DAGSTUHL
REPORTS

Volume 5, Issue 1, January 2015

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
May, 2015

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.5.1.i

Aims and Scope
The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.
In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:

- an executive summary of the seminar program and the fundamental results,
- an overview of the talks given during the seminar (summarized as talk abstracts), and
- summaries from working groups (if applicable).

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

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

Report from Dagstuhl Seminar 15021

Concurrent Computing in the Many-core Era

Edited by

Michael Philippsen[1], Pascal Felber[2], Michael L. Scott[3], and J. Eliot B. Moss[4]

1 **Universität Erlangen-Nürnberg, DE,** michael.philippsen@fau.de
2 **Université de Neuchâtel, CH,** pascal.felber@unine.ch
3 **University of Rochester, US,** scott@cs.rochester.edu
4 **University of Massachusetts – Amherst, US,** moss@cs.umass.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 15021 "Concurrent computing in the many-core era". This seminar is a successor to Dagstuhl Seminars 08241 "Transactional memory: From implementation to application" and 12161 "Abstractions for scalable multicore computing", respectively held in June 2008 and in April 2012. The current seminar built on the previous seminars by notably (1) broadening the scope to concurrency beyond transactional memory and shared-memory multicores abstractions, (2) focusing on the new challenges and potential uses of emerging hardware support for synchronization extensions, and (3) considering the increasing complexity resulting from the explosion in heterogeneity.

Seminar January 5–9, 2015 – http://www.dagstuhl.de/15021
1998 ACM Subject Classification C.1.3 [Processor Architectures]: Other Architecture Styles – Heterogeneous (hybrid) systems, D.1.3 [Programming Techniques]: Concurrent Programming – Parallel Programming, D.3.3 [Programming Languages]: Language Constructs and Features – Concurrent programming structures, D.3.4 [Programming Languages]: Processors – Compilers, Memory Management, D.4.1 [Operating Systems]: Process Management – Synchronization, D.4.2 [Operating Systems]: Storage Management; H.2.4 [Database Management]: Systems – Transaction Processing
Keywords and phrases Multi-/many-core processors, Concurrent Programming, Synchronization, Transactional Memory, Programming Languages, Compilation
Digital Object Identifier 10.4230/DagRep.5.1.1

1 Executive Summary

Pascal Felber
Michael Philippsen
Michael L. Scott
J. Eliot B. Moss

Context and Motivations

Thirty years of improvement in the computational power of CMOS uniprocessors came to an end around 2004, with the near-simultaneous approach of several limits in device technology (feature scaling, frequency, heat dissipation, pin count). The industry has responded with ubiquitous multi-core processors, but scalable concurrency remains elusive

for many applications, and it now appears likely that the future will be not only massively parallel, but also massively heterogeneous.

Ten years into the multi-core era, much progress has been made. C and C++ are now explicitly parallel languages, with a rigorous memory model. Parallel programming libraries (OpenMP, TBB, Cilk++, CnC, GCD, TPL/PLINQ) have become mature enough for widespread commercial use. Graphics Processing Units support general-purpose data-parallel programming (in CUDA, OpenCL, and other languages) for a widening range of fields. Transactional memory appears likely to be incorporated into several programming languages. Software support is available in multiple compilers, and hardware support is being marketed by IBM and Intel, among others.

At the same time, core counts are currently lower than had once been predicted, in part because of a perceived lack of demand, and the prospects for increased core count over time appear to be constrained by the specter of dark silicon. Parallel programming remains difficult for most programmers, tool chains for concurrency remain immature and inconsistent, and pedagogical breakthroughs for the first- and second-year curriculum have yet to materialize. Perhaps most troublesome, it seems increasingly likely that future microprocessors will host scores or even hundreds of heterogeneous computational accelerators, both fixed and field-programmable. Programming for such complex chips is an exceptionally daunting prospect.

The goal of this Dagstuhl research seminar was to bring together leading international researchers from both academia and industry working on different aspects of concurrent computing (theory and practice, software and hardware, parallel programming languages, formal models, tools, etc.) in order to:

- assess the state of the art in concurrency, including formal models, languages, libraries, verification techniques, and tool chains;
- explore the many potential uses of emerging hardware support for transactional memory and synchronization extensions;
- envision next-generation hardware mechanisms;
- consider potential strategies to harness the anticipated explosion in heterogeneity; and
- investigate the interaction of synchronization and consistency with emerging support for low-latency byte-addressable persistent memory. (This last goal emerged late in the planning process, but became a major topic of discussion.)

Participants came from a wide variety of research communities, which seldom have the opportunity to meet together in one place. The seminar therefore provided a unique opportunity to focus diverse expertise on a common research agenda for concurrent computing on new generations of multi- and many-core systems.

Research Challenges

As part of this seminar, we specifically addressed the following challenges and open research questions, which are the focus of substantial investigation both in academia and in industry. These issues were addressed during the discussion at the workshop from the various perspectives of theory, concurrent algorithms, systems software, and microarchitecture.

The Future of Transactional Memory

With the introduction this past year of TM-capable commodity processors from IBM and Intel, TM research is increasingly turning to the question of how best to use the new hardware.

What can and cannot be accomplished with the simple interfaces currently available? What might be accomplished with the addition of non-transactional loads and/or stores within transactions? (And how should such stores behave?) What support might be needed for nested transactions or nested parallelism?

Given that machines without TM will exist for many years, and that HTM will remain bounded by constraints on capacity, associativity, etc., how should hardware and software transactions interact? What hardware extensions might facilitate the construction of hybrid systems? Can hardware transactions be used to accelerate STM? Is TM hardware useful for purposes other than TM?

Beyond these basic questions, how do we integrate TM into the concurrency tool chain? How does one debug a black-box atomic operation? How should TM be embedded into programming languages? Should speculation be visible to the programmer, or should it be hidden within the implementation? How large can transactions reasonably become? Should they remain primarily a means of building concurrent data structures, or should they expand to encompass larger operations-even system-level functions like I/O, thread/process interactions, and crash recovery? As implementations proliferate, are there reasonable models of correctness that move beyond opacity? How should we benchmark TM code? What performance counters should future TM hardware provide to profilers? What kind of infrastructure is needed to perform regression testing of transactional code?

Heterogeneity

GPUs are increasingly regarded as general-purpose computational resources, in platforms ranging from cell phones to supercomputers. Cell phones commonly include additional accelerators as well, for (de)compression, (de)encryption, and media transcoding. These and other accelerators (e.g., for linear algebra, pattern matching, XML parsing, or field-programmable functions) are likely to appear across the computing spectrum over the next few years.

In contrast to traditional (e.g., vector or floating-point) functional units, whose operations are uniformly short, and to traditional I/O devices, whose operations are uniformly long, accelerators can be expected to display a very wide range of response times. Long and variable response times suggest the need for resource management, to promote fair use across threads and applications. Short response times suggest the need for direct, user-level access-as already provided by GPU drivers from nVidia and (soon) AMD.

The prospect of contention for shared accelerators, accessed directly from user-level code, raises a host of questions for concurrent programming. How do we arbitrate shared access? Can traditional notions of locality be extended to accommodate heterogeneity? What happens to the tradeoff between local and remote computation when the alternatives use different instruction sets? What abstract models of progress/performance/time complexity are appropriate? Can operations that employ shared accelerators ever be considered non-blocking? How should we benchmark code that makes use of accelerators? What performance measures should heterogeneous architectures should provide to profilers? What kind of infrastructure is needed to perform regression testing in the face of heterogeneity?

Persistence

Exceptions like magnetic core and battery-backed RAM notwithstanding, mainstream computing has long maintained a firm separation between fast, volatile working memory and slow, non-volatile (persistent) storage. Emerging low-latency, byte-addressable technologies

like phase-change memory, memristors, and spin-torque-transfer memory bring this tradition into question. While near-term implementations may simply use low-latency nonvolatile memory as an accelerator for conventional file systems, alternative APIs may prove attractive. Specifically, it seems likely that future systems will give programmers the option of computing directly on persistent state, rather than reading it into working memory, using it there, and writing it out again. This possibility raises variants of many of the issues that have long concerned the concurrency community – consistency and atomicity in particular.

How should pointer-rich, non-file-based data be managed? Will we need automatic garbage collection? What will be the persistent analogues of nonblocking concurrent data structures? How will we ensure linearizability? Composability? A seemingly obvious option would add the 'D' (durability) to transactional memory's ACI (atomicity, consistency, and isolation). With little near-term prospect for integration of persistence and hardware TM, how will we minimize the overheads of persistent STM? What will the tradeoffs look like with respect to lock-based Lock-based programming models? What will be the division of labor between the operating system, runtime, and compiler? What will be the complexity models? Will we count "persistent accesses" the way we currently count remote memory accesses for concurrent objects in memory?

Pedagogy

Once upon a time, concurrency was a specialized topic in the undergraduate curriculum, generally deferred to the operating systems course, or to an upper-level elective of its own. Now it is an essential part of the training of every computer scientist. Yet there is surprisingly little consensus on where it belongs in the curriculum, and how it ought to be taught. Alternatives range from "concurrency first," to infusion throughout the curriculum, to more extensive coverage in a more limited number of courses.

While the principal focus of the seminar was on research issues, participants had the opportunity to share both intuition and experience in the teaching of concurrency, during a dedicated panel session and as part of informal discussions. The following questions were notably discussed. What works, for which kinds of students? What languages and tool chains should we use? What textbooks do we need? What role (if any) should be played by deterministic parallel languages and constructs? Are there approaches, particularly for introductory students, that can offer parallel speedup for important applications, without the full complexity of the general case? Can these approaches reasonably be "staged" into intro-level courses?

Organization of the Seminar

The seminar lasted 5 days, each composed of short scientific presentations, with ample time for discussions, and break-out sessions during which various open questions were discussed in sub-groups. The first day of the seminar started with a general introduction and forward-looking presentations on concurrency and the challenges raised by heterogeneity and virtualization.

Ten technical sessions, with short presentations from the participants, took place during the seminar on:

- locks and TM;
- C++ status and standards;
- memory models;
- memory management and persistence;

- performance tuning and verification;
- distributed concurrency and fault-tolerance;
- thoughts on concurrency and parallelism;
- HW and portability;
- compilers, runtimes, and libraries; and
- languages and systems.

They were complemented by break-out sessions on "dealing with heterogeneity", the "future of TM", and "persistence", as well as a plenary discussion on "virtualization". Finally, a panel discussion was organized on the topic of "teaching concurrency". The seminar concluded with an open discussion on the future of concurrency and the challenges that will need to be adddressed in coming years.

The topic of the sessions and their diversity illustrate the complexity of the challenges raised by concurrent computing on multi- and many-core systems. As one can expect from such prospective seminars, the discussions raised almost as many new questions as they provided answers on the addressed research challenges. Indeed, while there has been significant advances since the previous seminars (08241 and 12161), notably in terms of hardware support, few of the outstanding problems have been completely solved and new ones have emerged. For instance, hardware support for TM is now available in consumer CPUs but it cannot be used straightforwardly in real applications without relying on hybrid software/hardware strategies, notably to deal with the lack of progress guarantees and the possibility of spurious aborts.

As detailed in the rest of this report, the seminar has allowed the community to make significant progress on a number of important questions pertaining to concurrent computing, while at the same time defining a research agenda for the next few years. Participants provided very positive feedback following the seminar and expressed strong interest in follow-up events. Organizers strongly support the continuation of this series of seminars on concurrent computing, one of the most important and challenging fields in the era of multi- and many-core systems.

2 Table of Contents

3 Jump-Start Talks

3.1 Heterogeneous Concurrency

Michael L. Scott (University of Rochester, US)

It appears increasingly likely that future multi-/many-core processors will be highly heterogeneous, with cores that differ not only in average performance and energy consumption, but also in purpose, with instruction sets and micro-architecture specialized for such tasks as vector computation, compression, encryption, media transcoding, pattern matching, and XML parsing. We may even see ubiquitous FPGAs on-chip.

The time appears to be ripe for concurrency researchers to explore open questions in this area. How will we write programs for highly heterogeneous machines? Possible issues include:

- What will be the policies and mechanisms to allocate and manage resources (cycles, scratchpad memory, bandwidth)?
- Will we continue to insist on monolithic stacks, or will it make sense to allocate frames dynamically (sometimes, perhaps, in local scratchpad memory)?
- How will we dispatch work to other cores? Hardware queues? Flat combining?
- How will we wait for the completion of work on other cores? Spin? Yield? De-schedule? Perhaps we shouldn't wait at all, but rather ship continuations?
- When work can be done in more than one place, how will we choose among cores with non-trivial tradeoffs (in power, energy, time, or load)? How will we generate code for functions that may use different ISAs depending on where they run?
- What is the right division of labor between the programming language, the run-time system, the OS, and the hardware?
- What features would we like architects to build into future machines?

Notes *(collected by members of the audience)*

The purpose of this talk is to jump-start conversation.

- Background: Why don't we have 1000-core multi-cores? Because they would melt (at least if you used all the cores at the same time). As a result, we're looking at a future with billions of transistors on the chip, but many of them will have to be turned off ("dark silicon"). ([Charles E. Leiserson]: Or you could clock them down.) One way of dealing with dark silicon is to build special-purpose, customized circuits. Anything that is a non-trivial portion of execution time could have a specialized circuit. This is already happening in mobile, where we may also have cores with different computational/energy tradeoffs.
- Future programs may have to "hop" between cores. There's a progression of functionality: (1) FPU: pure, simple function (e.g., arctan); protection is not really an issue. (2) GPU: fire-and-forget rendering. (3) GPGPU: compute and return (with memory access); direct access from user space; one protection domain at a time. (4) First-class core: juggle multiple contexts safely (really not an accelerator anymore); preemption, multiprogramming.

- Challenges with these heterogeneous cores: (1) How do we arbitrate access to resources (cycles, scratchpad memory, bandwidth)? (2) How do we choose among cores – e.g., the faster core or the more efficient one? (3) How do we get access to systems services on "accelerators"? (4) How do we handle data movement? The "best" core may not be best if we have to move data between cores, or if the "best" core may be overloaded with other computation. (5) How do we manage heterogeneous ISAs?
- Challenges for concurrency: (1) How do I dispatch across cores? (HW queues? flat combining?) (1) GPGPU accesses are not mediated by the operating system! (2) How do we manage stacks? (contiguous stacks? linked frames instead?) (3) How should we envision accelerator-based computing? Call function and get result back? Or do we want to think of this as shipping continuations? (4) What language support do we need? It would be nice to avoid writing code in a different language for every accelerator. (5) How do we manage signaling across cores? Wake up threads, etc.?
- Unsupported hypotheses: The traditional kernel interface is not going to last. It cannot capture everything as a pthread anymore. We're already heading in this direction, with extensions for user-space threads, etc. We really need to rethink how an OS supports threads. Contiguous stacks may need to be replaced with chains of dynamically allocated frames. That will need compiler support. Accelerator cores are going to need first-class status, with direct access to OS services. A tree-structured dynamic call graph will be too restrictive. Rather than assume call and return, the accelerator may need to decide what to do with the result, and where to run the current context next.
- Discussion:
  [David F. Bacon]: We may be over-generalizing lessons from GPUs. Many accelerators may be "fixed-function." An FPU-like model may be easier for managing complexity. IBM has a coherent accelerator interface for access to FPGAs, etc. My view is that I don't necessarily want to use coherent memory on an FPGA. Please have cores use the same ISA.
  [Michael L. Scott]: There may be important differences between an "encryption accelerator" with a standard encryption algorithm and an engine that knows how to do XSLT that can apply arbitrary function at each node of a tree.
  [Stephan Diestelhorst]: Jumping off single ISA thought. ARM has Big/Little. Sometimes there is a functional block that you want to leave out of the "small" ISA – e.g., the vector unit. So you have mostly the same ISA, but a "wimpy" core may not implement big instructions.
  [David F. Bacon] & [Michael L. Scott]: You can use microprogramming to implement "beefy" instructions – e.g., serializing vector instructions. That's the standard way to provide ISA compatibility.
  [Stephan Diestelhorst]: Would you want the support to be OS-visible?
  [David F. Bacon]: I want as much compatibility as possible. Heterogeneity is giving you so much hassle already; anything you can do to minimize this is worth doing.
  [Charles E. Leiserson]: Much depends on what you're trying to do. There's already a lot of difference between multi-/many-core/desktop computing vs. embedded. Over time, there may be even more separation among these sorts of things. A cell phone does a lot more specialized computing than a laptop does. In the cloud, there is more of a push for things to be homogeneous – e.g., Amazon Web Services turns off clock-frequency changing.
  [Michael L. Scott]: On the other hand, AWS will provide you a CUDA engine if you ask for it. As a general principle, answers may be different in different contexts. I tend to see

more convergence, rather than divergence. Embedded and general-purpose computing may be getting more similar, rather than different.
[Charles E. Leiserson]: Maybe at some point we'll have cloud-specialized processors?
[Michael L. Scott]: I would guess that we'd get even more specialized options – a menu – of possible processor choices.
[Charles E. Leiserson]: But that's hard to manage. AWS offers different kinds of machines, but they're concerned with keeping the number of offerings small.
[J. Eliot B. Moss]: If you offer a uniform ISA, you've "solved" the compiler problem. This becomes more of a scheduling/processor binding problem. It makes it easier to migrate threads between processors – and thus no different at the level of writing a program.
[David F. Bacon]: Maybe I'm not saying that all processors have same ISA, but you do want to minimize ISA heterogeneity.
[Torvald Riegel]: ISA is one layer of complexity, but not the only one. Performance differences still call for different types of code. It doesn't matter if every core supports vector instructions: if an accelerator is slow at vector code, you may still use a completely different kind of code. ISA uniformity just pushes the problem up to a different layer of the stack.
[J. Eliot B. Moss]: Note that performance heterogeneity is the whole reason for heterogeneity in the first place, so ultimately that's something you can't hide.
[Sven-Bodo Scholz]: Having a homogeneous ISA does not really make compiler people's lives easier. If the compiler is the place where you choose whether you use a vector instruction or not, the compiler still needs to know how fast the instruction is. If the compiler knows about heterogeneity it may be easier to produce good code than it is with "simulated" homogeneity (of the ISA) that under the hood turns out not to be homogeneous.
[Michael L. Scott]: I'm curious about whether contiguous stacks are an albatross.
[Charles E. Leiserson]: Absolutely. There was an opportunity when we went to a 64-bit software stack where we could have made a change, but we didn't. Current calling conventions are even more optimized for linear stacks than previous generations.
[J. Eliot B. Moss]: There are existing, widely used systems that do linked stack-chunks. Not at an individual frame level, but we don't always have fully-linear stacks.
[Hans-J. Boehm]: GCC supports discontiguous stacks?
[Torvald Riegel]: Mostly, but it's not that fine grained. You just don't have to reserve everything up front. But I don't think that contiguous stacks are the problem here. We need to start at the language level. What if we have execution agents that are not full pthreads (with scheduler guarantees, etc.) Looking at this from the languages/libraries side of things may be more productive.
[Stephan Diestelhorst]: Go has discontiguous stacks, threadlets, etc. These concepts may exist at the language level, but we may still need to do something at lower levels to make them faster.
[Dave Dice]: We've tried split chunk stacks in the JVM, but it's hard to get it to work across multiple platforms. We tried to do it for 32-bit code because it's faster, but we ran out of stack space. The JIT may be able to optimize out checks to see if there's enough space in current stack chunks. The big problem is that JVM interacts with C code, so the thread model/execution model must somewhat mirror the pthreads world.
[David F. Bacon]: We saw the same thing in Jikes. It's easier in a single-language environment.
[Charles E. Leiserson]: Cilk did linked frames for everything 15 years ago. Overhead in

GCC was only 1–2%. But that stuff is better optimized today, so overheads might be much higher. At the same time, the flexibility it gives you is so great, it may be worth the tradeoff.
[J. Eliot B. Moss]: Maybe part of the performance issue is not the number of instructions but what happens in the cache.
[Charles E. Leiserson]: I don't think cache is a big issue. Stacks may have more of an effect on the TLB. Even if you're allocating stack frames off the heap, if you're using the memory in a stack-like manner, you still get pretty good cache locality. C and C++ work well with malloc, because when you free something, that's what you allocate next, while managed languages don't give you this benefit.
[David F. Bacon]: This is really an architectural artifact, because we can't say that stuff we're freeing doesn't need to be in the cache.

3.2 Concurrency in Virtual Machines

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

Consider the problem faced by a designer of a virtual machine (instruction set and related facilities) intended to support a wide range of programming languages (static and dynamic, "low" level like C and "high" level like Haskell) on a range of hardware platforms. It is challenging enough to provide integer and floating point operations and basic control flow (compare, branch, call, return, exceptions). The situation is made rather more difficult with respect to concurrency. Not only is there variety around single-word atomic accesses and ordering of memory accesses, but "larger" abstractions such as messaging, block/wakeup, threads, and especially transactions, make it difficult to devise a suitable common denominator. We hope that discussion at this workshop will help advance our thinking about what a good collection of building blocks might be.

Notes *(collected by members of the audience)*

- Goal: define a language-independent and HW-independent intermediate representation (IR) that deals well with concurrency.
  "I have a very practical need to worry about this topic, because Tony Hosking and I are working on a new grant on building a new VM that deals well with concurrency. Problem setting: Language-level virtual machine that abstracts away hardware detail. Below the programming language – target representation for compilers. Would like to support a wide variety of languages. Similar to LLVM, but more targeted to managed languages. Similar to CLR. Want something close to JVM, but one that is less language-centric. Hard to target new languages to JVM, because you have to "bend over backwards" to fit things into JVM model. Support GC, threads, etc. Starting point: what should the instruction set/IR for the virtual machine look like? Points of agreement: arithmetic/logical instructions, primitive data types, call/return. But there's a lot we don't agree on." Assuming we do not want to impose specific high-level semantics such as race-freedom or a particular transaction model, what primitives or building blocks should we provide to the language implementer to allow them to roll their own and achieve good performance on various hardware? "Things nice to have for concurrency: some

single-word atomic primitives (CAS), guaranteed progress (FetchAndAdd) What about multi-word operations/transactions? No agreed-upon semantics in the languages, no standard support in hardware. So what primitives should a VM support? Goal: what are the key building blocks to build STM or exploit HTM." [Charles E. Leiserson]: This is not really VM specific: really common to any IR design problem.

