricht University, NL
- George F. Corliss
Marquette University –
Milwaukee, US
- Tibor Csendes
University of Szeged, HU
- Eva Darulova
MPI-SWS – Saarbrücken, DE
- Hubert Glesener
Universität Trier, DE
- Daniel Graça
University of Algarve, PT
- Stef Graillat
UPMC – Paris, FR
- Peter Hertling
Universität der Bundeswehr –
München, DE
- Fabian Immler
TU München, DE
- Luc Jaulin
ENSTA Bretagne – Brest, FR
- Akitoshi Kawamura
Kyushu University, JP
- Sunyoung Kim
Ewha Womans University, KR
- Michal Konecny
Aston University –
Birmingham, GB
- Margarita Korovina
Russian Academy of Sc. –
Novosibirsk, RU
- Vladik Kreinovich
University of Texas – El Paso, US
- Ivo List
University of Ljubljana, SI
- Wolfram Luther
Universität Duisburg-Essen, DE
- Fritz Mayer-Lindenberg
TU Hamburg-Harburg, DE
- Atsushi Minamihata
AIST – Tokyo, JP
- Norbert T. Müller
Universität Trier, DE
- Mitsuhiro T. Nakao
Waseda University – Tokyo, JP
- Eike Neumann
Aston University –
Birmingham, GB
- Katsuhisa Ozaki
Shibaura Institute of Technology
– Saitama, JP
- Sewon Park
KAIST – Daejeon, KR
- Michael Plum
KIT – Karlsruher Institut für
Technologie, DE
- Amaury Pouly
MPI-SWS – Saarbrücken, DE
- Robert Rettinger
Fachhochschule Dortmund, DE
- Fabrice Rouiller
INRIA – Paris, FR
- Siegfried M. Rump
TU Hamburg-Harburg, DE
- Michael Sagraloff
MPI für Informatik –
Saarbrücken, DE
- Matthias Schröder
Universität der Bundeswehr –
München, DE
- Svetlana Selivanova
Sobolev Institute of Mathematics
– Novosibirsk, RU
- Florian Steinberg
TU Darmstadt, DE
- Akitoshi Takayasu
University of Tsukuba, JP
- Holger Thies
University of Tokyo, JP
- Warwick Tucker
Uppsala University, SE
- Joris van der Hoeven
Ecole Polytechnique –
Palaiseau, FR
- Klaus Weihrauch
FernUniversität in Hagen, DE
- Nobito Yamamoto
The University of
Electro-Communications –
Tokyo, JP
- Yuuka Yanagisawa
Waseda University – Tokyo, JP
- Chee K. Yap
New York University, US
- Martin Ziegler
KAIST – Daejeon, KR
- Paul Zimmermann
INRIA Nancy – Grand Est, FR