- Some quick thoughts: (1) Support grouping operations together (some notion of "transactions"), (2) Should deal with ordering, (3) Should deal with policy (contention management), (4) Should handle multiple scales: single thread, hyperthreads, same socket, same box, more distributed.
- **Q**[Jose Nelson Ameral]: What do you have in mind when you say should deal with ordering? Is it TLS-style support, with single sequential order? **A**: Some notion of ordering between transactions. May want to specify a specific order in which transactions may commit. But may want to support general messaging, too. [Hans-J. Boehm]: If transactions are exposed at language level, also need to worry about memory visibility ordering between transactional code and non-transactional code. **Q**[Michael L. Scott]: Is it always the case that if B is ordered after transaction A, is it always the case that B will see every non-transactional operation that happens-before transaction A? **Q**[Torvald Riegel]: Are you adopting a data-race free requirement? Has implications for optimization. **A**: May be a good requirement. [Hans-J. Boehm]: But then you can't handle Java in its current form.
- Would also like to support semantics not just memory. Semantic conflicts, semantic undo/redo (open nesting, boosting, etc.). Not in hardware, but how can we integrate it into a system that perhaps uses HTM or other hardware support? How do we generate concurrency? What are the primitives? Fork/join? Do-across/Do-all? Communication/ordering? Wait/signal? Futures? If we put these things into a VM, could help support multiple languages. Though current project is not intended to support multiple languages simultaneously, due to library requirements. Scope: Tightly connected (cache coherent) to Loosely connected (distributed). How much can we assume is being handled in hardware vs. how much has to be managed by software. Likely to see less coherence in heterogeneous world. In summary, as a (language VM implementer): What should I offer to language implementers? As an abstraction of current/future hardware? To support current and future languages? [Michael L. Scott]: And what features should we suggest to architects?
- **Q**[Pascal Felber]: Do we want to have primitives for message passing in VM? **Q**[Maurice Herlihy]: What about existing languages? Common intersection between Ruby/ Python/Scala is difficult. **A**: Think more union, rather than intersection. [Michael L. Scott]: There is a lot to learn from CLR. Asked a lot of questions about how to support multiple languages. [Hans-J. Boehm]: Just don't copy the memory model! [Michael L. Scott]: Right. Learn from their mistakes, too. **A**: Worried that a model like CLR is mired in a past era of languages and hardware, rather than being more forward looking. [Michael L. Scott]: Useful mistake to learn from: wound up abandoning attempt to build managed language runtime on top of CLR. [Antony Hoskin]: My impression of CLR is that languages supported by CLR have a lot of commonality. I would worry that CLR constrains classes of languages that can be implemented. **A**: As an example of problems: what if VM allows arbitrary "pinning" of memory. That doesn't play well with some sorts of garbage collection. Would like to avoid that sort of hassle. [David F. Bacon]: Go allows interior pointers, which places a lot of limitations on what a VM can do. **A**: We support some limited use of interior pointers. But can't store them in arbitrary places,

send them around. Current VM supports something like structs, but anything higher level is supported more at the compiler/runtime level, instead of being baked in to IR. [Daniele Donetta]: Working on a similar project called Truffle. Attempt from Oracle to answer same questions. Support Java, JavaScript, Ruby, Python, R. But not focusing on interoperability. We answered first question (what to offer language implementers) offered API to language implementer. API is basically to write AST interpreter. But dealing with same problems of concurrency. **A**: We support "tagged" data type, and can implement optimizations where once the tag is tested, and you know what the type is, can generate specific code. [Pascal Felber]: Is there any idea of how much of a performance hit you would take by supporting multiple languages? The more general you are, the more compromises you have to make on performance/code generation. **A**: We would like to give "reasonable" performance, but not necessarily the best performance. The project is not trying to do the best JIT ever for a given IR. But the goal is for the IR to not substantially inhibit achieving good performance. Part of the project is to look at loop kernels and make sure that performance is within a few percent of GCC performance. Just because we support dynamic types doesn't mean that you have to use those dynamic types. [Antony Hoskin]: Idea is to regenerate new JIT-ed code as language-level compiler learns more about types. **A**: Different starting point than LLVM. Not looking at backend for heavily-optimized language. Instead, we're looking at situations where you're throwing new IR at VM (more-refined version of functions). [Charles E. Leiserson]: Right now, do-all is implemented as syntactic sugar on top of library code, so don't get the same optimizations as real for loops. How does that problem get solved in this context? What you want to do is optimize it as a real for loop, and then afterwards say "oh, this is parallel, so use language-specific parallelism construct to implement it." **A**: One way to phrase this question: how much of the optimization needs to happen before IR (language specific) and how much are we leaning on the language-independent JIT. [Charles E. Leiserson]: What you would want is some sort of callback: JIT implements strength-reduction/ code motion, then calls back to compiler for actual parallel language construct. **A**: Could imagine some sort of step-wise refinement. I see that it's only an integer, so I'll generate code with a test at the top, and if it turns out I get a non-integer, I'll call back to the language-level compiler. **Q**[Michael L. Scott]: One of the basic questions you have for a parallel for loop is whether the iterations are themselves schedulable tasks? Or are they units of work that you pass off to schedulable things? [Antony Hoskin]: We have some primitives for constructing schedulable things. We view this as a language-level issue.

3.3 Concurrency and Transactional Memory in C++: 50000 foot view

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

The 2011 C++ standard first added explicit thread support and a corresponding memory model to the language. This was refined in relatively minor ways in the 2014 version of the standard. This represents significant progress, but some difficult problems, mostly related to the definition of weak memory orders, remain.

Recent work of the committee has focused on the development of more experimental technical specifications. Specifications nearing completion include one for transactional

memory and one specifying a parallel algorithms library. The design of transactional memory constructs was described.

Notes *(collected by members of the audience)*

- Concurrency Study Group (ISO JTC1/SC22WG21/SG1); transactional memory is separate (SG5).
  Tend to be inventive; goal is technical specification capturing community consensus. It describes C++ language semantics, not implementation rules or allowable optimizations. It is not a formal mathematical specification or textbook.
- C++11: added threads API (benefits from lambda expressions), an atomic operations library that relaxes SC through specifying weaker models, and specifies memory model, that is shared variable semantics. Three important aspects are (1) sequential consistency for data-race-free programs, (2) undefined semantics otherwise, and (3) trylock() and wait() may spuriously fail/return – wait() (aside from lock release and re-acquisition) and failed trylock() do not have synchronization behavior. **Q**: How fast is the standard adapted by compilers? **A**: Rather fast, major compilers already comply to it before it gets standardized. **Q**: Is atomic limited in any way? **A**: Standardized optimizations for different base types, otherwise should be implementable with a lock.
- C++14: added rwlock, shared_timed_mutex, and some hand waving for known issues.
- There are a number of conspicuous holes: (a) memory_order_relaxed probably implemented correctly, but needs proper spec (out-of-thin-air is the problem with circular dependencies, but it is unclear what a dependency is) (b) memory_order_consume needs work, (c) async() beginner thread creation facility has a serious design flaw, (d) no concurrent data structures, and (e) incomplete synchronization library.
- Two optional additions to the standard may become candidates for inclusion: parallel/vector algorithms (STL plus a bit) and miscellaneous concurrency extensions: extendedfutures, latches, (OpenMP-style) barriers, atomic smart pointers.
- Longer-term: fix async(), fork-join parallelism, asynchronous computation without explicit continuations ("resumable functions"), low level waiting API: synchronic<T> to wait for a specific value change of a specific variable, more general vector parallelism, and various concurrent data structures.
- Further out: fix memory order spec, mix atomic and non-atomic operations on the same location, better specification of execution agents (beyond bare OS threads) and progress properties.
- Transactional Memory: Tech spec out for balloting. It is experimental: where SG could not decide, both options are included. In C++11 locks require lock ordering, but that's intractable with call backs and hard to define in heavily templatized environments. Use TM as a syntactical way to drop lock-order issues, not a performance point. Syntax: special basic block types synchronized . . . atomic_noexcept/cancel/commit . . . . Not a full replacement for mutexes. Interaction with condition variables is open question. atomic_* act the same in absence of non-transactional accesses or exceptions. TM-semantics: behave as if a global lock was held (but performance expected to be better).
- Different flavors: synchronized allows non-txn synchronization nested (including IO). (This is useful where a lock needs to be acquired inside the transaction in a rare case.) atomic_* has no support for non-tx synchronization. Shared semantics: no nesting, no exceptions; thus single-global lock semantics. Strongly atomic in the absence of data races. Issues around what synchronization is allowed within a transaction. atomic_* have different behavior when an exception is thrown inside the tx. atomic_commit

commits if exception is thrown; atomic_cancel unrolls on a throw (but hard to get the state right): exception is propagated from inside the transaction, but state is rolled back; needs closed nesting due to aborted inner child. atomic_noexcept disallows exceptions. Aborts are problematic. Synchronized vs. atomic_commit: same if code is compatible, but atomic_commit has the compiler check the synchronization freedom of the body (compiler can make stronger static guarantees). tx-safety is part of type. Functions can be declared transaction_safe to be included in atomic blocks.

- Remaining concerns: Optimization/synchronization removal: prove that single thread-local modification can drop the transaction (empty transactions have stronger semantics than no-op, idea to "lock" accessed objects rather); Should transactions logically lock individual objects rather than a single global lock? (Under the single global lock model, an empty transaction still has a semantic effect); Interesting cases: statics, memory allocation is legal inside of transactions in spite of synchronization, some dynamic checking remains for virtual functions.
- Specification keeps growing; more work needed in library interaction. There will be changes. Comments welcome on the draft specification: https://groups.google.com/a/isocpp.org/forum/#!forum/tm

4 Overview of Talks, sorted alphabetically by Speaker

4.1 Tuning X Choice of Serialization Policies

Jose Nelson Amaral (University of Alberta, CA)

Best-Effort Hardware Transactional Memory (BE-HTM) systems require non-speculative fallback policies, or serialization managers, in order to provide a guarantee of forward progress to each transaction. There are several choices of serialization managers that can be used to build a BE-HTM system and most serialization managers have one or more parameters that change their behavior. Several published studies compare two or more alternative serialization managers, but do not explore the tuning of these manager parameters. In this talk I will present evidence, based on experimentation with the IBM Blue Gene/Q machine, to support the claim that the tuning of parameters for a serialization manager is very important and that (1) a fair comparison of serialization managers must explore their tuning; and (2) tuning is essential for each new HTM design point and for each type of application target.

Notes *(collected by members of the audience)*

- A number of papers compare TM policies
- Information about Blue Gene packaging and HTM policy
  16 cores on a chip; L1 16 Kb; L2 32 Mb (where the magic happens!); most of the rest is not TM aware.
  Two modes: short-run and long-run modes; L2 must be aware of all accesses in a txn, so have writes bypass L1; long-run mode invalidates/flushes L1 completely before txn starts; associate a speculative ID with a running txn (there are a limited number of these: 128, may be more than hardware threads). Note: txns can survive OS calls, so this can be an issue in some applications; BG/Q supports orders of magnitude more speculative write state than other implementations.

Failure modes (transaction conflict, capacity overflow, attempt to perform an irrevocable action, design space); conflict detection granularity vs. storage available for speculative state. (Blue Gene allows unusually large transactions at finer granularity than many systems).

- Contention managers x Serialization managers

  If HTM fails, it doesn't tell you who you conflicted with, hence at some point need to use serial execution.

  Simplest policy: go serial if the number of retries exceeds a threshold. Some apps are not sensitive to the threshold, especially in short mode; some performance better (with sharp break points) with larger thresholds, some degrade.

  MaxRetry Policy: Serialize once a certain number of retries is exceeded.

  LimitMeanST policy: favor thread that does the most work, based on karma. Hard to do directly in HW, so track time spent in txn, and serialize if you exceed your max time budget. This often gives a sweet spot for long-run mode, but some apps still degrade, and short-run mode is perhaps more variable. Does any serialization manager dominate the others? No, depends on tuning parameter.

4.2 Complexity Implications of Memory Ordering

Hagit Attiya (Technion – Haifa, IL)

Joint work of Attiya, Hagit; Guerraoui, Rachid; Hendler, Danny; Kuznetsov, Petr; Levy, Smadar; Michael, Maged; Vechev, Martin; Woelfel, Philipp
Main reference H. Attiya, D. Hendler, P. Woelfel, "Trading Fences with RMRs and Separating Memory Models," submitted.

Compiler optimizations that execute memory accesses out of (program) order often lead to incorrect execution of concurrent programs. These re-orderings are prohibited by inserting costly fence (memory barrier) instructions. The inherent Fence Complexity is a good estimate of an algorithm's time complexity, as is its RMR complexity: the number of Remote Memory References the algorithm must issue.

Ensuring the correctness of objects supporting strongly non-commutative operations (e.g., stacks, sets, queues, and locks) requires to insert at least one read-after-write (RAW) fence. When write instructions are executed in order, as in the Total Store Order (TSO) model, it is possible to implement a lock (and other objects) using only one RAW fence and an optimal O(n log n) RMR complexity. However, when store instructions may be re-ordered, as in the Partial Store Order (PSO) model, there is an inherent tradeoff between fence and RMR complexities.

In addition to the main reference above, this talk is also based on [1, 2].

References

1 Hagit Attiya, Rachid Guerraoui, Danny Hendler, Petr Kuznetsov, Maged M. Michael, and Martin Vechev. Laws of order: Expensive synchronization in concurrent algorithms cannot be eliminated. In *Proc. of the 38th Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages* (POPL'11), pp. 487–498, ACM, 2011. DOI: 10.1145/1926385.1926442.

2 Hagit Attiya, Danny Hendler, and Smadar Levy. An o(1)-barriers optimal RMRs mutual exclusion algorithm: Extended abstract. In *Proc. of the 2013 ACM Symp. on Principles of Distributed Computing* (PODC'13), pp. 220–229, ACM, 2013. ACM. DOI: 10.1145/2484239.2484255.

Notes *(collected by members of the audience)*

- Talk models processor influence on shared memory with a reordering buffer located between each processor and the memory.
  Out-of-order execution avoided with fences & atomic operations. Memory model gives abstract conditions on how reordering can happen. Many models and sets of models have been proposed – e.g., Sun's sequential consistency (SC), TSO, PSO, RMO hierarchy. Customary to think of SC as "the good one."
- First result: a mutex algorithm must include a R-W fence (= flush of the reordering buffer) or equivalent atomic operation [1]. Holds for various other non-commutative ops (queues, counters, ...).
- Not all memory accesses are equal – for example, the Bakery Algorithm needs O(1) fences but O(n) accesses, which unfortunately must be remote, that is, served from the shared memory, not a local cache, i.e., with global communication.
- Tournament tree gives O(log n) fences and remote references.
- Without store reordering, one can get by with O(log n) RMRs and O(1) fences [2]. Uses a tree to combine processes into a queue for the lock.
- With store reordering (e.g., PSO), one cannot optimize both RMRs and fences. We can illustrate this with a tree of varying fan-out. The number of levels f determines the number of fences. The fan-out times the number of levels determines the number of RMRs.
- One can prove this is optimal: when stores can be reordered, any mutex algorithm has an execution E (one in which every process gets through the CS once) in which $F_E log(R_E/F_E) \in \Omega(nlogn)$ The proof uses an encoding argument, which captures the order in which processes enter the critical section [Attiya, Hendler, and Woelfel, submitted].
- A nice corollary of this work: there is a complexity separation between TSO and PSO. This suggests that TSO is "nice" in a strong way – analogous to how we have traditionally thought of SC as "nice".
- Also F log (R/F) is $\Theta(nlogn)$, where F is the number fences and R is the number of remote references; this cost is for all n processes to acquire the mutex once.
- Lower bound was instructive in finding the algorithm that meets it.
- **Q**[Nir Shavit]: How much of this translates to search trees? **A**: Not yet clear. We could use a sharper definition of the objects to which the theorem applies.

4.3 Heterogeneous Computing: A View from the Trenches

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

Based on experiences with the Liquid Metal project at IBM Research, I describe the challenges ahead for heterogeneous computing. While compiler and run-time technologies can significantly reduce the complexities, radically different hardware organizations will still require fundamentally different algorithms. This will limit improvements in programmer productivity and keep costs of heterogeneous systems significantly higher. Nevertheless, over the long term heterogeneity will inevitably pervade computing systems.

Notes *(collected by members of the audience)*

- Various kinds of heterogeneity (ISA, scale,performance (thin vs. fat cores), fundamental organization (e.g., CPU vs. GPU), implementation technology, interconnect, language/library, algorithm)
- What users want: single language, compiler deals with platforms transparently, run-time handles variations, code migrates between platforms and responds (dynamically?) to load / grain / input variations.
- IBM Liquid Metal project
  - Single language Lime: Java-like, integration of a degree of static-ness in a dynamic language, data-parallel operators, stream graphs of isolated tasks, fine-grained primitive types, compile-time evaluation, exclusion on a per-platform basis (certain features not implemented on certain platforms), common run-time.
  - Transparent loading and data movement
  - Dynamic replacement
- Reality check
  - Organization tends to dictate the algorithm. So multiple implementations needed even in a single language.
  - Scientific comparison impractical
  - More heterogeneity tends to lead to lower utilization
  - HW specialization subject to obsolescence
  - More specialization = more total cost of operation = lower value
- Example of a challenging situation: Arrays in registers (via scalar replacement) versus an indexed block RAM
- Whither heterogeneity: Dark silicon is our friend; adoption will be slow, due to external factors; algorithmic heterogeneity is inescapable; pressure to minimize variants will remain; things look good with a 30-year horizon.
- Overarching experience: heterogeneity is *really* hard to manage; worth our while to avoid it wherever possible, and shield application-level programmers from it wherever possible.

4.4 Scalable consistency in distributed systems

Annette Bieniusa (TU Kaiserslautern, DE)

Joint work of Bieniusa, Annette; Shapiro, Marc; Preguica, Nuno; Zawirski, Marek; Baquero, Carlos
URL https://syncfree.lip6.fr

Replicating dynamically updated data is a principal mechanism in large-scale distributed systems, but it suffers from a fundamental tension between scalability and data consistency. Eventual consistency sidesteps the synchronization bottleneck, but remains ad-hoc, error-prone, and difficult to prove correct.

In this talk, I introduced a promising approach to synchronization-free sharing of mutable data: consistent replicated data types (CRDTs). Complying to simple mathematical properties (namely commutativity of concurrent updates, or monotonicity of object states in a semi-lattice), any CRDT provably converges, provided all replicas eventually receive all operations. A CRDT requires no blocking synchronization: an update can execute immediately, irrespective of network latencies, faults, or partitioning; the approach is highly scalable and implies fault-tolerance.

Notes *(collected by members of the audience)*

- **Problem:** Data is replicated, failures are common, latency is high. With software transactional memory, the approach is to restart, while in this distributed setting, such rollbacks are more difficult.
- **Solution:** Instead of conflict detection, this work performs correct conflict resolution. One technique they suggest us to use Replicated Data Types (RDTs) of which there are two kinds: convergent and commutative. These are standard objects with the restriction that all modification operations must commute, or if they do not, one has to define a conflict resolution policy (e.g., to reconcile the effect of 2 add's into a set performed at different places that now need to be merged). Hence, they need to manually define semantics of merging when there is a conflict. Another restriction on the APIs of the RDT is that the API cannot both modify the structure and return an observed result at the same time (e.g., add(k)/r interface is not allowed). A specific example of an RDT that was presented is the observe-remove Set (ORset).
  The correctness condition is defined w.r.t a particular data type (i.e., the correctness condition is specific to the data structure). For the ORset, the condition has the form "for all , there exists", which is expensive to check. The work did not present a general correctness condition and it was unclear how to obtain such a condition (say to non-commutativity). They mentioned that they performed dynamic test generation of replicated data types and have done some work in formalizing the different data type implementations.
- Large scale sharing involves data replication which is easy for immutable data but hard for mutable data. Assume: distributed, large-scale, heterogeneous, partial replication, high latency, failures, ... Conflict detection or prevention does not scale – need conflict resolution. Could use only commutative/convergent data types, also called confluent, etc. Primarily data types for containers, but also things like counters and editing of a sequence – point is to achieve conflict freedom by design. Eventual consistency: replicas that have seen the same updates achieve the same state. Discussion of possible semantics for sets, and what happens when different nodes perform add and remove on the same element. Work on defining semantics based on causal history. Many interesting follow-up issues: composing these data types; What about transactions? What semantics? Dataflow programming model; partial replication; bounding divergence.
- **Q**: How generic is the approach that you proposing? The specifications are object-dependent? **A**: There are some fundamental rules that can be abstracted, but the semantics of the objects need to be taken into account.
  Q: What is the relation between linearizability and the specification of the CRDT set? **A**: There are some similarities in the definition but it is clearly not equivalent.
  Q[Michael L. Scott]: It seems that the way you define the semantics for dealing with concurrent updates, affect the actual probability of serializing updates? **A**: This is actually true, but it cannot really be avoided.
  Q: What is the state of the art for checking the correctness of CRDTs? **A**: Some work on static checking, not dynamic. The problem of dynamic testing in distributed settings is actually quite complicated. The problem here does not appear to be significantly different.
  Q: What are the overheads that you need to pay? It seems that you need to transmit a large amount of information capturing chains of updates applied by all replicas among which interactions have occurred. **A**: This can be an issue in fact. There are however solutions that try to address precisely this issue, e.g., (dotted) version vectors.

4.5 Remaining foundational issues for thread semantics

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

Shared memory parallel machines have existed since the 1960s and programming them has become increasingly important. Nonetheless, some fairly basic questions about parallel program semantics have not been addressed until quite recently. Multi-threaded programming languages are on much more solid ground than they were as recently as a decade ago. However a number of foundational issues have turned out to be surprisingly subtle and resistant to easy solutions. We briefly look at a few such issues focusing on finalization in Java, and on issues related to detached threads and std::async in C++.

Notes *(collected by members of the audience)*

- Specifications for multi-threaded languages have improved but there are remaining problems: Out-of-thin-air (OoTA), managed languages and finalization, and C++ detached threads and object destruction.
- Java finalization is problematic. The method finalize() runs after object found unreachable by GC. Java.lang.ref helps but does not fix everything. Only way to work around absence of finalization is to reimplement GC in user code. Problem comes up for example in mixed language case, where Java finalizer frees corresponding C++ object: Finalizable object can be collected while method operating on object is still running, but "this" pointer is no longer live resulting in call to native method on native pointer field with dangling pointer. Various dubious and awkward solutions; synchronized (this) prevents compiler from eliminating dead references. These don't seem viable. Possible solution: "KeepAlive"-decoration. Alternative solution: annotation that prevents compiler elimination of dead references to the type; current favored solution; Rule: annotate if field is invalidated by finalization or reference queue processing.
- Issues of detached threads (= a thread that can no longer be waited for by joining it) and object destruction. Thread is no longer joinable, so resources could be reclaimed when thread terminates. Problem: there is almost no way to guarantee that a detached thread finished before objects it needs are destroyed. No way to guarantee that a detached thread completes before objects it needs are destroyed. More of an C++ issue. Recommendation: do not use detached threads.
- (Reflects insights from many WG21 committee members.) Related issues with async and futures: accidentally detached threads. C++11/14 provides blocking futures only through async. Another issue: async(f); async(g) runs serially. Future's thread can possibly try to refer to stack allocated object which can go away. Various fixes with unintended consequences. Std::async() was a mistake, hoping to fix in C++17. For example by means of separate handles on results and underlying execution agent. (Reflects insights from many WG21 committee members.)

4.6 Parallel JavaScript in Truffle

Daniele Bonetta (Oracle Labs – Linz, AT)

In this talk I presented the support for parallel execution in the Truffle/JS JavaScript engine. Truffle/JS is a JavaScript engine developed using Truffle, a multi-language development framework based on the GraalVM Virtual Machine.

Parallel execution in Truffle/JS is enabled through a combination of compiler optimizations and a built-in runtime based on Transactional Memory. The work leads to several research questions about parallel programming models and runtimes for popular dynamic languages with none or very limited support for parallel execution such as JavaScript, Ruby, and Python.

Notes *(collected by members of the audience)*

- Truffle is a framework for writing high-performance language runtimes in Java. Truffle separates the language implementation from the optimizing system. So, for JavaScript, the language runtime is an AST interpreter using the Truffle API. The interface between the language (e.g., JavaScript) and Truffle is AST nodes and compiler directives: the language implementer has to write an AST interpreter in Java, using the API provided by Truffle. On ordinary Java VMs the language runtime will be executed as a regular Java application. When run using the Graal VM, the AST interpreter will benefit from automatic partial evaluation and improved performance. Wide range of languages are already implemented in Truffle (e.g., JavaScript, Ruby, R, Python). The AST interpreter self-optimizes its nodes to improve them, for instance by determining and propagating type information, inserting appropriate guard nodes, profiles, assumptions (create, check and invalidate, ...). Correct usage of the Truffle API is responsibility of the language implementer. The Graal VM takes care of automatic compilation, de-optimization to interpreter, and re-compilation of the ASTs.
- The JavaScript implementation is quite solid: runs all the ECMAScript 5 standard tests and has increasing support for ECMA 6. It also supports many extensions, including most of Node.JS (i.e., JavaScript for server-side code). Performance of Truffle/JS is comparable to V8.
- Parallelism in JavaScript is important because of the language's popularity. However, the language is single-threaded, and developers are not familiar with threading and synchronization primitives such as locks. We take a simple approach: exposing parallelism via an API with sync/async patterns. The API should be safe (i.e., semantics same as single-threaded JavaScript) and the runtime implementation should be fast (i.e., never slower than sequential). In particular, it should do well on read-dominated, functional, or scope-local workloads. The operation "map" is a simple example for such API.
- We use Truffle to enable parallelization of JavaScript functions by adding to their AST specific synchronization barriers. In this way, dynamic conflict checks are used to back out to sequential implementation. Functions can also be executed in SW transactions to resolve potential conflicts. In this case, the runtime initially assumes that all accesses are read-only or local to a transaction, and Truffle produces optimized code for this case. Guards are used to check when the workload is not read-only or tx-local.

- There is some interesting related work concerning dynamic language runtimes that shares some aspects with our approach. Examples are ASM.JS, PyPy STM, Concurrent Ruby (with STM), and RiverTrail.
- There also are some open **Q**: How can we improve best effort performance of our runtime? How can we generalize the run-time to work with other languages? What VM-level concurrency mechanisms do we need in such a multi-language scenario?

4.7 Robust abstractions for replicated shared state

Sebastian Burckhardt (Microsoft Corp. – Redmond, US)

Joint work of Burckhardt, Sebastian; Leijen, Daan; Fahndrich, Manuel

Concurrent programming relies on a shared-memory abstraction that does not perform well in distributed systems where communication is slow or sporadic and failures are likely (such as for geo-replicated storage, or for mobile apps that access shared state in the cloud). Asynchronous update propagation (a.k.a. eventual consistency) is better suited for those situations, but is challenging for developers because it requires dealing with weak consistency and conflict resolution. In this talk I explain GSP (global sequence protocol), a simple operational model for shared data using asynchronous update propagation. GSP is similar in name and mechanism to the TSO memory model, but is suitable for use in a distributed system where communication and nodes may fail.

GSP supports synchronization primitives sufficient for on-demand strong consistency and update transactions. Moreover, GSP is expressive: all replicated data types and conflict resolution policies we know of (including OT, operational transformations) can be layered on top of it.

Notes *(collected by members of the audience)*

- **Problem:** Client-cloud shared storage where one has persistence, replication and failure. Difficult to program distributed applications in this setting yet used in say mobile computing (e.g., TouchDevelop).
- **Solution:** Adopt a replicated shared state model (client has virtual copy of entire state): easier to program against than say message passing, but then one needs to relax the consistency model, due to the CAP theorem. The particular work proposes a programming model which adapts the TSO weak memory model to distributed systems (the motivation is that this model is best understood and formalized so far). The new model is called GSP: here reads are not synchronous, unlike in TSO. In GSP, there are 2 kinds of stores: confirmed and unconfirmed stores. The system propagates stores to the confirmed buffers of other processes. The runtime system also performs combining of the effects of different operations (stores) performed on the data type (e.g., add(1) + add(1) become add(2)). The approach defines types (called cloud types) which can be used to program with the GSP model and which also avoid running arbitrary server code.
- Multiple users, variety of devices, access to my workspace from all devices (and ability to run code on all these devices). Programming these things is a mess – distinguish between RAM and GUI state, persistent state, decompose into parts that run on stateless server and with persistent storage back-end.

- Lowest level: message passing (actors); next level: shared state: stateless cloud server accessing cloud storage; highest level: replicated shared state: sync when connected, etc.
- Ladder of consistency models: linearizability, sequential consistency, causal consistency, eventual consistency, quiescent consistency. Last three are about asynchronous updates. See his book on Eventual Consistency.
- How close can we get to strong consistency (earlier in the ladder)? Compare with memory models: Not a good match since memory models are for fast communication and no failures; analog of TSO perhaps?; coherent shared memory; store buffer that drains to shared memory when it can; stores asynchronous but reads synchronous; maybe more that local replicas with reliable total order broadcast?; this moves to a view of "memory" as a log of operations, and may need more general view of operations – not just read and write, but can usually be partitioned into reading and updating operations.
- Leads to model with: globally confirmed updates (a sequence), local unconfirmed updates (also a sequence). Can answer local queries from unconfirmed updates but can receive global confirmed items that precede my unconfirmed updates. State when issuing an update may be different from the state when the update takes effect. Can get used to this, but it can also bite you! May need to add various atomic operations (describe as a (pure) update; key trick, since not referring to the state).
- Invented a fixed set of Cloud Types, suitable for this kind of computing. Example: Cloud Table = ordered sequence of rows; can append at end, delete anywhere, and ask whether a row is confirmed. Can implement something like a bank account. Handles editing via entering a row describing the state change and applying a three-way merge operation.
- Reduction = a process of reducing the prefix of a log to a small state; Transactions = explicit push, pull, and confirmed property.
- **Q**: Why not using transactions? **A**: You may use transactions but this is an alternative mechanism. **Q**: There are still some anomalies that can occur in this model. If you observe the state of something in your buffer, you may observe state that can then later on be updated by a remote update.

4.8 The Adaptive Priority Queue with Elimination and Combining

Irina Calciu (Brown University, US)

Joint work of Calciu, Irina; Mendes, Hammurabi; Herlihy, Maurice
Main reference I. Calciu, H. Mendes, M. Herlihy, "The adaptive priority queue with elimination and combining," in Proc. of the 28th Int'l Symp. on Distributed Computing (DISC'14), LNCS, Vol. 8784, pp. 406–420, Springer, 2014.
URL http://dx.doi.org/10.1007/978-3-662-45174-8_28

Priority queues are fundamental abstract data structures, often used to manage limited resources in parallel programming. Several proposed parallel priority queue implementations are based on skiplists, harnessing the potential for parallelism of the add() operations. In addition, methods such as Flat Combining have been proposed to reduce contention by batching together multiple operations to be executed by a single thread. While this technique can decrease lock-switching overhead and the number of pointer changes required by the removeMin() operations in the priority queue, it can also create a sequential bottleneck and limit parallelism, especially for non-conflicting add() operations.

We describe a novel priority queue design, harnessing the scalability of parallel insertions in conjunction with the efficiency of batched removals. Moreover, we present a new elimination algorithm suitable for a priority queue, which further increases concurrency on balanced workloads with similar numbers of add() and removeMin() operations. We implement and evaluate our design using a variety of techniques including locking, atomic operations, hardware transactional memory, as well as employing adaptive heuristics given the workload.

Notes *(collected by members of the audience)*

- Review of prior techniques, report on work presented at DISC 2014. Elimination (get rid of ops that "cancel out"), delegation (server thread does work on behalf of others), and flat combining (does both, with threads taking turns as server).
- Implementation is based on skip list. removeMin is a challenge: little concurrency, flat combining good for this. The operation "add" parallelizes nicely but not so great for flat combining.
- Can we put this together and get best of both worlds? Use typical add for "large" values. Use elimination on smaller values (near removeMin active region). Small values posted to an elimination array, larger ones go straight to skip list. So, in effect have two skip lists, one for smaller values, one for larger. Must adaptively adjust the boundary, in either direction. Boundary movement is done with a reader/writer lock.
- Better scalability than previous methods. If removeMin not as common, scalability can suffer, apparently because of RW lock.
- What about using HTM on that? Simplistic approach has too many conflicts. But when done sensibly (put all CAS for a given add into a single transaction), obtains better speedup. On 8-thread Haswell machine, get maybe 30% better throughput.

4.9 Concurrency Restriction Via Locks

Dave Dice (Oracle Corporation – Burlington, US)

URL https://blogs.oracle.com/dave/resource/Dagstuhl-2015-Dice-ConcurrencyRestriction-Abr.pdf

As multi-/many-core applications mature, we now face situations where we have too many threads for the hardware resources available. This can been seen in component-based applications with thread pools, for instance. Often, such components have contended locks. This talk shows how we can leverage such locks to restrict the number of threads in circulation in order to reduce destructive interference in last-last caches, as well as in other shared resources.

Notes *(collected by members of the audience)*

- Scalability collapse (often because of locks). Difficult to choose best number of threads since adding more degrades performance after a certain point.
- Describes a synthetic benchmark demonstrating performance of various kinds of locks.
- Solution approach: constrain concurrency at any given lock.
- The scalability collapse point may be at more threads than the best-performance point!

- Improvement may come for subtle reasons, such as improved cache miss behavior in the critical section when fewer threads are running. Competition can be for a variety of different resources, even in hardware. Effects are amplified with transactions because of transaction aborts' wasted work.

4.10 Why can't we be friends? Memory models – A tale of sitting between the chairs

Stephan Diestelhorst (ARM Ltd. – Cambridge, GB)

With my relatively recent transition from a strong memory model (AMD64, similar to TSO) to a weakly ordered architecture (ARM), and some exposure to software while working in a HW company, I would like to put out some points for discussion on why useful weak HW models cannot keep their reasonable properties once they are lifted to a language level programming model without causing prohibitive amounts of fences or other ordering primitives (such bogus branches) behind every global memory access.

I think in the many-core programming world, we can make coherence stay, but I would argue that keeping a strong memory model might not be possible. Therefore, we ought to see what makes using weak memory models hard on today's and tomorrow's weakly ordered machines and fix the semantics for future architectures.

Notes *(collected by members of the audience)*

- Stephan recently moved from AMD to ARM, and thus from a strong to a weak memory model. ARM has weak ordering and no store atomicity (some processors may see a store while other do not yet). At the same time: address dependency preserves order, data dependency preserves order, and control dependency orders subsequent writes.
- The dependencies prevent (at least certain) out-of-thin-air cases, but compiler optimizations can change or remove dependencies! Naive examples violate intuition, but one can modify the examples in ways that make the intuition go away, and thus help to illustrate why the Java memory model has been so difficult to nail down. Can't afford to make every read/write "sacred", so role of compiler, language definition and CPU spec all gets complicated. (We see a 3-way dance between architecture, language, and compiler.)
- Hard to fix at any single point of language, compiler, and architecture. Architecture isn't likely to change. For ARM, "fixes" would mess with goal to be small, fast, and energy efficient. Compilers are unlikely to change either [slides skipped for time].
- Hypothesis: we need to fix the languages. But how? Force a function to implement dependence of all output on all inputs? Probably not: can't implement Haskell or R or anything else that wants to avoid fully evaluating everything. Force a fence or conditional branch after every load? Use use ARM v8 ld.acq a lot?
- **Q**[Charles E. Leiserson]: What about "observer functions"? These indicate, at every point, what write you'd see if you chose to read [Frigo & Luchangco]. They're dependency-based. They avoids anomalies where threads A and B go through a common state (and thus should have equated views) but didn't actually *look* at anything – Heisenberg-ish. **A**: Dependences serve to break "cycles of self fulfillment" in out-of-thin-air examples. **Q**[Torvald Riegel]: It's not clear you can fix everything at the language level: the compiler

wants to "change the program". **Q**[J. Eliot B. Moss]: And even if the language is "right", programmers get it "wrong". **Q**[Hans-J. Boehm]: The real problems arise when the program fails to specify enough. What are the semantics then? **A**: It would be really useful to have compelling "real world" examples – things architects would accept as "more real" than the usual "brain teaser" examples.

4.11 Application-Directed Coherence and A Case for Asynchrony (Data Races) and Performance Portability

Sandhya Dwarkadas (University of Rochester, US)

Joint work of Dwarkadas, Sandhya; Shriraman, Arrvindh; Zhao, Hongzhou
Main reference A. Shriraman, H. Zhao, S. Dwarkadas, "An application-tailored approach to hardware cache coherence," *Computer*, Special Issue on Multicore Coherence, 46(10):40–47, 2013.
URL http://dx.doi.org/10.1109/MC.2013.258

I described an application-tailored approach to supporting coherence in hardware at large core counts and present two complementary approaches to scaling a conventional hardware coherence protocol. SPACE is a directory implementation that reduces directory storage requirements by recognizing and storing only one copy of the subset of sharing patterns dynamically present at any given instant of time in an application. Protozoa is a family of coherence protocols designed to adapt the data transfer and invalidation granularity to match the spatial access granularity exhibited by the application. Compared to conventional one-size-fits-all approaches, these designs match coherence metadata needs and traffic to an application's sharing behavior, allowing an application's inherent scalability to be leveraged.

I also showed empirical data to make a case for architectures, runtimes systems, and parallel programming paradigms to allow applications to tolerate data races (operate asynchronously) where desired and to design for performance portability. I discussed our efforts at the operating system, runtime, and programming paradigm level to enable automated techniques incorporating both application and hardware knowledge of sharing and memory access behavior for resource-commensurate performance.

Notes *(collected by members of the audience)*

- The talk is concerned with different ways of implementing coherence. Whether it is done in SW or HW is, at least in principle, irrelevant. User-defined consistency can be useful. Distributed domains often do not require strong consistency.
- Scalability enhanced by metadata storage compression. Conventional Full Map Directory is rather large as you need 1 bit per cache line. 1 bit per processor per line, 64 byte line, 128 cores, means directory is 1/4 of shared cache size. Compression: multiple blocks may have same sharing pattern. Compress the information in application-specific ways. Limited number of sharing patterns allow compression in a directory Decouple sharing patterns from cache blocks for smaller directory. Experiments on benchmarks shaw very high rates of sharing. Uses only patterns found in the applications. Sharing PAttern-based CoherncE (SPACE): n to log n bits to describe sharing. Sublinear scaling. 57 % area, 50 times energy cost at 16 cores. Can be a lot of waste in cache lines – perhaps as low as 21% of a 64 byte line is accessed. An alternative to SPACE are "shadow tags" that are similarly compressed but are more energy costly. Want adaptive granularity.

- Communication efficiency. Key to a good communication efficiency is granularity control. The Protozoa adaptive coherence granularity protocol reduces the communication demand significantly. Adapt coherence traffic to sharing behavior. Eliminate read-write and write-write false sharing. Metadata storage comparable to conventional schemes. Reduces on-chip traffic by 26%, traffic 37% across 28 benchmarks. Overhead is a function of application behavior.
- Case for data races. Some applications converge despite data races. Asynchronous algorithms. Successive over-relaxation, SVM. Converges even in the presence of data races. Here the case is being made that the compiler needs to allow for having racing reads and writes without enforcing consistency. This enables noticeable speedups. Synchronization needed to know whether data is current round or previous can be eliminated.
- Genome analysis: When looking at clusters of multi- or many-cores, overall performance very much relies on locality as can be enforced through pinning. Need to allow programmer to say this and have it carried out efficiently. Can detect sharing and adjust things in the OS to improve scalability. Performance portability remains a challenge. Proposed Linux runtime monitor detects sharing in applications.

4.12 Future of Hardware Transactional Memory

Maurice Herlihy (Brown University, US)

Maurice Herlihy led a discussion of the future of hardware transactional memory (HTM) focused around three questions.

First, are progress guarantees a prerequisite for widespread adaptation of HTM? Opinion was divided: some felt that such guarantees were necessary, but many felt that lock elision provided enough of an alternative that stronger guarantees were not necessary.

Second, is the ability to issue non-speculative instructions from a hardware transaction essential to constructing hybrid schemes that combine hardware and software? Here, the opinion was mixed. IBM's Power architecture provides the ability to suspend a transaction to execute a limited set of non-transactional operations, but there was some question whether that mechanism was to inefficient to use.

Third, there was a broad consensus that lock elision was an effective technique, but only if the application programmer could control the retry policy, implying the Haswell's built-in lock elision mechanism was too inflexible.

Finally, there was widespread agreement that better debugging support was needed.

Notes *(collected by members of the audience)*

- Is HTM doomed without progress guarantees? Too many code paths: fast path HTM, slow path on HTM abort, slow-slow STM-only version. How would you state a guarantee? Different systems may need different design points.
- Did the Haswell bug ruin everything? Is HTM just too hard to get right? No correctness bugs reported for IBM implementations.
- Is Hybrid TM hopeless without non-transactional operations? But then what do non-transactional writes mean? (We could argue for a decade!) Immediate NT reads,

immediate NT writes, and delayed (on-commit attempt) writes are all possibilities. Maybe logging read/write sets would be better HW primitive?
- What should our HW "ask" be?
- HTM and memory management? Hazard pointers make the common case expensive (because of memory barrier at each traversal). May choose transaction size adaptively to match appropriate degree of speculation - do multiple state transitions as a single one, speculatively.
- Is lock elision unloved? Customers want to code their own retry policies ...

4.13 On verifying concurrent garbage collection for x86-TSO

Antony Hosking (Purdue University, US)

Joint work of Peter Gammie; Hosking, Antony; Kai Engelhardt

I reported on the machine-checked verification of an on-the-fly, concurrent, mark-sweep garbage collector in Isabelle/HOL. The collector is state-of-the-art in that it is designed for multi-/many-core architectures with weak memory consistency. The proof explicitly accounts for both of these features, incorporating the x86-TSO model for relaxed memory semantics on x86 multiprocessors. To our knowledge, this is the first fully machine-checked proof of such a garbage collector. We couch the proof in a framework that system implementers will find appealing, with the fundamental components of the system specified in a simple and intuitive programming language. The framework is sufficiently detailed that correspondence between abstract model and assembly coded implementation is straightforward so as to allow formal refinement from model to implementation.

Notes *(collected by members of the audience)*

- Proved essentially that garbage collector on multi-/many-core architecture doesn't collect non-garbage (i.e., correctness). Concurrent, on-the-fly mark&sweep collector that does not compact. Fragmentation tolerant (cache line size fragments). Schism CMR RTGC.
- Challenges: Concurrent system; memory is not sequentially consistent (Sewell et al.'s x86 TSO model); mutators are not data race free; model is fairly realistic; and formulating model and invariants in a manner friendly to systems people.
- Model: mutator processes, collector process, system component (handles the HW memory model). Tricolor abstraction used; marking propagates a grey wavefront across the reachable heap.
  Collector techniques: insertion barrier incremental update; deletion barrier snapshot; white allocation when not marking; black allocation when marking.
  Collector code structures: Series of initial handshakes that establish some invariants; mutators put roots into a worklist; collector processes worklist, inserting new objects as necessary; termination check phase (grab any more fodder from mutators); reclamation sweep phase.
  Marking: use a CAS to mark (and claim) an object, then add to worklist.
- Modeling x86-TSO: from Sewell et al.; Buffers writes in order; reads from the buffer; bus lock for larger atomic operations.

- Proved correctness of model of the code. Boundedness of TSO buffers was discussed. They are not. Proof has to consider reachability of pointers in TSO buffers. The roots may be in the write buffer. Snapshot ensures reachable white objects are reachable from a grey object. Write barriers insure greying, and the CAS causes an immediate effect. Model TSO via message processing to system process. Proof uses Lamport-like techniques from 70s and 80s.
- Modeling language: imperative language with message passing, CIMP. Original code turns into something very similar in CIMP.
- Invariants: Universal – only data; Local – talk explicitly about control locations ("pc values"); at_p l s – process p is at label l with state s. Push all invariants across all transitions (in practice some things end up being local invariants; can use full HOL in doing the proof).
  Constructing the invariants: Track what the mutators know about the current GC phase; order of writes to different variables mostly doesn't matter. It can slice the system and get small relations over smaller parts. Worst operation is marking (of course!). TSO subtlety: deletion barrier marks a reference that is read, so what exactly is read? Finally use tricolor invariants.
- Proof technique: monster induction over all states. Tactics eliminate the trivial cases, allowing focus on the interesting parts. Annoying thing: have to "carry" invariants across the TSO buffer to memory, i.e., invariants get stated twice, once in a local form and again universally.
- Intended to eventually also model ARM Power. Feasibility not yet clear. Lack of store atomicity on ARM is likely to complicate matters. Similar proof for C11 model possible?
- Result takes 2 CPU hours of proof. Only a safety property. Liveness not yet proven. Would like to eventually prove that all objects are eventually collected. This model does not really allow thinking about performance, only correctness.
- **Q**: Do redundant work instead of CAS when marking? **A**: Probably wouldn't require much additional work.

4.14 Efficiently detecting cross-thread dependences to enforce stronger memory models

Milind Kulkarni (Purdue University, US)

Joint work of Kulkarni, Milind; Bond, Michael; Sengupta, Aritra; Biswas, Swarnendu; Zhang, Minjia; Cao, Man; Salmi, Meisam Fathi; Huang, Jipeng
Main reference M. D. Bond, M. Kulkarni, M. Cao, M. Zhang, M. Fathi Salmi, S. Biswas, A. Sengupta, J. Huang, "OCTET: Capturing and controlling cross-thread dependences efficiently," in Proc. of the 2013 ACM SIGPLAN Int'l Conf. on Object Oriented Programming Systems Languages & Applications (OOPSLA'13), pp. 693–712, ACM, 2013.
URL http://dx.doi.org/10.1145/2544173.2509519

In this talk, I discussed two results from an ongoing project.

First, I described a system called Octet, that provides detection of dependences between threads. Octet operates by associating a thread ownership state with every object. Prior to accessing an object, Octet checks the state of the object; an access incompatible with the object's state implies a potential cross-thread dependence. Octet uses an optimistic protocol for managing these ownership states: most accesses do not require state changes, and Octet

requires no synchronization for these accesses; synchronization is only required when the state must change.

Second, I described how we use Octet to enforce a stronger memory model, statically bounded region serializability. The memory model considers each thread to be a sequence of statically-bounded regions; execution appears to be some serial interleaving of these regions. We implement this memory model through a combination of Octet to provide two-phase locking (ensuring region atomicity) and compiler transformations to support region rollback to avoid deadlock.

Notes *(collected by members of the audience)*

- Safely and efficiently detecting cross-thread dependences (think: ownership tracking). At least two threads accessing an object and at least one writes to the object. Safely: time of check vs. time of update of the meta data; Efficiency: need to protect the meta data. Safe to do an atomic operation on an object's metadata before each use. But that gives a 3x slow down. **Q**[Charles E. Leiserson]: Is 3x slowdown is reasonable for debugging? **A**: It is, but if I want to do record/replay or use it for STM, I want it to be faster.
- Octet protocol: Fast path checks current ownership (meta data is already in a "good state"), and if ok, proceed without expensive synchronization operations. But if state is not what I need, there is a more complex, instrumented, bias-locking slow path (presumably rare). The slow path causes a thread to wait until the object's current owner reaches a safe point.
- Proposal is Statically Bounded Region Serializability (SBRS). Optimize heavily within a region. Static bounding is not a limitation: dynamic regions do not lead to larger regions in practice.
- Capturing of dependencies can be be used for atomicity checking, STM, record & replay. Example: implementing a stronger memory model (using Octet) using a 2PL approach and combined with some rollback mechanism. Overhead of benchmarks: 13% cost for Octet with no coordination, 26% with coordination, 30% for SBRS.
- Slide 9: **Q**[Stephan Diestelhorst]: What is the initial state of an object? **A**: Could be invalid. Could be owned by the allocator.
  Slide 11: **Q**[Michael L. Scott]: Is this like asymmetric/biased locks? **A**: Yes. Essentially, biased read/write locks on every object.
  Slide 13: **Q**[Stephan Diestelhorst]: Do you require thread 1 to wait until it gets to the safe point? **A**: Thread 1 does not have to do unconditional wait. There is a potential for deadlock. Trick: while a thread is spinning, it can give access to other threads. This scales. T1 does not need to know which object it needs to access. [J. Eliot B. Moss]: Moss: T2 must know that T1 is still the youngest. The CAS serves the function of saying "I need this object no other thread should have access to it" **Q**[Martin T. Vechev]: Can I encode arbitrary state transition, such as type state? **A**: Yes, we believe we can encode other types of state. **Q**[Martin T. Vechev]: Is the protocol specific to this automata? **A**: It is not specific to this automata. Coordinate with a share state requires you to coordinate with all other threads.
  Slide 18: **Q**[Stephan Diestelhorst]: Are function calls allowed in a region? **Q**[Jose Nelson Ameral]: Does inline changes the atomicity properties of the program? If the programmer writes a set of statements into a function expecting those to be atomic, then after the compiler inlines that function the atomicity property would be lost. **A**: You can use synchronization annotations to ensure that the atomicity is enforced.

4.15 Hardware Transactional Memory on Haswell-EP

Viktor Leis (TU München, DE)

Intel's Hardware Transactional Memory feature TSX was initially launched for Haswell desktop CPUs with 4 cores. Only recently, systems based on Intel's mid-level server platform Haswell-EP became available. Haswell-EP supports two sockets and up to 72 hardware threads. On such many-core systems, transactional memory is both more desirable and more challenging.

In this talk, I will present a number of experiments on a dual-socket Haswell-EP system with 28 cores. The results show that TSX can indeed achieve good performance and scalability on NUMA systems with many cores. However, there are a number of non-obvious pitfalls that must be avoided.

Notes *(collected by members of the audience)*

- The talk is concerned with performance evaluation of HTM in the context of data bases.
- Server-class Haswell is Intel's mid-level server platform. It comes with up to 18 cores per socket, 72 HW threads with 2 sockets, and supports TSX (must be explicitly enabled, due to "the bug").
- Experimental setup: global fallback lock using Hardware Lock Elision (HLE); Alternative: implemented by RTM (still lock elision); Workload: (a) adaptive radix trie (fanout 2-256), as for a main memory DB, (b) random lookups in 64M entry trie, (c) 64M random inserts into initially empty trie (hard to turn into a totally non-blocking data structure; typically touches 10 to 12 cache lines; should not lead to capacity issues in most cases). Measurements on Intel Xeon E5-2697 v3 on 2 sockets 14 cores each 2 HW threads each system. (One interconnect ring per 7 cores, two rings are linked, then those are linked across socket with QPI, rw_spin_lock totally does not scale, no sync does).
- A conflict-free look-up benchmark with locking test shows bad speedup with read-write spin lock. The theoretical peak is almost 100M ops/s locks bring it down to less than 25M. An atomic counter does not do much better. Built-in HLE does not speed up, but customized HTM performs much better, but sensitive to the restart policy. If you are willing to do enough restarts, get speedup almost as good as no sync. The more restarts, the better. 7 or more restarts is scalable. Why? In this case, aborts are mostly spurious, so fallback is harmful.
- For random inserts, which do have conflicts, the scalability was dominated by the memory allocation policy, with malloc the worst (no speedup), tcmalloc better (scaling stops at 28 threads), with a combination of memory pre-allocation and zeroing out doing the best.
- The NUMA behavior was tested by placing threads on 1 cluster, 1 socket, and 2 sockets. Speedups were roughly the same, even though the actual time was faster for more local setups. (Lookup: better speedup by spreading threads around, both across clusters and across sockets; Insert: same effects). Overall rate better when memory being used is restricted to unit running the threads. The more local the threads are the better the performance is; the scalability stays the same, though. HTM works over NUMA.
- Conclusions: HTM scales to NUMA, built-in HLE does not scale. Despite very few collisions, or maybe because of very few infrequent collisions, large restart numbers (>20) seem essential. Kernel traps within the transactions have deadly effects.

4.16 What the $#@! Is Parallelism? (And Why Should Anyone Care?)

Charles E. Leiserson (MIT – Cambridge, US)

Many people bandy about the notion of "parallelism," saying such things as, "This optimization makes my application more parallel," with only a hazy intuition about what they're actually saying. Others cite Amdahl's Law, which provides bounds on speedup due to parallelism, but which does not actually quantify parallelism. In this talk, I reviewed a general and precise quantification of parallelism provided by theoretical computer science, which every computer scientist and parallel programmer should know. I argued that parallel-programming models should eschew concurrency and encapsulate nondeterminism. Finally, I discussed why the impending end of Moore's Law – the economic and technological trend that the number of transistors per semiconductor chip doubles every two years – will bring new poignancy to these issues.

Notes *(collected by members of the audience)*

- Parallelism: simple model of parallel computation: DAGs. Strand is a serial chain of executed instructions. Dependency is a necessary ordering relationship. Usual notion of forks and joins in the DAG. Programming language can express these. Can schedule dynamically at run time. Amdahl's Law – it does not of itself quantify parallelism, only potential speedup. Can model the time required on P processors using the task DAG. Work Law: longest path gives min time required. Span Law: largest number at once gives max speedup. Theoretical model says super-linear speedup is not possible; in reality, other effects can sometimes produce super-linear speedup. Still, can describe max (theoretical) speedup as ratio of time required for one processor to time required by an unbounded number of processors. If you use more processors, they cannot be fully utilized. Can prove that Cilk's scheduler gets near perfect linear speedup if the parallelism substantially exceeds the number of processors available. Enables a straightforward scientific approach to speeding up your programs.
- Concurrency: Situation is much more complex. Concurrency introduces interactions between threads that often reduce available parallelism. Theoretical models not very strong. Should eschew concurrency on most programming. Need to move away from concurrency toward determinism. Concurrency essential to implementing the platform. But best done by experts, once. Historical analogy to "Goto Considered Harmful", which led to structured control constructs (arguably "complicated" things: goto is "simple" – but it was good in the end).
- Why care? Moore's Law. At 14nm now; will get down to 5nm but likely not much more (at least not economically). Limit about 2020 or 2022 according to Bob Colwell. Solution: replacement technologies can still help, but they're going to be software technologies: computer science.

4.17 Bringing concurrency to the people (or: Concurrent executions of critical sections in legacy code)

Yossi Lev (Oracle Labs., US)

In this presentation I have discussed a few of the challenges that people are likely to encounter when making critical sections in legacy code executed concurrently (e.g. using transactional memory), and provide some examples of how some of the data structure and infrastructure work we've been doing in the last few years can help addressing these challenges.

Notes *(collected by members of the audience)*

- How do we make HTM useful to as many people as possible in the near term (as well as the long term)? Need near-term benefits to motivate vendors to keep investing.
- Code in most critical sections was not designed to run concurrently! Need to avoid writing same value: turn x = v; into if (x != v) x = v; , if it is likely that x==v will hold in most cases. Clearly, we do not want all writes to become conditional, but for some writes this can be very effective, esp. for variables whose types have a small number of values – booleans, node color in RB tree, phase indicators, ...
- Minimize time period from a write to the end of the critical section – will reduce conflicts/aborts; pad data that frequently changes, to avoid false conflicts;
- Counters: shared counters in critical sections are common and are often the cause for failure of optimistic approaches (such as HW transactions). Per-thread counters may lead to bad performance if the total value (i.e., the sum of these counters) needed to be calculated frequently; per-core/node counters work better, as there is only a small set of counters to read, and this set is known upfront. In some cases the update to the counter dos not have to happen atomically with the rest of the critical section operation, in which case a separate fetch-and-add outside of the critical section, can help. Can sometime also use a solution that increments counters probabilistically, if approximation will do [1]. Finally, sometimes you do not have to know the exact value of the counter, but simply some property of it – e.g., use SNZI: Scalable Non-Zero Indicator, to only know if it is 0 versus not-0. SNZI works quite well with HTM.
- **Q**[Hans-J. Boehm]: May want to get compilers to do many of these optimizations. Also need to prevent compiler from undoing them if you've done them by hand. **A**: Declaring variables "volatile" helps prevent the compiler from undoing, but at the longer term we want compilers to optimize code that is executed inside a transaction differently.

References

1 Dave Dice, Yossi Lev, and Mark Moir. Scalable statistics counters. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA'13, pages 43–52, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1572-2. DOI 10.1145/2486159.2486182.

4.18 Towards Automated Concurrent Memory Reclamation

Alexander Matveev (MIT – Cambridge, US)

Joint work of Alistarh, Dan ; Eugster, Patrick; Herlihy, Maurice; Leiserson, William M.; Matveev, Alexander; Shavit, Nir
Main reference D. Alistarh, P. Eugster, M. Herlihy, A. Matveev, N. Shavit, "StackTrack: An automated transactional approach to concurrent memory reclamation," in Proc. of the 9th European Conf. on Computer Systems (EuroSys'14), pp. 25:1–25:14, ACM, 2014.
URL http://dx.doi.org/10.1145/2592798.2592808

The concurrent memory reclamation problem is that of devising techniques to allow a deallocating thread to verify that no other concurrent threads, in particular ones executing read-only traversals, have a reference to the block being deallocated. To date, there is no satisfactory solution to this problem: existing tracking methods like hazard pointers, reference counters, or epoch-based RCU, are either prohibitively expensive or require significant programming expertise, to the extent that using them is worthy of a conference paper. None of the existing techniques are automatic or even semi-automated.

This research project will take a new approach to concurrent memory reclamation: instead of manually tracking access to memory locations as done in techniques like hazard pointers, or restricting accesses to specific methods as in RCU, we plan to use the operating system and modern hardware's transactional memory tracking mechanisms to devise ways to automatically detect which memory locations are being accessed, and allow accesses in any point in the code. This will allow to design and implement a new class of automated concurrent memory reclamation frameworks, making them relatively easy to use and allowing them to scale, so that the design of such structures can become more accessible to the non-expert programmer.

Notes *(collected by members of the audience)*

- Consider case of logical-then-physical deletion in a (concurrent) linked list. Hand-over-hand locking has too much overhead therefore use unsynchronized traversals. But that complicates memory reclamation. How do we know when a node can be reclaimed? Need to track both thread-local (stack) references and global references, e.g., passed through task queues in the heap.
- Current solutions: reference counting, hazard pointers [Maged Michael, Herlihy et al., Braginsky et al.] or epoch/RCU-based.
- Traditional approaches worry only about the thread-local case. "Extended memory reclamation" addresses the global exchange references as well – things that point to nodes that have been removed from a shared abstraction but are not gone from the system (and might be added back in), and thus should not be reclaimed. Need to distinguish "permanent" references between node of a data structure.
- Stack Track system uses HTM to scan thread stacks for transient references. Automatically adapt and split these into smaller transactions when capacity aborts are detected. If this is not an option (no HTM, or too size-restricted), can emulate in SW. Do stack scan of a thread to find interesting pointers. Software StackTrack as fallback for HTM StackTrack.
- Doesn't have a general solution yet for global exchange variables. Currently using a visible pool of global references. Change mappings on pages to prevent concurrent changes.

4.19 Portability Issues in Hardware Transactional Memory Implementations

Maged M. Michael (IBM TJ Watson Research Center – Yorktown Heights, US)

Recent hardware transactional memory implementations that became commercially available in recent years have differences in their architectural and performance features. These differences can lead to programming pitfalls and raise functional and performance portability issues. There is a risk that HTM users learn the wrong lessons about HTM in this early stage of its commercial availability, and influence future HTM designs and uses based on such lessons. In this talk, I discussed differences among HTM implementations and potential pitfalls.

Notes *(collected by members of the audience)*

- This talk presents an overview of current HTM designs and discusses interfacing issues. Differences between main architectures.
- Z: has constrained transactions that restrict the operations that can occur within a transaction – the other systems don't. An obstruction-free transaction. If you follow certain constraints, then in the absence of conflicts, the transaction will eventually complete. No need for failure handler. Not portable to Intel or Power8. In effect, really small txns.
- Haswell: has built-in hardware lock elision – the others don't. Backwards compatible. Library may not have a standard "I'm free" value. Hazard: reading lock bit in critical section can see unexpected unlocked value. RTM does not impose many unique constraints on coding.
- Power8: Power8 has suspend/resume. Allows even system calls within the transactions. Suspends current transaction, allows non-transactional execution until resume. Some restrictions: while suspended, cannot write to memory read by transactions. Can check for transaction failure while suspended, but handlers called only on resume. Non-transactional loads good for loop parallelization, hybrid TM, reducing read set. Non-transactional stores good for debugging, conflict resolution, must be used with care. Read and write sets can be explicitly controlled through switches between suspend and resume states. Resolution is explicit. Nested entry primitives can lead to stack corruptions upon abort. This requires explicit programming support.
- (Non-)Portability examples: (a) TLE does not work well with indiscriminate non-transactional stores (e.g., to thread stack). Start in function, then return before the end of the transaction, pop stack, write to stack frame non-speculatively. If the transaction fails, we end up with a corrupted stack. Z supports non-transactional stores. Invisible to other threads until the transaction ends. They become visible even if the transaction aborts. Useful for debugging, not for communication. (b) Power8 Rollback only transactions. Single-thread speculation, no shared data, no conflict detection, no order guarantees. Rollback Only Transactions does not do any conflict detection; that are intended for single threaded speculation only. However, they can be interleaved with regular transactions. In that case they are treated as atomics. (c) XEND/TEND outside of transactions: some processors fault, others don't, so not convenient for portability. Cannot call Haswell XEND outside transaction (must check first), but ok to call TEND

in Z and Power8. (d) Big variation in capacity. Encourages different programming styles. (e) HTM caching policies also important and different across systems, but hidden from programmers. (Handling for Power8 depends on the mode but it typically bypasses L1.) (f) Variation in overhead of using transactions (compared with atomic ops). Depending on the application circumstances, the overhead can easily be around 50%. Different conditions on different systems.

4.20 Local Combining on Demand

Erez Petrank (Technion – Haifa, IL)

Main reference D. Drachsler-Cohen, E. Petrank, "LCD: Local Combining on Demand," in Proc. of the 18th Int'l Conf. on Principles of Distributed Systems (OPODIS'14), LNCS, Vol. 8878, pp. 355–371, Springer, 2014.
URL http://dx.doi.org/10.1007/978-3-319-14472-6_24

Combining methods are highly effective for implementing concurrent queues and stacks. These data structures induce a heavy competition on one or two contention points. However, it was not known whether combining methods could be made effective for parallel scalable data structures that do not have a small number of contention points. In this paper, we introduce local combining on-demand, a new combining method for highly parallel data structures. The main idea is to apply combining locally for resources on which threads contend. We demonstrate the use of local combining on-demand on the common linked-list data structure. Measurements show that the obtained linked-list induces a low overhead when contention is low and outperforms other known implementations by up to 40% when contention is high.

Notes *(collected by members of the audience)*

- Lock Combining = Threads help each other instead of contention on a single lock. Combining waiting pushes and pops: one thread grabs the lock then does all the pending operations. Known to speed up stacks and queues.
- Lock combining for sorted linked list implementation of set. Effective combing when there is contention: elimination of inserts and removing of duplications (same key). Locks on individual nodes, but designed so that contains and search do not need to lock. Apply combining on contended locks: holder does all operations that queue on the same lock; can also eliminate complementary operations (interestingly, regardless of order (because order isn't really defined)). When a waiter wakes up, may need to check the situation; the combiner could also wake him up earlier, telling him that he needs a different lock. Doing this on demand, i.e., only when there is contention. Don't do combining if you get the lock immediately. Combining is local: happens only on the contended lock.
- What about operations that require multiple locks (such as remove)? Split into separate sub-operations, each acquiring one lock. Note that first lock remains held while second lock is acquired and its sub-operation done. Need to preserve serializability. Preserve two phase locking.
- Have integrated this with reentrant Java locks [Doug Lea]: use the waiting-thread list of the lock instead of duplicating the wait queue.

- Performance comparison with other implementations of same data structure on machine with 64 HW threads; lock-free list eventually wins at high contention, lock combining list is competitive or dominant in lower/moderate contention situation. Lock-free does better for high thread counts. Better than combining with single global lock.
- [Yossi Lev]: Skip lists would be an interesting candidate. [Martin T. Vechev]: There are variants that don't use the deleted bit. May be easier.

4.21 Current GCC Support for Parallelism & Concurrency

Torvald Riegel (Red Hat GmbH – Grassbrun, DE)

I gave an overview of the new features related to parallelism and concurrency in the upcoming release 5 of the GNU Compiler Collection.

Notes *(collected by members of the audience)*

- Presentation refers to GCC 5 which is in stabilization phase, get from SVN.
- Parallelism in C++: OpenMP4 support (including offload to Xeon Phi and Nvidia PTX back end), OpenACC, and Cilk Plus ([Charles E. Leiserson]: metadata is missing. Intel mostly working on it.)
- C/C++ memory model fairly well supported. C++11 memory model is complete, frontend parses everything; testing of the front-end is done.
- TM support in C++ experimental. Older version of the TM for C++ spec is implemented. _transaction_atomic = atomic_commit and _transaction_relaxed = synchronized. Some additional attributes for the tm-safety annotations; additional control for bypassing instrumentation, and manually specifying both versions directly; new feature: multiple/different code paths for instrumented and non-instrumented code, possible to plug-in custom libraries.
- TM Runtime library libitm supporting different STMs and HTM for a few architectures. STM: various algorithms, running on most ISAs, including ARM, Aarch64. HTM version for Power8, s390, and Intel HTM.
- **Q**: Any users of this out there? **A**: Not aware of users outside of experimentation or research; but that does not mean there are none. **Q**: What is transaction safe, list of functions from the standard library? **A**: ISO C++ study group 5 members are going through the API, marking things as safe. Problem: Claiming functions to be transaction_safe might restrict future implementations.

4.22 Forward progress requirements for C++

Torvald Riegel (Red Hat GmbH – Grassbrun, DE)

I presented forward progress requirements for C++ implementations that I have proposed to the ISO C++ committee. These requirements define what progress means in C++ and

what the differences are between, for example, OS threads and parallel tasks running on a bounded thread pool.

Notes *(collected by members of the audience)*

- "Execution agents" (EA) proposed for a Technical Specification (TS) (potential inclusion in a future versions of C++). EAs are threads of execution with different execution properties (e.g., light-weight threads, OS threads, etc.).
- Problem: Current spec reads "every unblocked process eventually makes progress". But it is open what "progress" and "unblocked" mean.
- EAs needed to talk about and specify forward progress, but with potentially weaker guarantees than OS thread. Talk describes EAs on an abstract level. Classes of progress: concurrent, parallel, and weakly parallel. Bootstrapping progress: "boost-blocking".
- C++ semantics defined in terms of an abstract machine. As-if rule = must act observably the same as the abstract machine. Progress defined in terms of steps, resulting in termination, access/change to a volatile object, or sync/atomic operation. Progress means executing a step. Also delimits what compiler writers can elide. Blocking operations and IO may be conceptually implemented as busy-waiting on a condition. **Q**[Michael L. Scott]: When writing while(); (infinite loop), would that make progress? **A**: undefined behavior.
- Flavors of EAs, i.e., classes of progress: Concurrent = every EA will progress; Parallel = every EA will progress once it has executed its first step (this allows bounded thread pools as implementation); Weakly parallel = no guarantee, but see boost-blocking (this allows non-preemptive execution and lock-step execution (such as SIMD)).
- Boosting progress: form groups of agents, if P uses boost-blocking to wait on a group of agents G, then agents in G will have at least one boosted to P's level of guarantee, if it is higher. Boost blocking vaguely like priority inheritance with transitivity.
  Example implementation: Concurrent EA = one OS thread for each EA plus round-robin OS scheduler.

4.23 Self-tuning Hardware Transactional Memory

Paolo Romano (INESC-ID – Lisboa, PT)

Joint work of Romano, Paolo; Diegues, Nuno
Main reference N. Diegues, P. Romano, "Self-tuning Intel transactional synchronization extensions," in Proc. of the 11th Int'l Conf. on Autonomic Computing (ICAC'14), pp. 209–219, USENIX Association, 2014.
URL https://www.usenix.org/conference/icac14/technical-sessions/presentation/diegues

Efficiency of best-effort HTM (like Intel TSX) is strongly dependent on the efficiency of the policies used to regulate the usage of software the fall-back path.

In this talk, I will first present experimental data highlighting the relevance of designing self-tuning mechanisms aimed to dynamically adapt the HTM fall-back policy. Then I will discuss recent and ongoing work aimed at pursuing this goal by exploiting lightweight on-line reinforcement learning algorithms.

Notes *(collected by members of the audience)*

- How many retries until going to fall back to lock/STM? How to cope with capacity aborts? Does capacity abort count as just one retry? Could: give up (drop all retries left), half (drop half of retries left), or stubborn (reduce retry count by one). How to implement fall back synchronization: single global lock, none (retry), aux (serialize on an auxiliary lock). How well does static tuning work? Compared a heuristic (as suggested by Intel) and gcc.
- There is room for improvement over these two policies. No single policy dominates. Results vary with benchmark and number of threads. Not all the optimization dimensions are relevant: it turns out that wait and aux are similar and none is rarely better (and not by much), so that dimension can be dropped. One size doesn't fit all.
- Adaptive self-tuning approach needed. How should parameters be learned, off-line or on-line? On-line seems reasonably feasible (affordable cost). Chose particular lightweight reinforcement learning methods: upper confidence bounds (for capacity aborts) and gradient descent (for number of retries in HW). At what granularity should we adapt? Per-thread and atomic block, or whole application?
  What metrics should we optimize for? Performance, power, or combination? Are they correlated? On average 0.81 correlation. But much stronger between optimal configuration for each target: 0.98. So go for time (since easier and cheaper to measure than energy).
- Two tuners (fine and coarse grain): one per thread per Atomic block, the other global. Integrated into gcc. Speedups relative to single-threaded, non-instrumented. Auto-tune works well (speedup around 4 for 8 threads for both SG and NoRec), the gains largely outweigh exploration cost. Gradient descent can get stuck in local maxima. Use random jumps to get unstuck. Sometimes it pays to rerun transaction on capacity abort.

4.24 How Vague Should a Program be?

Sven-Bodo Scholz (Heriot-Watt University Edinburgh, GB)

For many well researched problems there exist several alternative algorithms that compute solutions. Which alternative is best suited does not only depend on the overall goal but it typically also depends on many other factors, such as the actual data, the executing hardware or the way the algorithm is actually mapped onto that hardware. The choices that need to be made in this context are always a mix of decisions made by the programmer and decisions made by the tool chains that are being used.

In particular in the light of the ubiquitous availability of increasingly heterogeneous many-core systems implementation choices do not only become more complex, but the impact of the choices made are also becoming much more pronounced. With programmer productivity in mind it seems that shifting the decision making progress towards increasingly sophisticated tool chains is the only economically viable way to go. Although a lot of progress in that direction has been achieved over the last few decades, pushing this agenda further raises many rather fundamental questions such as (a) If our programs provide increasing freedom to the tool chain to adjust the programs for parallel execution, *what* is the notion of an algorithm? (b) Is it enough to specify one algorithm as a problem solution; or should we provide several a la peta-bricks? (c) What happens with determinism or provable properties

if we allow for more than one alternative? (d) Does a discussion about complexities still make sense? (e) Do we have mechanisms that allow tool chains to choose the "best" hardware to execute on?

Notes *(collected by members of the audience)*

- Problem: too many programming paradigms for parallelism; huge challenge for scientific practitioners. Desire: raise the level of abstraction. Particularly challenging in the multi-/many-core situation (consider programming GPGPUs)
- Single-Assignment C = like C without pointers and with N-dimensional arrays. Declarative/functional programming, backed with aggressive compiler optimizations. Map to lambda calculus. Tools to generate it from a variety of front-end languages, including things like MatLab.
- Pure functional intermediate form reveals many opportunities for high performance parallelism optimizations. In particular, allows major restructuring for different platforms without having to restructure source code. Allows more flexible choice of what gets translated onto special multi-cores (such as GPGPUs).
- Significant empirical evidence of viability of this approach. Can even beat hand-written CUDA code – e.g., on Anisotropic Diffusion image processing benchmark.
- Difficulty of comparing approaches. Could have multiple implementations – multiple algorithms – from which to choose. But then, what is the "algorithm"? How do we argue correctness when what's going on under the hood can vary so much? C compilers transform what is actually executed - arguably "same algorithm", but blocking, etc., are substantial transformations. But to do well on multi-/many-core versus single-thread, you need a "different" algorithm. Choice of algorithm depends on many things, most of which are not statically determined – so needs to be determined dynamically and perhaps not even deterministically. How then do we reason about correctness? Or complexity?

4.25 Persistent Memory Ordering

Michael Swift (University of Wisconsin – Madison, US)

Non-volatile memory (NVM) technologies, such as phase-change memory, memristors, spin-transfer torque MRAM, and others promise high-bandwidth, low-latency persistent storage through the standard memory interface. However, making memory persistent poses a number of challenges, including how to ensure data is durable in the presence of processor caches, and how to ensure consistency of updates.

A key challenge in persistent memory is that data residing in caches is not durable; it must be written back to NVM first. When to do this and how to order writes back to NVM are an open research question.

I discussed several models of persistent memory ordering and then raised some open questions.

Notes *(collected by members of the audience)*

- Persistent memory is a hot topic (memristors, etc.) Many research questions regarding how to enforce atomicity and consistency? This talk focuses on consistency. What is different to non-persistent memory? We need to have a commit record that allows us to recover after a crash. Why is this a problem? Write back cache: commit log is not written out in order. There is a volatile memory ordering that is defined in terms of views of CPUs. More interesting for consistency is the order at the DRAM. In the presence of persistence, state can be uncertain after a crash. Without ordering, we cannot enforce that the "commit" record be updated strictly after the other updates. **Q**[Milind Kulkarni]: Do you need to make sure that the cache is flushed in some manner? **A**: You write a record after flashing the cache to record what you have written.
- Simple (but expensive) solutions: disable caching for logs; write through cache, flush entire cache at commit. Other approaches: Mnemosyne, BPFS/epochs, and Intel's new instructions.
  [Stephan Diestelhorst]: There is a misconception that if it is out of the cache it is durable, but there are lots of buffers between cache and memory, see PCOMMIT in Intel's HW Support.
- Mnemosyne: Goal to not use any new instructions or special hardware. Primitives: ordered writes with non-cached stores (they can bypass cache, or force a line to be flushed) or using flush/fence instructions MOVNTQ/CLFLUSH. Transactions (with durability) based on tinySTM. Note: undo logging approach not great for this situation.
- BPFS/epoch barriers: An epoch is a group of writes that are delimited by a new form of barrier. Cache tracks epoch id. Epochs have to be written by the processor in epoch number order, older epochs need to be written before newer epochs. Also need to handle cross-CPU dependences. The Problem is that one does not know when the data has become durable. No way to force durability (except perhaps by force a line to be flushed). The idea is to get ordering by accessing persistent data written by a different processor. **Q**[Michael L. Scott]: Is this for current processors (EPOCHS)? **A**: No, it is for new hardware.
- HW Support by Intel (Spec from last August): CLFUSHOPT: an unordered flush; CLWB: writes back modified data but data stays in the cache (as unmodified), i.e., without flushing; PCOMMIT: commits data queued in the memory system to persistent memory, lets you know that persistent data are now durable in memory; need to use SFENCEs between these (since they're unordered).
- Generalizing persistent order: Memory Persistence: [Pelley, Chen, and Wenisch (Univ. of Mich.)]; Recovery observer model: defines order of visibility at non-volatile memory (not caches); Persist order: orders writes to non-volatile memory.
- Epochs to Strand persistence. Epochs require fitting everything into a linear order. Instead Strand associates writes with strand numbers. Strands are not ordered. Programmer or compiler can introduce explicit order strands.
- Open questions: Are there more efficient HW mechanisms (than epochs, say)? What HW mechanisms do we need for efficiently enforcing ordering? How can stores be ordered across cores (distributed transactions)? Do we need arbitrary dependence graphs? What granularity do we want for writes, e.g., a cache line? What is the appropriate programmer API? Library (key/value or object store)? Load/store? Transactions?

4.26 NumaGiC: a garbage collector for NUMA machines

Gael Thomas (Télécom & Management SudParis – Evry, FR)

Joint work of Gidra, Lokesh; Thomas, Gael; Sopena, Julien; Shapiro, Marc; Nguyen, Nhan

When running on contemporary cache-coherent Non-Uniform Memory Access (ccNUMA) architectures, applications with a large memory footprint suffer from a large garbage collector (GC) overhead. As the GC scans the reference graph, it makes many remote memory accesses, saturating the interconnect between memory nodes. This talk presents NumaGiC, a GC that addresses this problem with a mostly-distributed design. In order to maximize memory access locality during collection, a GC thread avoids accessing a different memory node, instead notifying a remote GC thread with a message; nonetheless, NumaGiC avoids the drawbacks of a pure distributed design, which tends to decrease parallelism. On Spark and Neo4j, two industry-strength analytics, with heap sizes ranging from 160 GB to 350 GB, and on SPECjbb2013 and SPECjbb2005, NumaGiC increases the performance of the collector by up to 5.4x over Parallel Scavenge, the default throughput-oriented collector of Hotspot, which translates into an overall performance improvement by up to 94%.

Notes *(collected by members of the audience)*

- Problem: large multi-/many-cores have lots of computing power but it is hard to build a GC that scales. Scaling is limited by data analytics and NUMA. Collector forces accesses to remote memories and parallel collection ends up saturating the interconnect because of the cache coherence protocol.
- Idea: Use messages. Observation: a thread mostly accesses objects it has allocated, i.e., threads mainly access local memory. Send message to the GC thread of the node to maximize locality. Requires cross-node references to be relatively rare. Heuristic: keep objects allocated by a given node on that node. But although this avoids remote memory accesses, being so strict degrades collector parallelism. **Q**[J. Eliot B. Moss]: There tends to be locality to the object reference. **A**: Sending a message is more costly than accessing one remote object. **Q**: Is this something that you tried and did not work well? **A**: Yes, for all the benchmarks, including DaCapo, etc. **Q**[Michael L. Scott]: You describe the problem as a bandwidth problem, but the solution may also affect the latency. **A**: Scalability problem comes from the saturation of the network. [Michael L. Scott]: The critical path length of the GC could be smaller with your approach even with infinite bandwidth. [Stephan Diestelhorst]: Could you prefetch to pull the object and have the same benefit in the end?
- Adaptive algorithm: Local mode: send messages when not idling; Thief mode: grab objects (pull to your node) when idling.
- Results: GC throughput 2–5x better (with heap size 3–4 x live size); application speedup of 12–66%; GC shows good scaling; memory access locality very important to GC performance. **Q**[Michael L. Scott]: AMD and Intel are both TSO. Are we reaching a point architecturally where we can see a difference in memory order between scalability of the machines. Will there be a difference in scalability between ARM and POWER chips? **A**: I do not know.

4.27 Utilizing task-based dataflow programming models for HPC fault-tolerance

Osman Ünsal (Barcelona Supercomputing Center, ES)

In this talk, I argued that task-based programming models are a good substrate to build fault-tolerant frameworks for High Performance Computing (HPC) systems.

In particular, I further advocated the use of dataflow runtimes for resilience. These runtimes facilitate fault isolation, minimize fault propagation, and help failure root-cause analysis.

I provided examples showing how leveraging task-based dataflow PMs could lead to efficient asynchronous checkpoint/restart and selective replication implementations.

Notes *(collected by members of the audience)*

- Application resilience, for instance, in the domain of climate change predictions, is an increasing concern. This is due to larger circuits, complex substrates, and complex software. MTBF on the order of tens of minutes without applying more techniques.
- Adopt a task-based dataflow programming model where task runs when all dependencies are satisfied. Envisioned for coarse grained tasks. The runtime system checkpoints the data at the start of a task as it knows the data inputs of each task. If failure occurs, the task is re-ran (as its effects are local); scales well with fault rate. Clarifies where checkpoints can be taken. Likewise, recovery tends to be fairly local. The approach uses Software CRC for error-correction, can exploit existing Intel instructions to reduce overhead. Protects only application tasks, not run-time system or OS. Task redundancy (pairs compare output; on difference, run third and take majority vote; do this only on tasks that are more likely to experience errors (based on their memory size)). Hard to handle global shared state. The approach could potentially benefit from non-volatile memory.
- **Q**: Does non.volatile memory help to achieve fault-tolerance with data flow programs? **A**: It can help to perform selective checkpointing more efficiently, but the issues do not seem different.

4.28 Commutativity Race Detection

Martin T. Vechev (ETH Zürich, CH)

Main reference D. Dimitrov, V. Raychev, M. Vechev, E. Koskinen, "Commutativity race detection," in Proc. of the 35th ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI'14), pp. 305–315, ACM, 2014.
URL http://dx.doi.org/10.1145/2666356.2594322

In this talk, I introduced the concept of a commutativity race. A commutativity race occurs when two method invocations happen concurrently yet they do not commute. Commutativity races are an elegant concept enabling reasoning about concurrent interaction at the library interface and generalize classic data races. I also discussed a way to dynamically detect

commutativity races based on a technique which combines vector clocks with commutativity information.

By generalizing classic read-write race detection, the work leads to many new interesting research questions at the intersection of program analysis and distributed computing. These questions are of both theoretical and practical importance. In particular, I discussed several open directions and in-progress results including: impossibility of simulating race detectors, discovering logical fragments for capturing commutativity, and black box learning.

Notes *(collected by members of the audience)*

- Commutativity race = 2 high-level (atomic) operations that do not commute and are not ordered. Useful for debugging and correctness checking; also for state space exploration.
- Knowledge of commutativity properties of operations is essential to practical model checking (it reduces the search space).
- To check for races efficiently, start with a logical specification of commutativity. Convert this commutativity specification into a structural representation. Combine with some specification of happens-before. All of this yields a race detector.
- Example: hashmap. Put, size, and get operations. Specification indicates commutativity using logical formulas on arguments. In a hashmap, insert(i) and insert(j) commute if i != j; for a register, write(i) and write(j) commute if i == j. The latter is harder.
- For more complex objects (e.g., array list from [Deokhwan Kim and Martin Rinard]), commutativity specs can be complicated, so they have devised a scheme to learn the commutativity spec from an abstract (or concrete) implementation.
- The commutativity race detector employs a "micro operations" representation and combines it with a happens-before scheme (vector clocks): it maps conflicts on high-level operations to conflicts on low-level objects, drawing on SIMPLE by [Milind Kulkarni]. It can be tricky to develop a succinct representation. Good representations are important because they enable optimizations of the sort employed by FastTrack for read-write races.
- Conflict checking is $O(n^2)$ in general, but if the conflict predicate is expressed in a particular form, the cost become linear (constant for each new operation). Open question: What is the richest logical fragment that results in linear cost? And which data structures have commutativity specs that lie in that fragment?
- There are various other challenges in both formal specification and and tool construction. For example: Can a read-write race detector precisely detect commutativity races? (Note that space matters.) Also: is there an interesting "consensus-like" hierarchy of concurrency analyzers?

4.29 Application-controlled frequency scaling

Jons-Tobias Wamhoff (TU Dresden, DE)

Joint work of Wamhoff, Jons-Tobias; Diestelhorst, Stephan; Fetzer, Christof; Marlier, Patrick; Felber, Pascal; Dice, Dave
Main reference J.-T. Wamhoff, S. Diestelhorst, C. Fetzer, P. Marlier, P. Felber, D. Dice, "The Turbo Diaries: Application-controlled frequency scaling explained," in Proc. of the 2014 USENIX Annual Technical Conf. (USENIX ATC'14), pp. 193–204, USENIX Association, 2014.
URL https://www.usenix.org/conference/atc14/technical-sessions/presentation/wamhoff

Most multi-/many-core architectures nowadays support dynamic voltage and frequency scaling (DVFS) to adapt their speed to the system's load and save energy. Some recent architectures additionally allow cores to operate at boosted speeds exceeding the nominal base frequency but within their thermal design power. In this talk, we propose a general-purpose library that allows selective control of DVFS from user space to accelerate multi-threaded applications and expose the potential of heterogeneous frequencies. We analyze the performance and energy trade-offs using different DVFS configuration strategies on several benchmarks and real-world workloads. With the focus on performance, we compare the latency of traditional strategies that halt or busy-wait on contended locks and show the power implications of boosting of the lock owner. We propose new strategies that assign heterogeneous and possibly boosted frequencies while all cores remain fully operational. This allows us to leverage performance gains at the application level while all threads continuously execute at different speeds. Our in-depth analysis and experimental evaluation of current hardware provides insightful guidelines for the design of future hardware power management and its operating system interface.

Notes *(collected by members of the audience)*

- Dynamic Voltage and Frequency Scaling (DVFS) leveraging existing X86 multi-cores. Novelty is applying DVFS on the application level.
- Idea of P (Performance) states (pre-defined frequency/voltage pair) and C-states: power states. C0 active; other C states have varying power usage and wake-up time. Trade-off: state transition latency vs. power consumption in that state. Access to states : HLT or MONITOR-MWAIT instructions.
- Investigated AMD Turbo core (modules = frequency domain; AMD: 2 x86 cores + 1 FPU) and Intel Turbo boost (package; both hyperthreads at some frequency). AMD Turbo Core and Intel Turbo boost modes are different. Intel also takes temperature into account. AMD is deterministic by load, can do asymmetric frequencies with manual boost (for one core). Intel reacts to thermal conditions, cores have to be at the same P level. Cores can run at different frequencies on AMD but not Intel. Boosting can be deterministic and thermal; must disable half the cores to give "head-room" to allow it.
- Tested on application Critical Section (CS) benchmark, uses "decorated" MCS lock. Turbo boosted on CS, when it is profitable to do so w.r.t. the overhead of changing V and f. Will (sometimes) trigger DVFS when waiting. Tested both automatic and manual freq scaling; identified the costs of transition. Energy implications of spinning vs. blocking (futex). Goal: run critical section on "fast" CPUs. How big does CS need to be for this to be interesting? Spinning (allows automatic scaling) vs. blocking (OS control). Break even time performance is 1.5M cycles for AMD, about 4M for Intel. Break even for energy is 7M cycle wait time on AMD; much less on Intel. Manual scaling. Overheads in 10s

of 1000s of cycles. Can: spin; owner boost (600k cycles); delegate (dedicated adjustable core); 200k cycles); or migrate (to already boosted core; 400k cycles).
- Developed a turbo library to change P states; simple interface. Ongoing work: boosting STM, async STM up to 50% speedup with 2% energy. Next steps: Haswell-EP supports per-core P-states.
- **Q**: Did you look at Intel SCC processor? It has DVFS domains including the interconnect. **A**: A student is looking into this.

5 Breakout Sessions

5.1 Group Discussion on Heterogeneity

What is a heterogeneous architecture? Numerous sources of heterogeneity are possible, from multiple micro-architectures for a single ISA to integrated CPU/GPU or CellBE-style architectures to fixed function accelerators to FPGA. In addition, heterogeneity exists for communication between near and far components. An old paper on the cyclical nature of display processor design was mentioned.

There was discussion of layers of software that are involved with heterogeneity. For example, in a single-ISA system like ARM big.LITTLE, the OS can migrate threads because all cores share an ISA. In a classic GPU system, the application or perhaps the runtime decides where to run code. The role of each layer (hardware, OS, runtime application) should be considered.

One problem that can arise in heterogeneous systems is poor memory behavior: if accelerators work on data in bulk, then often data must be spilled to DRAM, re-fetched to an accelerator and then re-written to DRAM before it is consumed by another accelerator. Better interfaces between accelerators, or methods to break problems into pieces that fit in the cache, could address this problem. Ideally, the programming model should preserve locality across modules.

Whether accelerators should share the same memory hierarchy as the CPU is also a question; for example, a GPU may trash the CPU cache because of its massive demand for memory bandwidth. There needs to be some control over how cache is shared.

A large concern was who looks out for holistic performance: heterogeneous systems involve many designers who look out for local performance but don't consider the whole system. For example, a GPU may consider that it owns the cache and hurt code on the CPU. When accelerators are integrated on-chip, it may be easier to make tradeoffs because there is tighter integration. In addition, with dark silicon the marginal cost of an accelerator is lower as compared to having to buy an external card.

A large question is when and how to decide where to run code, on which type of core or which accelerator? The ideal, we agreed, was that a programmer writes a single program that is then compiled for multiple architectures. If the placement decision can be made dynamically, then it allows undoing a bad decision. A challenge is that algorithmic changes may be needed to leverage different architectures, such as when moving from a CPU to a GPU, which means the programmer must be involved. The tradeoff here depends on the setting: for a mobile device programmer targeting a heterogeneous set of devices, a single code makes sense. Many programmers today do not want to target GPUs because architecture is moving too fast and they may need to rewrite code in the near future. For Google programmers targeting their own set of machines, then writing multiple versions of the code may be worthwhile.

Ideally, a runtime would dynamically decide what to run where. This is difficult in the face of different ISAs. Locality matters as much as specialization.

Another suggestion was to use libraries: experts can write different versions of a library for different accelerators, and applications can call into the appropriate one. This assumes, though, that the majority of execution time is spent in such libraries today to achieve a good speedup. Furthermore, the overhead of moving data in and out of a library through a procedural interface may be high.

Another concern was the complexity of heterogeneity and lack of predictability. It was noted that Amazon turns off dynamic performance features, such as hyperthreads and turboboost, to provide more predictable performance for customers. It was debated whether cloud vendors would want heterogeneous hardware or prefer homogeneous, as it makes machines more interchangeable and easier to manage. Heterogeneity is coming to the cloud from other places as well, with customized chips from Intel, a mix of DRAM and NVM, different transistor technology, etc.

A final big concern was the complexity of multiple accelerators: it may happen that there are incompatible interfaces, and some pushback may be needed for simpler, more orthogonal interfaces that can be composed easily, even at the loss of some performance.

Selected Contributions to the Discussion

[Torvald Riegel] worried about tool chains. How do we ship code for heterogeneous platforms? How do we integrate code from multiple sources? How do we debug?

[David F. Bacon]: the biggest performance benefits come from the most specialized accelerators. PowerEN suffered from being all wimpy cores.

[Charles E. Leiserson]: Moore's Law is going to end in about 5 years. Read Sutherland's paper [1] on the wheel of reincarnation

[Stephan Diestelhorst]: We tend to write all data to DRAM before starting the next SW module (from different vendor), which then pulls it back into cache. Sharing DRAM with accelerators is clearly good; sharing cache is not so clear.

[Charles E. Leiserson]: Everybody who cares about performance wants to solve the whole problem (in their world, at their level) in a way that writes everybody else out of the equation. Really worried about what this looks like in a world of heterogeneous chips.

[Sven-Bodo Scholz]: Hopeless to expect programmers to cope with heterogeneity.

Q[Michael L. Scott] (strawman): Can we just hide accelerator code behind library interfaces? **A**: [Charles E. Leiserson]: not if they're stateful. **A** [Sven-Bodo Scholz]: and not if state is huge and has to be piped through main memory.

[Milind Kulkarni]: GPUs are a counter-example: we keep data on the GPU across calls. (But this may require compiler help.)

Q: Will heterogeneity permeate the cloud? **A**: [Osman Ünsal]: yes [Charles E. Leiserson]: skeptical. An economic argument: will do it if it's cheaper. [Michael Swift]: Want predictability in billing.

References

1 T. H. Myer and I. E. Sutherland. On the design of display processors. *Commun. ACM*, 11(6):410–414, June 1968. ISSN 0001-0782. DOI: 10.1145/363347.363368.

5.2 Group Discussion on the Future of TM

- How do we make HTM perform well? Issue of cost of start/end transaction on Haswell hardware. Issue of PPC where cost of locking is high, so HTM gains (artificial?) benefit. Unreasonable to expect it to speed up compared with carefully crafted non-blocking algorithms (for example).
- So the space of interest is algorithms for which we do not yet have hand-crafted versions – and goal would be comparable performance.
- One opportunity might be new programming languages – can obtain both simpler code and good performance; on a related point, you can build more sophisticated atomic data structures – such as double-ended queues for work-stealing.
- Need HTM designed to play well with Hybrid schemes.
- Even lock elision is not exactly a "no code changes needed" proposition – consider adding a counter to a critical section: to avoid high rate of abort, it needs to be at the end of the critical section. Observed that a JIT can do this a lot of the time.
- Need debugging and analysis tools that tell developers why aborts occur.
- What about breaking TM down into building blocks with hardware assist? Various program analyses do things quite similar to HTM. Obvious: detection of conflicting accesses. Buffering speculative writes. Ability to pull items back out of a set.
- Non-transactional reads/writes to leak information intentionally.
- Will HTM just fade away? IBM folks think not, but in the Intel space it seems more iffy – consensus is that it needs to be more broadly offered to get more customer usage, but that it appears Haswell TSX will indeed be more broadly available.

5.3 Two Group Discussions on Persistent Memory

Notes from Group 1

[Hans-J. Boehm]: Hardware issues: We must control the order in which things become visible to non-volatile memory. Typically we force data from cache to the memory controller, but not to the non-volatile memory. It seems that the hardware must give you mechanisms to flush the memory controller buffers to memory. Cache line flush instructions (coming on new Intel machines). There are non-temporal store instructions that can be used to write things all the way down to non-volatile memory. There was work at HP that was leveraging "non-temporal stores", which are far from perfect. You may keep multiple versions of the same data in non-volatile memory at the same time. One option is to keep a write-through, non-volatile shadow of volatile DRAM.

[Torvald Riegel]: What impact will there be be on the programming model? 1) Provide a file-system interface – wastes the potential of NVRAM. 2) Libraries and/or 3) Abstraction with loads/stores? We cannot let people to essentially program with persistent loads and stores – cannot use in templates then.

[David F. Bacon]: What are the use cases? Optimize data-base transactional manager? What is the actual application where I will pay the performance overhead for NVRAM and gain?

General agreement that MemCacheD is a nice example (a key-value pair storage). [Torvald Riegel]: Can we write a *portable* nonvolatile memcached?

[Hans-J. Boehm]: There is lots of code in Android to serialize and deserialize data structures. Does NVM mean we can, effectively, just mmap the file that holds the pointer-rich data?

It would be nice to have only one copy of these data structures instead of having to serialize. There will be still need to synchronize with the NVRAM. Can you treat NVRAM simply as a faster way to do serialization? Snapshoting in general as a use case? To make the consistent state persistent. Write a portable C++ for MemCacheD, but need something that the compiler can handle.

System persistence proposed by someone at Microsoft. The idea is to provide enough capacitor/battery power to flush the cache when things are about to die. But may not be able to build a consistent state. The cache-line flushing goes away, the other issues do not go away.

Are transactions the right model? Does it mean that there must be a transactional programming model in place, or is snapshoting sufficient? Can use a lock-based system to obtain the desirable characteristics of a transactional system. We want transactions to protect the integrity of the file system. [Michael L. Scott]: Transactions are atomic methods of concurrent (or stable) objects. The txn system builds bigger abstractions from smaller ones. One seldom (never?) wants pointers from NV to V state.

[David F. Bacon]: But what notation do we give to the programmers to build the NV abstractions?

Many (most? all?) of us have the intuition that persistent pointer-rich structures are "more dangerous" than file-based data. Why, exactly?

Still need partitioning between consistent data and modified data, but we should be able to avoid serialization, so that we do not need to convert into blocks.

How to handle pointers from non-volatile memory to volatile memory? Use a type system?

[Osman Ünsal]: Note that persistent != stable. It doesn't eliminate the need for replication.

[David F. Bacon]: Maybe NVM will be the natural successor for DRAM, for density and cost (and speed?) Maybe persistence will just be a sidelight.

[Michael L. Scott]: Would we then start saying "what can we do with the feature we've been ignoring?" Maybe post-crash forensics of some sort?

Partition memory space so that there is simply a separate partition that is non-volatile.

The window for rollback is much smaller than in the HPC world. And this smaller window is transformative.

HPC-style checkpoint-restart would not work in the case of a buggy program because you do not want to restart in that buggy state.

Is there an opportunity to simply use NVRAM the same way that we use it as a RAM?

Should all storage to persistent state be forced to go through some barrier, such as a system call?

[Hans-J. Boehm]: Lots of people are pursuing APIs similar to mmap (with different implementation under the hood). For example, Facebook has an Mmapped file that persists through the rebooting of a process.

[Michael L. Scott]: What about durable STM? Is there a straightforward path toward adding 'D' to 'ACI'? Are there implications for STM if we add persistence?

[Hans-J. Boehm]: One question is how to do ordering wrt non-transactional accesses – publication and privatization, essentially. If we persist a pointer (transactionally, say), we want to make sure the stuff it points to is already persisted.

Any non-transactional write that is observed by a transaction must be persistent before the transaction is made persistent.

[David F. Bacon]: Maybe we should require *all* persistent writes to be in transactions.

[Torvald Riegel]: But that skips the important optimization of eliding metadata maintenance for update and persistence of the (then) private data.

Note that persisting all previous NV writes of my thread before persisting a txn isn't enough: I have to persist everything that happened before.

[Torvald Riegel]: It's not clear a txnal model is the right level at which to support portability. We may want something lower level.

Should persistence becomes part of the type system?

Focusing on transactional interface, it is hard to pin down semantics for transactions. Would it be better to specify something below that, so that people can then write their transactional abstraction on top of it?

Notes from Group 2

We discussed that while persistence in memory is not new, due to density and power considerations, technologies such as phase change memory/STT-MRAM (reading faster than DRAM/writing slower, 4K reads in 2 μsecs)/memristors will likely replace DRAM. Already, Flash-backed DRAM DIMMs with supercapacitors are available, and in Linux, are treated like a NUMA zone. While typical SSDs are block-based, with kernel-level file system style access, these DIMM replacements will allow fine-grain reads and writes at low latency. The ability to manage durability (persistence) in software is hampered by the high speed.

We discussed what support might be needed, in particular, the ability to "push" data to persistent memory to force durability, and the need for atomicity and ordering. What is needed is:

- Control over when data reaches persistent memory.
- Control over ordering of modifications.
- Evaluating the need for (and ability to avoid) redundant pointers and checksums.

Existing support to flush individual cache lines (e.g., Intel's clflush instruction) could be useful. However, just as clflush interferes with transactions, careful support will be needed to decide on when data is flushed to persistent memory and in what order. To achieve atomicity, maybe create a nonvolatile cache of logs with hardware that later updates the data. We need a combination of volatile and nonvolatile memory in order to control when persistence is attained. How can we simplify the process of synchronously updating data structures and avoid the need for code maintaining these data structures?

We discussed some existing efforts in this direction. [Michael Swift] has provided a bibliography as part of his seminar materials, which is also replicated below). HP Labs did some work on consistent durable data structures. Microsoft has a proposal for whole system persistence via epochs in the cache flush process across all memory controllers in the system and with the ability to explicitly flush the caches when the power goes out. The epoch approach creates a new copy of data that is updated rather than updating in place. Michael Swift's Mnemosyne project moves this to software and writes data at the granularity of an update.

We discussed the need for a "killer" application. Michael Swift summarized that when using NVM for filesystems, in his experiments evaluating a file system benchmark meant as a stress test, only 1% of total accesses were to non-volatile memory (reading/writing data or metadata). One possible application where fine-grain persistence might be useful is high frequency trading, where transactions are small, and all actions are logged for replay by the SEC for possible replay up to 72 hours later. Another is the increasing use of in-memory

databases, where there would be direct reads in processing a query, but still the need to write out logs for persistence. This would reduce commit times from 100 microseconds for flash to 1 microsecond. Could also simplify software.

Other issues to consider and benefits of persistent memory:

- Need for proactive versus reactive support for failure.
- Linux – execute in place – execute operating system from flash.
- Intel's PMFS – directly accesses file data from persistent memory.
- mmap has copy-on-write, persistent memory would need something efficient.
- What about a persistent CAS – in-memory CAS? Or in-cache+extra step for persistence?
- File systems have an important property of consistency and naming; how to retain this?
- Application programming interface to durability needs to change.
- Concurrency control over NVM – managing locks using epoch numbers; epoch mechanism used to create snapshots.
- Persistence as a property of the type system?
- Narrow interface that file systems provide help prevent stray pointer corruption issues; this may be a challenge for fine-grain access to persistent memory.
- Use of a "building block" approach for HTM that can then cover persistence as needed might be beneficial.

References

1. Haris Volos, Andres Jaan Tack, and Michael M. Swift. Mnemosyne: Lightweight persistent memory. *SIGPLAN Not.*, 47(4):91–104, March 2011. ISSN 0362-1340. DOI: 10.1145/2248487.1950379.
2. Jeremy Condit, Edmund B. Nightingale, Christopher Frost, Engin Ipek, Benjamin Lee, Doug Burger, and Derrick Coetzee. Better I/O through byte-addressable, persistent memory. In *Proceedings of the ACM SIGOPS 22Nd Symposium on Operating Systems Principles*, SOSP'09, pages 133–146, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-752-3. DOI: 10.1145/1629575.1629589.
3. Steven Pelley, Peter M. Chen, and Thomas F. Wenisch. Memory persistency. In *Proceeding of the 41st Annual International Symposium on Computer Architecuture*, ISCA'14, pages 265–276, Piscataway, NJ, USA, 2014. IEEE Press. ISBN 978-1-4799-4394-4. DOI: 10.1145/2678373.2665712.
4. Youyou Lu, Jiwu Shu, Long Sun, and O. Mutlu. Loose-ordering consistency for persistent memory. In *Computer Design (ICCD), 2014 32nd IEEE International Conference on*, pages 216–223, Oct 2014. DOI: 10.1109/ICCD.2014.6974684.
5. Jishen Zhao, Sheng Li, Doe Hyun Yoon, Yuan Xie, and Norman P. Jouppi. Kiln: Closing the performance gap between systems with and without persistence support. In *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-46, pages 421–432, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2638-4. DOI: 10.1145/2540708.2540744.
6. Ellis Giles, Kshitij Doshi, and Peter Varman. Bridging the programming gap between persistent and volatile memory using WrAP. In *Proceedings of the ACM International Conference on Computing Frontiers*, CF'13, pages 30:1–30:10, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2053-5. DOI: 10.1145/2482767.2482806.
7. Shivaram Venkataraman, Niraj Tolia, Parthasarathy Ranganathan, and Roy H. Campbell. Consistent and durable data structures for non-volatile byte-addressable memory. In *Proceedings of the 9th USENIX Conference on File and Stroage Technologies*, FAST'11, pages 5–5, Berkeley, CA, USA, 2011. USENIX Association. ISBN 978-1-931971-82-9. URL: http://dl.acm.org/citation.cfm?id=1960475.1960480.

6 Panel and Plenary Discussions

6.1 Panel on How to Teach Multi-/Many-core Programming

Panelists: Maurice Herlihy, Charles E. Leiserson, Michael L. Scott, and Nir Shavit.

- Teach concurrency from the beginning.
- Concurrency versus parallelism: avoiding non-determinacy and interaction.
- Carefully distinguish easy cases from the hard ones.
- Can introduce message passage, shared memory, and things like memory models later, in appropriate contexts.
- Need a suitable language for teaching deterministic concurrency in (say) data structures course.
- Java useful because of GC and the concurrency package.
- Abstract algorithms such as Baker's algorithm help give intuition.
- Computer organization has been taught bottom-up and top-down.
- Introduce simple abstractions first.
- We do not have good textbook(s).
- Performance engineering course at MIT teaches parallelism/concurrency in context of larger set of techniques including caching, pipelining, etc.
- Concurrency (interaction) done for reasons not necessarily related to performance; parallelism is for performance.
- Some of this is not about programming – applies to (say) constructing a building.

6.2 Plenary Discussion on VM Design for Concurrency

What primitives should a VM provide? What about threads in a cloud, each from different languages? What sharing of code/metadata would happen between instances of the VM? What do you provide to the language implementer? Channels for communication – can be shared across languages? Great to have hardware help with bookkeeping. Would also be great to be able to capture the logs. May need to capture reads as well. Greater risk if you go too high level as opposed to lower level. How would HTM work in a virtualized environment? Something like Haswell should directly abort. Hazards of a high-level implementation in HW – getting things like page faults, etc., can be problematic. Would it slow things down to use smaller primitives? No intuition that the cost would be higher or lower. Interaction with non-transactional accesses needs to be considered carefully. Park/Unpark has useful properties, and so do futexes – general conditions hard with only futex since it looks at a single location; unpark means you have control over which thread to wake up (can build that with futex as well) Generally want to use these in a style where you re-check the condition (see synchronic<T>). Identities when grabbing a lock or accessing a resource: What is the identity of the agent? Is it a process, thread, whatever? In his talk, [Michael L. Scott] suggests a single hierarchical mechanism – perhaps similar to concurrent nested transactions, or to re-acquisition of the same lock in a subroutine in Java – that is, re-entrant locks depend on a notion of identity. What about accessing "thread-local" storage . . . at different levels of identity – does this get us back into cactus stacks, though maybe needed only for identity and explicitly managed identity-specific storage? Should that be part of the model of the VM? Maybe want something analogous to futexes on identities? Distinction between logical and physical execution agents, e.g., Java thread versus OS thread versus CPU hyperthread –

to what extent would a VM model need to expose this distinction? Could the hierarchy have more to do with what features you have? Along the lines of the concurrency, parallel, and weakly parallel execution agents in the C++ model described in the talk by [Torvald Riegel].

7 Some Results and Open Problems

7.1 Deterministic algorithm for guaranteed forward progress of transactions

Charles E. Leiserson (MIT – Cambridge, US)

The following latest revision of a contention-management algorithm is one of the results of numerous discussions at this Dagstuhl Seminar:

```
SAFE-ACCESS(x)
 1  if lock(x) ∈ L
 2      // do nothing
 3  else
 4      M = {l ∈ L : l > lock(x)}
 5      L = L ∪ {lock(x)}
 6      if M == ∅
 7          ACQUIRE(lock(x)) // blocking
 8      elseif TRY-ACQUIRE(lock(x)) // nonblocking
 9          // do nothing
10      else
11          roll back transaction
12          for l ∈ M
13              RELEASE(l)
14          ACQUIRE(lock(x)) // blocking
15          for all l ∈ M in increasing slot order
16              ACQUIRE(l) // blocking
17          restart transaction // does not return
18  access location x
```

Accessing a memory location x within a transaction with lock set L. The *lock* function maps the space of all locations to a finite ownership array, each slot of which contains an anti-starvation (e.g., queuing) lock. The slots are ordered by an arbitrary linear order, most conveniently, the index in the ownership array. At transaction start, the lock set L is initialized to the empty set: $L = \emptyset$. When the transaction commits, all locks in L are released.

Notes *(collected by members of the audience)*

Deadlock-free because locks on which you wait are acquired in order. On repeated abort, the lock set keeps growing, and that helps with eventual progress. Allows a compilation strategy that figures out locations, acquires in order, and has guaranteed progress. Interesting policy

questions – can do bounded waiting rather than immediately aborting, can wait before restarting, etc. How does this interact with dynamically allocated blocks of memory? Their lock numbers may be different on each try. Probably have a mapping from addresses (say) to a smaller set of lock numbers, which may be some kind of hash – but could (in the SW case) be done in terms of something like Java hash codes, which work even when an object is relocated. The pessimistic style is both its strength and its weakness – but may be able to start optimistic and go pessimistic.

7.2 Thoughts on a Proposal for a Future Dagstuhl Seminar

- Maybe less TM, and more language/tool chain.
- Maybe more people (could have been timing that kept this workshop smaller).
- Heterogeneity should still be part.
- More of the systems-oriented formal methods people.
- Benchmarking and performance evaluation methodology.
- More about abstractions and programming models, and how they might appear in languages.
- In scientific computing domain a new GPGPU + POWER9 machine will be coming available, and that could be relevant.
- The situation around non-volatile storage may be different and that could affect the mix of topics.
- More broadly, the HW picture is evolving, e.g., end of Moore's law.

Participants

- José Nelson Amaral
University of Alberta, CA
- Hagit Attiya
Technion – Haifa, IL
- David F. Bacon
Google – New York, US
- Annette Bieniusa
TU Kaiserslautern, DE
- Hans-J. Boehm
Google – Palo Alto, US
- Daniele Bonetta
Oracle Labs – Linz, AT
- Sebastian Burckhardt
Microsoft Corp. – Redmond, US
- Irina Calciu
Brown University, US
- Dave Dice
Oracle Corp. – Burlington, US
- Stephan Diestelhorst
ARM Ltd. – Cambridge, GB
- Sandhya Dwarkadas
University of Rochester, US
- Pascal Felber
Université de Neuchâtel, CH
- Christof Fetzer
TU Dresden, DE
- Maurice Herlihy
Brown University, US
- Antony Hosking
Purdue University, US
- Milind Kulkarni
Purdue University, US
- Viktor Leis
TU München, DE
- Charles E. Leiserson
MIT – Cambridge, US
- Yossi Lev
Oracle Corp. – Redwood Shores, US
- Alexander Matveev
MIT – Cambridge, US
- Maged M. Michael
IBM TJ Watson Res. Center – Yorktown Heights, US
- J. Eliot B. Moss
University of Massachusetts – Amherst, US
- Erez Petrank
Technion – Haifa, IL
- Michael Philippsen
Univ. Erlangen-Nürnberg, DE
- Torvald Riegel
Red Hat GmbH – Grassbrun, DE
- Paolo Romano
INESC-ID – Lisboa, PT
- Sven-Bodo Scholz
Heriot-Watt University Edinburgh, GB
- Michael L. Scott
University of Rochester, US
- Nir Shavit
MIT – Cambridge, US
- Michael Swift
University of Wisconsin – Madison, US
- Gael Thomas
Télécom & Management SudParis – Evry, FR
- Osman Ünsal
Barcelona Supercomputing Center, ES
- Martin T. Vechev
ETH Zürich, CH
- Jons-Tobias Wamhoff
TU Dresden, DE

Report from Dagstuhl Seminar 15022

Quality of Experience: From Assessment to Application

Edited by

Katrien De Moor[1], Markus Fiedler[2], Peter Reichl[3], and Martín Varela[4]

1 **NTNU – Trondheim, NO,** katrien.demoor@item.ntnu.no
2 **Blekinge Institute of Technology – Karlskrona, SE,** markus.fiedler@bth.se
3 **Universität Wien – Wien, AT,** peter.reichl@univie.ac.at
4 **VTT Technical Research Centre of Finland – Oulu, FI,** martin.varela@vtt.fi

Abstract

This report provides an overview of the program, discussions and outcomes of Dagstuhl Seminar 15022 "Quality of Experience: From Assessment to Application", which took place from 4–7 January 2015 at Schloss Dagstuhl – Leibniz Center for Informatics. The seminar and the challenges that were addressed have their roots in the earlier Dagstuhl Seminars 09192 "From Quality of Service to Quality of Experience" and 12181 "Quality of Experience: From User Perception to Instrumental Metrics". The main goal of the seminar was to strengthen and go beyond the current understanding on Quality of Experience (QoE) and its assessment, in order to start tackling the logical yet highly challenging next steps: to move from assessment to application and to translate insights on QoE and knowledge from this research field into forms of economic and/or societal value. This report contains the personal statements and main challenges brought forward by the participants, who were on the fly clustered into six main discussion topics. We here report on the discussions and outcomes from the group work, organized around these bottom-up generated topics: "QoE theory and modeling", "QoE methodologies", "User factors and QoE", "QoE management", "Monetization of QoE" and "QoE in new domains".

Seminar January 4–7, 2015 – http://www.dagstuhl.de/15022
1998 ACM Subject Classification H.5.1 Multimedia Information Systems, H.5.2 User Interfaces
Keywords and phrases Quality of Experience, Usability, User experience, Content, Network monitoring, Quality measurement, Service pricing, Network management, Application management
Digital Object Identifier 10.4230/DagRep.5.1.57

1 Executive Summary

Katrien De Moor
Markus Fiedler
Peter Reichl
Martín Varela

Within the past few years, Quality of Experience (QoE) has gone through an explosive growth and established itself as an independent, multidisciplinary field of research, both in academic and industrial communities. Significant advances have been made with respect to the conceptual understanding of QoE as well as in terms of methodology and instrumentation, and the earlier Dagstuhl seminars 09192 "From Quality of Service to Quality of Experience"

Quality of Experience: From Assessment to Application, *Dagstuhl Reports*, Vol. 5, Issue 1, pp. 57–95
Editors: Katrien De Moor, Markus Fiedler, Peter Reichl, and Martín Varela
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and 12181 "QoE: From User Perception to Instrumental Metrics" have played a catalyzing role in this process: for example, by putting key challenges on the agenda, by stimulating (collaborative) activities that address them and by contributing to the establishment of a multi-disciplinary community around the topic, involving a range of actors with sometimes very different, yet complementary perspectives, priorities and motivations in relation to QoE. The main goal of this seminar was to strengthen and go beyond the current understanding on Quality of Experience (QoE) and its assessment, in order to address a logical yet highly challenging next step, namely to move from assessment to application and to translate insights in QoE and knowledge from this research field into forms of economic and/or societal "value". The main underlying motivation is that – even though the conceptual grounds and methodological implications of QoE are a very interesting and worthy research topic as such – they also represent milestones on the road to reach another ultimate goal: translating the theoretical and empirical understanding of QoE, its assessment and measures, into "value". This value can be rather explicit and concrete (e.g., increased revenue, or reduction of number of customer complaints), but it can also be intangible and more latent (e.g., customer loyalty, strengthened relation between a customer and a provider, enabling user empowerment, contributing to well-being).

The seminar brought together 27 participants to work towards this challenging goal. They were representing 13 different countries and 17 different institutions, resulting in a variety of different backgrounds and specific expertise domains. The seminar took place over 2.5 days and was organized in such a way that time for group discussion and interaction was maximized, while the time for individual presentations was kept to a minimum. At the beginning of the seminar, every participant was invited to write down three challenges fitting within this overall scope of the seminar. Thereupon, a concise presentation round was organized. Every participant was asked to make a short statement (5 minutes/1 slide) related to her or his challenges. These personal statements are included in the form of short abstracts in this report.

The main challenges and questions put forward by the participants were clustered on the fly into six broader topics, around which the seminar group work was organized, namely: "Theory and modeling", "QoE methodologies", "User factors and QoE", "QoE management", "Monetization of QoE" and "QoE in new domains". The group work was divided into two parts, with three topics being discussed in parallel in both parts of the group work. The initial assignment of participants to the six groups was deliberately organized randomly instead of thematically. The intention was to mix up participants with different backgrounds and interests as much as possible in order to stimulate open discussions and flow of thoughts. Participants had the possibility to switch to another group by exchanging with another participant in case they had a strong preference for another group. Every participant was involved in two discussion groups.

In between part 1 and 2 of the group work, a plenary reporting session was organized. During this plenary session, each group briefly presented the main points discussed and potential joint activities. During the final plenary reporting and closing session, the main points and outcomes from the second part of the group work were presented. Extensive summaries of the discussions and main outcomes for each of the six working groups are presented in Section 4 of this report. Due to the time constraints, there was unfortunately not enough time for deep follow-up discussions in the plenary sessions. The seminar as such was also very briefly evaluated in the final plenary gathering. One important factor which would have further improved the participants' QoE and which was mentioned several times, is more time for "digestion" and "reflection" between the sessions (which was indeed limited,

given the duration of the seminar). Overall however, and supported by the participants' feedback during and after the seminar, we can look back on a successful and productive seminar during which plans for several future and follow-up activities were made.

2 Table of Contents

3 Overview of Talks

3.1 Individual experience? Adaptive QoE for monetization

Jan-Niklas Antons (TU Berlin – Berlin, DE)

QoE has been a rapidly growing topic in the last few years. One big future topic of research is how to show the value (in terms of monetization) to for instance industry or customers. Customer behaviour – in terms of media usage – is important as one key performance indicator when it comes to monetization. The known models of QoE for quality estimation and quality driven technology development are mainly focused on larger customer groups, but these average values must not always represent the majority of customer perception or behaviour. Therefore adaptive QoE estimation strategies could be a solution. As additional estimates and moderator variables, user groups could be identified and the influence of the individual user's state could be utilized.

3.2 QoE from active network management

Alemnew Asrese (Aalto University – Espoo, FI)

The existing frameworks Quality of Experience models for Internet applications such as web applications are mainly based on subjective assessment. This subjective assessment needs much resources and time. On the other hand, today's Internet applications are becoming more complex, there are many in number and type, and they have different requirements for QoE. One of the challenges is how to model and predict the perceived QoE from the low level (active) network measurement. Thus we need to have a framework that could correlate the subjective QoE with the one which is predicted from active network measurement results.

3.3 QoE and the re-assessment of the assessment

Katrien De Moor (NTNU – Trondheim, NO)

The new definition of QoE [1, 2] represents an important step forward for the field: QoE is equaled to a dynamic affective state (a *degree* of *delight* or *annoyance*), and its transient and relative character is fully underlined. Explicit reference is made to the extent to which a person's expectations and needs are fulfilled (with respect to the utility and/or enjoyment) through the experiencing of an application, service or system, relative to the user's context, personality and current state (e.g., mood). It is furthermore acknowledged that a range of complex and strongly interrelated factors may influence QoE and a classification of potential influence factors has been presented. However, this more holistic and humanistic theoretical

perspective on QoE has implications for the way in which QoE is and should be evaluated, measured and predicted, and can be linked to new challenges:

1. **Put the new definition of QoE into practice.** The traditional, standardized measures used in quality assessment and subjective testing are insufficient and need to be extended with robust and validated alternative measures of QoE (beyond Mean Opinion Scores, MOS) that allow to grasp QoE in terms of human affect (including delight and annoyance) and that enable to capture also hedonic (and not only utilitarian) features of QoE. Only in a limited fraction of the literature, this challenge is addressed and as a result, there is a barrier to advancing the state of the art regarding fundamental relations that are highly of interest in research on QoE, for instance between (indicators of) technical performance, perceived technical quality, delight and annoyance, enjoyment, user engagement. The lack of such shared standard measures furthermore limits the comparability of studies.
2. **Increase the ecological validity of QoE research.** Most QoE experiments take place in a controlled and artificial laboratory setting, meaning that the human subject and the experience as are completely taken out of their natural, real world environment and that ground truth data are collected in research settings characterized by a very low ecological validity. There is therefore a need to study and understand QoE also in real world contexts and over longer periods of time in order to better current understanding the relevance, impact and weight of individual or combined influence factors.
3. **Investigation of QoE in less traditional application domains targeting higher level goals (e.g., continuous care, learning, . . .) instead of monetary gains.** One important question is what should/could be the outcome of striving for high QoE. From a business perspective, QoE is linked to a potential for increased revenues, enhanced customer loyalty and satisfaction, and considered as a strategy for reducing the risk of churn. However, striving for QoE could also be relevant in non-profit contexts and sectors with goals that go beyond purely economic drivers, such as enabling positive, valuable experiences as a goal in itself, provision of continuous care (e.g., in a hospital setting), learning, . . . Investigating QoE in such less traditional application domains requires an adaptation of the tools and measures that are currently used.

References

1 Patrick Le Callet, Sebastian Möller, and Andrew Perkis. Qualinet white paper on definitions of quality of experience, version 1.1, june 3, 2012. *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, 2012.
2 Alexander Raake and Sebastian Egger. Quality and quality of experience. In Sebastian Möller and Alexander Raake, editors, *Quality of Experience*, T-Labs Series in Telecommunication Services, pages 11–33. Springer International Publishing, 2014.

3.4 Challenges

Philip Eardley (British Telecom R&D – Ipswich, GB)

Within BT we measure the performance of our broadband network by using active measurements from probes at the home gateway in the homes of a few thousand volunteers. This is proving very useful to our BT Retail operational division, and I see several challenges to extending its capabilities. Firstly, how to help the human expert who studies the measurement

results, for example by automatically identifying likely issues in the network that should be investigated more closely. Secondly, how to measure aspects that the current tests struggle with, such as the performance of home networks and of end-to-end services like gaming; I suspect this needs passive or hybrid tests. Thirdly, how to ensure the security and privacy of such a monitoring capability, especially as it becomes more widespread and pervasive.

3.5 The human, the technology and the business: bridging QoE, user experience, technology experience and customer experience

Sebastian Egger (AIT – Wien, AT)

Since the advent of QoE (and the first Dagstuhl seminars on that topic) several things have changed. QoE is nowadays a well accepted research strand in telecommunications and is properly addressed by a number of research institutions. Despite this valuable achievements, a large share of nowadays technological and telecommunication systems is still not fully considering user requirements with respect to quality as well as utility. Therefore, my main aim for a future research agenda on the topic of QoE is to subdue technology from a human perspective. This aims to continually assess existing as well as future applications for their subordination with respect to aforementioned utility and quality requirements. As existing QoE models and assessment methodologies only tackle a rather limited (and telecommunication centric) subset of applications, the first challenge is to extend the types of applications that are currently investigated in the QoE research context to get a broader range of human technology interaction considered. The second major challenge will be to bridge the current gap between QoE, user experience (UX), technology experience (TX) and customer experience (CX). This will include identification of appropriate (experience) time spans for each of these four experience categories as well as the understanding of interrelations and differences between these concepts. Addressing these two challenges will lead to a more holistic picture of experience of human subjects with technological systems as well as the contribution of different dimensions (QoE, CX, TX) to the overall customer experience of a service as a whole.

3.6 Teletraffic-inspired QoE models: the basis for more effective SLA

Markus Fiedler (Blekinge Institute of Technology – Karlskrona, SE)

Main reference M. Fiedler, J. Shaikh, V. J. D. Elepe, "Exponential On-Off Traffic Models for Quality of Experience and Quality of Service Assessment," PIK-Praxis der Informationsverarbeitung und Kommunikation, 37(4):297–304, 2014.
URL http://dx.doi.org/10.1515/pik-2014-0031

From a teletraffic perspective, Quality of Experience (QoE) modeling has many yet unexploited facets. Some traditional QoE modeling efforts and models suffer from loose couplings between cause (e.g. bit error rates in a wireless channel) and consequence (e.g. delayed delivery due to intermediate repair actions instead of pure image quality reduction). Teletraffic models that capture and keep track of impacts on QoE all through the communication system

and stacks help to create stronger links between observable Quality of Service (QoS) issues and QoE estimations. In particular, the model parameters tell stories about sensitivities and thresholds regarding observable QoS parameters.

Such teletraffic-inspired QoE models address a first challenge on how to pave the way from QoE assessment to application, namely to devise effective QoE-based Service Level Agreements (SLA). Further challenges are to take QoE concepts to new domains, such as the Internet of Things (IoT), Machine-to-Machine (M2M) and Business-to-Business (B2B) communications, and to express (good or lousy) QoE in monetary values.

3.7 Joint QoE assessment for software validation

Farnaz Fotrousi (Blekinge Institute of Technology – Karlskrona, SE)

QoE in the context of telecommunication networks is affected by the quality of the software applications as a proxy for the user experience of those networks.That makes a joint collaboration between telecommunication and software engineering areas. Validations of software features that integrate network features could be linked to the user acceptance or churn. Correlation between quality (software and network) and its impact on the users (QoE) contributes for conceiving, evolving and maintaining software systems. For this purpose, one of the challenges is about mapping the definition of quality in software engineering and telecommunication domains where a different categorization for quality definition has been established. The second challenge would be capturing a joint measurement where some quality attributes are dominated in software engineering and/or telecommunication. Defining the quality threshold for an acceptable software would be another challenge depends on the software functionality and relevant network elements.

3.8 Challenges for quality-driven content delivery

Pantelis Frangoudis (INRIA Rennes – Bretagne Atlantique, FR)

The availability of QoE models for various application domains makes it possible to utilize them for quality-driven service delivery. These models may require information from multiple layers, which should be provided by different stakeholders (user, content provider, ISP) in a timely manner, to drive application behaviour adaptation. We focus on an over-the-top (OTT) content delivery environment, where data are disseminated over the infrastructure of ISPs, which is outside the content provider's (CP) control. In this environment, it is critical to address issues of ISP-CP cooperation, since network awareness is important for QoE-driven delivery, while at the same time decisions by the OTT provider can impact the operation of the ISP. Significant challenges thus emerge. From an operational perspective, the level of ISP-CP cooperation defines the granularity of network-level information (e.g., congestion in network segments in the user-data center path) that the ISP is willing to share with the CP, but also affects the process of placing and controlling probes in the ISP network, so that accurate QoE estimation can be achieved. At the business level, an open issue is whether it is feasible and if there are incentives for ISPs and OTT providers to offer QoE-driven SLAs.

3.9 Towards smart determination of appropriate quality

Samuel Fricker (Blekinge Institute of Technology – Karlskrona, SE)

Joint work of Fricker, Samuel; Farnaz, Fotrousi; Fiedler, Markus
Main reference F. Fotrousi, S. Fricker, M. Fiedler, "Quality Requirements Elicitation based on Inquiry of Quality-Impact Relationships," in Proc. of the 22nd IEEE Int'l Requirements Engineering Conference (RE'14), pp. 303–312, IEEE, 2014.
URL http://dx.doi.org/10.1109/RE.2014.6912272

The telecommunication and software engineering disciplines approach each other increasingly to integrate knowledge and concepts across disciplines. This partnership creates opportunities to address hard problems in surprisingly new ways. An important challenge is the determination of appropriate quality for software products and services. Too little quality leads to churn where users look for alternatives. Too much quality leads to poor return of investment because the development and operation of systems becomes unnecessarily expensive. The software engineering body of knowledge did not contain appropriate methodology to address this problem. It was just recently that a first method was proposed.

The QoE community would approach this challenge by studying the relationships between software quality and the impact generated by such quality experimentally. Unfortunately, such experiments are costly and hard to implement for software products and services with a non-trivial set of features. Too many experiments would need to be performed and the requirements on the expertise of the average software practitioner would be excessive.

We are looking for ways to alleviate the problem by sensible selection of QoE experiments to be performed and smart timing of requests for user feedback. Beyond doing the obvious, the automated execution of QoE tests, we aim at answering the following questions: (1) How can we generalize results from QoE tests across users, features, and contexts? (2) When is QoE data valuable enough to be collected?

The answers to these two questions will allow us to determine good-enough quality in a more lightweight manner than we do it today. While automation will ease the adoption of the techniques by practitioners, the reduced need for experimentation will reduce cost and annoyance that are generated with them. As a result, a much broader range of software products can benefit from QoE methodology and delight rather than disturb.

3.10 QoE for education services at primary and middle school

Juan Pablo González Rivero (Plan Ceibal – Montevideo, UY)

The success of one to one educational models relies heavily on the learning services and communications infrastructure that support them. The appropriation of new educational tools supported by technology requires users to be motivated and attracted to use these new services. When these models are carried at a national level, as in the case of Plan Ceibal in Uruguay, user heterogeneity, vast differences in experience with technology, cultural, socioeconomic and geographic variety make it particularly difficult to understand the different assessments about the quality of services experienced in the educational process. Another particular aspect of the case is that the dynamic is not the usual in which users pay for the service they want, in this case there is no payment and services are imposed. There is a

need for having quality indicators from the user perspective with the objective of acting proactively on services to ensure success in the appropriation of tools and in the improvement of learning. To cope with this, the techniques and tools of QoE are a promising way. This in turn brings big challenges:

1. **To develop methodologies and models of QoE for educational services.** The main services provided in education can be separated into two classes, videoconferencing based services and web adaptive learning platforms. Currently there are no models oriented to educational applications. The differences between both types of services and the particularities of each one enlightens the need for development of methodologies and specific models to monitor the QoE. In turn, the models to allow proper management should enable the root analysis and require adequate incorporation of the factors associated with the users. Regarding methodologies, classic laboratory tests would be inadequate. There is a need to develop ecological and indirect methods of assessing the QoE taking into account the particular characteristics of children and adolescents.
2. **Development of new QoE estimation tools.** For QoE-based services management, there is a need to develop and operate tools to estimate QoE in each of the relevant services. In the case of Plan Ceibal it is possible to develop tools in a distributed manner, deployed on network equipment, user terminals and centralized platforms. The challenges would be in the definition of the architecture and management of tools.
3. **Measure the relationship between QoE and its impact on learning.** QoE measurement as an input for deployment and optimization services is a key but not the ultimate goal. We sense that there is an intimate relationship between QoE services and its impact on learning, (ie. The relationship between the quality of audio in an English videoconferencing class and intonation learned by students). In order to understand the educational value related to the QoE the link between these two concepts needs to be taking into account.

3.11 QoE in digital ecosystems

Poul Einar Heegaard (NTNU – Trondheim, NO)

Most services are now provided over what is sometime referred to as a *digital ecosystem*. This is a heterogeneous system with multiple stakeholders/actors with one or more roles, over multiple domains, including content and network providers, and the end users. In digital ecosystems the various technical systems can be tightly coupled, while the control and management in the different domains exchange the management information only through what is given in a Service Level Agreements (SLA).

In this context it is important to understand what can be observed and controlled to enhance the QoE and the underlying QoS. This can be studied both from the user and provider perspective. The user perspective typically focuses on how the users perceive the service, but also more insight in what the user her/himself can do to change and improve her/his QoE is important. The (content/network) providers are more and more concerned with what they can do with respect to technical presentation and delivery, and on customer support and relations, to enhance the QoE from their users' perspective. For instance, the providers are interested in a "loyalty curve" over time and to understand how and why the

loyalty changes, and how the loyalty develops over time. Of particular interest is to get more insight in what the *long term effects* on QoE are, including use of both subjective test and objective measurement in real life environments.

A large number of QoE factors have been identified and the effects of these been (objectively and subjectively) studied and measured. The technical factors and impairments are mostly related to the performance of the services, but it is important also to increase the understanding of the effect on QoE of other technical, non-performance, such as *trustworthiness* factors, which includes dependability/reliability, (information) security, and safety.

The relative weight between the various QoE factors needs to be studied more. One approach is to use a modeling framework where a weighted convolution of the empirical distribution of the opinion scores of different factors, which gives an overall *distribution of the opinion scores*. The sensitivity of changes in the weights, and potentially in their cross-correlation can then be studied analytically. In general, it is very useful for the providers to know more about the the uncertainty of the opinion scores, measured for instance as standard deviation, quantiles in the distribution, or the probability of observations below an acceptance threshold.

It is a great challenge to define a *functional relation between QoE and QoS*. From studies of QoS, it is know that this is really hard to do even for QoS factors at different technical (protocol) levels, and across different domains. What is the best approach to define measurements and management strategies from the providers' perspective to enhance QoE? Furthermore, how can we operationalize the new QoE definition? For example, if we assume that the technical factors influence the annoyance level only, and that the annoyance is low and the delight will dominate. How can we then determine what level of annoyance will make the QoS-related factors, measured by technical attributes and MOS-score, to become a significant, dominating effect on the QoE (and not "just one of many factors")?

3.12 QoE++: From ego- to eco-system?

Tobias Hoßfeld (Universität Duisburg-Essen – Essen, DE)

QoE research has advanced significantly in recent years with a focus on the QoE ego-system. This means that QoE has been mainly addressed within a single session on a short-time scale for a single user of one concrete application. Thereby, different facets have been addressed by the research community, like subjective user studies to identify QoE influence factors for particular applications like video streaming, QoE models to capture the effects of those influence factors on concrete applications, QoE monitoring approaches at the end user site but also within the network to assess QoE during service consumption and to provide means for QoE management for improved QoE. Recent research has focused for example on HTTP adaptive video streaming which monitors QoE at the application layer at the end user site and adapts the video to the current network conditions. As major QoE influence factors stalling events, initial delays, and video quality switches have been identified and basic QoE models have been derived which build the basis for quality adaptation mechanisms. This enables service providers to improve resource utilization and QoE by incorporating information from different layers in order to deliver and adapt a video in its best possible quality.

However, in order to progress in the area of QoE, new research directions have to be taken. There is a need for QoE++. The application of QoE in practice needs to consider the entire QoE eco-system and the stakeholders along the service delivery chain to the end user. In comparison to the traditional QoE ego-system thinking, the QoE eco-system addresses among others the following research topics: in-session vs. global system perspective, short- vs. long-time scales when considering QoE, single vs. multi-user QoE, single vs. concurrent usage of applications and services, user vs. business perspective by addressing all key stakeholder goals. From the user's perspective, current QoE models mainly quantify the influence of various parameters on the perceptual quality. However from a service provider's perspective, it may be more relevant how the user is behaving, as a consequence of the experienced QoE, but also as a consequence of other context factors like pricing, privacy, etc. Thus, QoE++ requires (*a*) to extend current QoE models by the different perspectives of the QoE eco-system including the service provider perspective, (*b*) to incorporate user behavior as part of the model, and (*c*) to identify and include relevant internal and external context factors including physical, cultural, social, economic context.

QoE++ faces the following three major challenges. (1) Can we utilize QoE for network & service management? Or is it more appropriate to consider user engagement or user behavior? Which context factors are relevant or are such context-factors even more important for network & service management, e.g. in order to foresee and react on flash crowds? (2) Can we transform QoE into business models, SLAs, etc.? Or is it possible to 'trade' QoE? For example, offering WiFi sharing at home, a user may get improved service delivery and QoE by its ISP. (3) Do we understand fundamental models and natural relationships of QoE++? How can we extend existing QoE models to take into account the service provider's perspective? How can we include user behavior in the models? What is the relationship between QoE and user behavior?

Following QoE++ will shift from ego- to eco-systems and give answers to those questions.

3.13 Quality of Experience – objectives and challenges

Kalevi Kilkki (Aalto University – Espoo, FI)

QoE has become a popular topic during the last years, at least, in certain circles of communication networks and services. One of the main reasons for introducing QoE as a "scientific concept" was the limited usage of the concept of QoS. In practice, QoS was used to describe some technical attributes of communication services like packet loss ratio, delay and the available bit rate.

QoE is aimed to extend, compared to QoS, the analysis towards human and to economic aspects. This objective poses immediately a hard measurement challenge, because human beings are strange objects to measure. Our behavior is complex, often seemingly irrational, and always highly context-dependent. Thus an artificial (laboratory) arrangement is just one very specific context in which humans behave in certain specific way. Then if human behavior is observed in real, everyday situations, there can be huge variations which makes it extremely hard to make clear, statistically significant conclusions.

And finally, we are still lacking a general framework to assess in a systematic way what we prefer in our lives, not just what we prefer in a specific situation, for instance, when

watching to a video clip through Internet. Thus, the QoE framework shall locate in the area of human life instead of the area of technology (where QoS naturally to locate). Any QoE measurement result shall truly describe human experience.

3.14 QoE from a telecom operator's perspective

Eirini Liotou (National Kapodistrian University of Athens – Athens, GR)

In telecommunication networks, despite the catholic presence of Quality of Service (QoS) mechanisms, QoE has always been an "afterthought". This means, that although QoS provisioning has been an integral part of these networks, QoE has never been an original design intention. However, a QoS-based, system-centric view of the network is no longer sufficient, and it needs to be complemented or replaced with more user-centric approaches. This shift is currently an emerging, open challenge.

The acquiring of QoE awareness and the management of a network in a QoE-centric way may be beneficial not only to the end-users and telecom providers, but also to any other stakeholders involved in the service provisioning chain, such as service, content and cloud providers, or even customer care and support agents. Focusing on the telecom operator's perspective, we may identify three potential opportunities (and incentives, thereof) that derive from acquiring QoE and controlling a network in a QoE-centric way: (*a*) to increase the loyalty curve of the customers and, equivalently, to decrease customer churn, (*b*) to drive business operations and Customer Experience Management solutions, and (*c*) to cut costs by identifying and exploiting the non-linear relationships between QoS parameters and the perceived QoE.

Towards this direction, various research questions need to be addressed: (1) How can QoE be measured, monitored and controlled in telecommunication networks? Such a QoE management framework is essential before any operator-specific business decisions are made. (2) What kind of business opportunities are created for the operator and other stakeholders, such as OTT service providers, assuming that QoE can be managed? New QoE-based business models need to be designed, carefully considering Network Neutrality issues. (3) What is the new (more active) role of the end-user in such a QoE-aware/QoE-centric network (e.g., users may provide feedback about their preferences, priorities and experience)? Moreover, how can the end-user be convinced to "buy" QoE? Potential strategies of the network operator may include to "personalize" each end-user and provide QoE accordingly, or to build more aggregated user-profiles, facing the fact that the "average user" does not exist.

Among others, these questions should be considered as fundamental, before starting to design the QoE-centric networks of the future.

3.15 Motivations, consumer behaviour and QoE

Toni Mäki (VTT Technical Research Centre of Finland – Oulu, FI)

Quality of Experience is already well understood regarding the multimedia and telephone services. This understanding is established on good understanding of the human sensory

system, related mental processes and also well defined activities and so called motivational objects people are pursuing. Relation of these concepts is important when trying to understand the QoE of a given service. E.g., for multimedia services it is, while not easy, possible to see how motive (e.g. enjoyment) and mental processes handling sensory inputs are closely related. When moving towards non-multimedia services this coupling becomes looser. Gaining better understanding on this is a one challenge in modelling contemporary services. We are working towards a methodology that allows analysing these relations and how much of the QoE of a service can be accounted for technical quality in systematic manner (based on activity theory). My other interest and more widely a great challenge is quality-related consumer behaviour. How people experience services and how that translates into deliberate and also unintentional behaviour and decisions is a complex process.

3.16 What to do with Quality of Experience? A proposal for new research directions

Sebastian Möller (TU Berlin – Berlin, DE)

Whereas considerable progress has been made in defining, measuring and predicting Quality of Experience (QoE) for a number of commercial services, QoE still comes in the form of a self-directed object of measurement: QoE is measured in order to improve QoE. However, little attention has been drawn to what else can be done in case that QoE gets below or above a certain threshold. One could think of a mobile telephone service which switches to a different communication channel (such as SMS) when a call is lost during a train ride, and which automatically re-installs the call when the network connection is back for a sufficiently long period of time. Or, a video service which automatically starts recording and time-delayed playback in case that the IP-based television line is overloaded. Identifying potential use cases where QoE monitoring leads to new, proactive types of services first requires user studies in the field to identify hidden needs. Then, the technical feasibility needs to be analyzed, and finally the implemented services need to be evaluated as to whether they really provide additional value. In order to evaluate such pro-active services, it will be important to move beyond the classical borders of QoE measurement, by investigating utility and acceptance as a trade-off between user need fulfillment and QoE provision.

3.17 From Service Level Agreements (SLA) to Experience Level Agreements (ELA)

Peter Reichl (Universität Wien – Wien, AT)

Based on several years of advances especially concerning applications around media and Web, QoE research has now reached a state of maturity which allows (and at the same time demands) turning these insights into operational reality. In order to achieve this transition in a smooth way, it is proposed to learn from the rather successful concept of Service

Level Agreements (SLA) which today form a well-established broad base for a common understanding of quality provision on a networking level, i.e. with respect to Quality of Service (QoS) and the related characteristic parameters. Similarly, as far as Quality of Experience is concerned, it is now necessary to design and specify corresponding "Experience Level Agreements" (ELA) taking account of the specific user-centric perspective on service quality. This step still poses several key challenges, including the appropriate parametrization of quality features and user context factors as well as the applicability towards a broad spectrum of new fields approaching the horizons of QoE research, ranging from the Internet of Things up to arts and culture.

3.18 Acceptability towards a better User Experience

Miguel Ríos Quintero (TU Berlin – Berlin, DE)

In order to improve next generation audiovisual services, telecommunication companies traditionally focus on improving parameters based on Quality of Service (QoS). Unfortunately, QoS mostly focuses on performance and reliability parameters of the service. So far, this approach has proved to be insufficient to predict customer satisfaction.

The main concern is that every user has a different perception, based on several influencing factors. Factors such as emotional state and intrinsic user characteristics are often not considered or vaguely explored in traditional evaluation methods. Here, it is necessary to expand the view from a performance-driven approach towards actual Quality of Experience (QoE), addressing what a technical quality means to a subject in terms of accepting that quality for actual usage. In turn, an increased QoE may be beneficial in terms of how long a user spends with the service per usage, whether he actually purchases the service at all, and in terms of whether he stays with the service or drops it for the benefit of a concurrent service.

3.19 From momentary "within-session" QoE to a more global and ecologically valid picture

Werner Robitza (TU Berlin, DE)

In the domain of Quality of Experience we have reached a level where we are well equipped to give valid predictions of perceived momentary audiovisual quality or delight/annoyance, that is answering the question of "What would the quality rating be if we asked an average person now?" However, this only works for small time frames, from several seconds up to a few minutes, and only considers the typical user, since in traditional tests, all ratings are combined into the Mean Opinion Score (MOS). However, the remembered quality and experience for a user is not necessarily just the average (or a simple function) of the hypothetical momentary quality in terms of a Mean Opinion Score.

When we deal with the challenge to predict experienced quality for a longer time period, a specific region, or a group of people, how is the value of QoE meaningfully integrated over

time? From a similar perspective, we could also ask: How does it manifest for different kinds of users? This is a challenge to be solved in the near future.

To this aim, we also need ecologically valid test settings and models that go beyond short term stimuli of just a few seconds, content that was purposely created to have no influence on quality ratings by being boring, and "within-session" experiments, where users have no way to interact with a service. By putting the human into the role of a passive viewer, when in reality they would actively experience a service, we are invariably getting test results that may not be representative. Standardization efforts in this direction are crucial to enable a global understanding of what these concepts entail. For example, a collection of general guidelines for more ecologically valid tests would have huge benefits for the future evaluation of existing services and creation of new experiences altogether. In the long term, we expect QoE models to be based on such tests.

But what else could be improved when performing tests with users? Humans behave differently in the presence of good or bad QoE. They may adjust their behavior so as to improve their positive affect towards a service, or they may show signs of aggression or confusion. The inclusion of behavioral patterns in QoE studies will help improving services and delivering better QoE for the specific user in the long run.

3.20 YouSlow – why and where is YouTube slow?

Henning Schulzrinne (Columbia University – New York, US)

Joint work of Nam; Hyunwoo; Kim, Kyung-Hwa; Calin, Doru; Schulzrinne, Henning
Main reference H. Nam, K.-H. Kim, D. Calin, H. Schulzrinne, "YouSlow: A Performance Analysis Tool for Adaptive Bitrate Video Streaming," in Proc. of the 2014 ACM SIGCOMM Conf. (SIGCOMM'14), pp. 111-112, ACM, 2014.
URL http://dx.doi.org/10.1145/2619239.2631433

YouTube is one of the most popular media delivery platforms, and many users equate the QoE of YouTube with their Internet experience overall. The "buffering" circle of YouTube and similar applications has become iconic. However, it is often very difficult or impossible for users to determine why their experience is bad or highly variable – is it a local Wi-Fi problem? Maybe a microwave oven or baby monitor? Access network issues? NATs, proxies and firewalls? Network interconnection disputes? We built YouSlow, a follow-on effort to our DYSWIS (Do You See What I See), to map the performance of YouTube on a global scale, and see clear patterns in user-visible QoE artifacts, i.e., re-buffering.

3.21 Potential for integrated/cooperative QoE management schemes

Lea Skorin-Kapov (University of Zagreb – Zagreb, HR)

We are witnessing many different players involved in end-to-end service delivery, ranging from various cloud providers, content providers, network operators, etc. We are further witnessing what has been referred to as a paradigm shift towards an Internet of Services, envisioning everything on the Internet as a service. Such a transition will potentially lead

to new services being realized as service chains combining and integrating the functionality of (potentially many) other services offered by third parties (e.g., infrastructure providers, software providers, platform providers, etc.).

QoE models dictate the parameters to be monitored and measured, with the ultimate goal being effective QoE optimization strategies. Hence, the question is how can new QoE models be exploited (in a practical sense) in the context of QoE management schemes? The majority of QoE-based management approaches to-date may be primarily related to either network management (based on monitoring and control on access and core network level) or application management (adaptation of quality and performance on end-user and application level). For example, OTT services (e.g., YouTube, Netflix) delivered by third party service/content providers commonly implement QoE control schemes on the application level. As a result, different QoE optimization and control loops are involved (e.g. dynamic application adaptation, network management), with the question being how do they interact? A further question is to what extent can cooperative management schemes and underlying business models involving multiple players achieve more efficient management of network/system resources, while enhancing customer QoE? What are the stakeholder incentives? What (novel?) solutions are needed for coordination and information exchange among actors involved in the service delivery chain in order to provide channels for effective QoE control/improvement? Importantly, the aforementioned questions should also be considered in the context of regulatory policy and the ongoing network neutrality debate.

3.22 QoE assessment in new generation of multimedia services: the need for adaptations

Samira Tavakoli (Universidad Politécnica de Madrid – Madrid, ES)

Nowadays QoE is a valuable and well-known concept in the multimedia services. Because of the trade-off between the user's QoE and the cost for providing the high quality services, content providers find themselves in need of understanding the perceived quality of provided service to the user. Numerous researches in this regard have been conducted with an ultimate goal of optimizing the user's viewing experience. Considering the subjective assessment as one of the common approaches to evaluate the QoE, several test methodologies have also been provided as international recommendations. However, by immersion of new generation of multimedia services, the suitability (even validity) of these methods to accurately illustrate the perceptual quality of these services, is questionable. As addressed by previous studies, it is quite likely that the relative impact of impairment type changes with the setting of subjective experiment.

As an example of new multimedia services, the HTTP Adaptive Streaming (HAS) can be mentioned, which has gained widespread popularity as a cost-efficient way to distribute pre-encoded video content. With adaptive streaming, it is probable that the quality switching takes over periods up to several minutes, providing a novel type of impairment which is time-varying quality sequence.

Most common evaluation methodologies, like Absolute Category Rating (ACR) recommend the use of short video sequences to be evaluated by the test subjects. However, in the case of some techniques such as HAS, using short test sequences cannot be appropriate. It is

not clear whether the perceptual quality of individual stimuli (cf. only adaptation event) would be the same as evaluating the stimuli when occurring in a longer sequence. Other approaches such as Single Stimulus Continuous Quality Evaluation (SSCQE) could also not be suitable. This is because the effect of recency and hysteresis of the human behavioral responses while continuously evaluating the quality could lead to an unreliable evaluation through this methodology. On the other hand, in traditional testing methodologies, the quality of the video in audiovisual services is often evaluated separated and not in the presence of audio. Nevertheless, the requirement of jointly evaluating the audio and video within a subjective test has been addressed by some previous studies.

From another side, in regard to subjective experiment planning, having multiple technical variables in such services (in the case of adaptive streaming these parameters could be dimension, amplitude, frequency and period of adaptation), makes the full-matrix design not feasible. This implies that the experimental results are limited to few number of content types or test conditions, so that making a generic conclusion becomes impractical. Another question that can be raised is whether lab/crowdsourcing studies give a reliable knowledge about "actual" application of these techniques (e.g. evaluating the perceptual quality in live streaming event). To summarize, novelties of new multimedia services such as adaptive streaming, highlight the need to investigate new assessment approaches compatible to the nature of application under study to reliably picture the real user's perceived quality.

3.23 Decompiling QoE

Christos Tsiaras (Universität Zürich – Zürich, CH)

Joint work of Tsiaras, Christos; Stiller, Burkhard
Main reference C. Tsiaras, B. Stiller, "A Deterministic QoE Formalization of User Satisfaction Demands (DQX)," in Proc. of the 39th IEEE Conf. on Local Computer Networks (LCN'14), pp. 227–235, IEEE, 2014.
URL http://dx.doi.org/10.1109/LCN.2014.6925776

Measuring the impact of technical variables, such as latency, bandwidth, or resources priority-access, on Quality of Experience (QoE) of various services demands an extensive feedback from end-users, when those variables change. Estimating QoE in a given scenario becomes harder, when non-technical variables, such as price, need to be considered in addition to technical ones. In any case, detailed feedback that correlates all variables affecting QoE is needed by end-users for each service separately. A deterministic mathematical model (DQX) [1] encapsulating user demands, service characteristics, and variable specifications is proposed to formalize the QoE calculation, considering one or multiple and diverse variables. The output of QoE functions presented in DQX can be normalized such that results will be compatible with the five-point scale Mean Opinion Score (MOS), proposed by the ITU-T.

References

1 Christos Tsiaras, Burkhard Stiller, *A Deterministic QoE Formalization of User Satisfaction Demands (DQX)*. The 39th IEEE Conference on Local Computer Networks (LCN), Edmonton, Canada, 8–11 September, 2014.

3.24 Application of QoE to create business value in telecom

Astrid Undheim (Telenor Research and Development – Trondheim, NO)

Network quality is seen as one of the most important drivers for customer satisfaction in Telenor, and the user perceived network quality is measured using the Net Promoter Score (NPS) framework. The results are valuable for insight into capacity problems for specific base stations etc, however the NPS measurements are based on limited sms-feedback. There is clearly a need to introduce real-time, per-application QoE measurements based on available data from the network, handset and application.

At the same time, QoE measurements should be operationalized, and without explicit user feedback. One way to achieve this is to look into new QoE measures such as: how often a service is used, how long sessions, etc. These are implicit QoE data that can more easily be mapped to the status of the network, and that even allows for longitudinal studies.

Machine-learning methods and big data technologies are attractive to go one step further in operationalizing QoE, through building hypotheses on the influencing factors for QoE, and testing these in real operation. The resulting model will give insight into which factors are most important for customers in given situation, as well as help predicting the QoE in similar situations. These models and methods will provide actionable insight for a large set of customers, with increasing accuracy as the model learns from new customers, situations and service usage.

3.25 $elling QoE

Martín Varela (VTT Technical Research Centre of Finland – Oulu, FI)

QoE has been a trending research topic for a few years now, and it has become a buzzword in some business domains as well. However, most (almost all, really) of the work done on the field has been either for modeling/estimation/measurement or as a tool to guide technical improvements (e.g. QoE-driven management). I believe that the next big advance in making QoE operational will come from showing to the different stakeholders that QoE is worth ($$$) caring about. My question is now, how do we go about it?

3.26 QoE in next-generation networks and services

Min Xie (Telenor Research – Fornebu, NO)

As a research scientist at Telenor, an operator with a strategy prioritizing Customers, we aim to provide services that are perceived by users as satisfactory services. To do that, we need to understand how users experience and then operate to optimize user experience.

The first challenge is how to measure and analyze QoS and QoE in the mobile networks. Mobile operators can gather data from both the network side and the user side. We need measurements from the users perspective to reflect their perception of the delivered services. However, the user end device has limited functionality and resource whereas the number of variables needed to be measured could be huge. It is therefore our task to decide which metrics to measure and where to measure them, i.e., to distribute the measurement tasks properly between operators and users (also service providers). The collected measurements need to be analyzed efficiently, e.g., via machine learning, to identify the role and significance of different variables for different services. Identifying the key QoS metrics and their thresholds would help operators to adapt the network operations more effectively to serve users.

The analytical results serve as the base to solve the second challenge, predicting QoE without or with limited user feedback. In a mobile network, it is not realistic to perform subjective QoE assessment towards individual users to get the real-time QoE indicator. Instead, it is more practical to create a QoS-QoE map and estimate the QoE based on available QoS measurements. The challenge lies in the limitation of available measurements and the influence of contextual factors such as emotion, environment, and social status. An accurate QoE prediction would help operators to allocate resource more efficiently and react to potential QoE degradation proactively.

With a better understanding of user experience, we then aim to define delivered service quality from the perspective of Experience Level Agreement (ELA), as comparison to SLA. The ELA will describe the service quality in terms of "experience", which could be better understood by the users. On the other hand, it will keep the quantitative specifications of the service quality as references for operators. This ELA concept is expected to bridge the gap between operators and users so that they can communicate in a same language on the service quality.

3.27 On QoE monetization and forming a global customer satisfaction metric

Patrick Zwickl (Universität Wien – Wien, AT)

Main reference P. Reichl, P. Maillé, P. Zwickl, A. Sackl, "A fixed-point model for QoE-based charging," in Proc. of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking (FhMN'13), pp. 33–38, ACM, 2013.

URL http://dx.doi.org/10.1145/2491172.2491176

The transition from theory to practical applicability of QoE is exacerbated by the following challenges: (1) monetization and utilization/exploitation strategies for QoE information, (2) formation of a global QoE picture (bridging the bounds of individual test ranges, model bounds, and current biases), and (3) concepts easing the handling of the QoE marketisation complexity.

The monetization or other kind of utilization of QoE information is notoriously difficult, as it involves the fixed-point problem between willingness-to-pay and service perception [1], but also immanently requires the integration of other factors, e.g., socio-economic classification of users and the service usage preference (in the current context or in general). This is further hampered by the non-linearity of QoE metrics, which is contrasted by the requirement for linear ISP and user utility presentations whenever feeding QoE results into economic optimization models.

A related major obstacle is the absence of a global QoE view, which overcomes isolated QoE results, which are bound to single test case, test scenario, methodology, test limitations or biases, and parameter settings. Today's QoE results are hardly comparable across service types and cannot be regarded to be a generically comparable measure, e.g., comparing the perception across two different kind of services. However, the mapping from QoE to any kind of utility representation will require an understanding of the service usage preferences across service types and quality levels by individual users for various kind of quality conditions, forming a key block for the overall valuation of services, i.e., utility. This will likewise also affect the ISP utility perspective in terms of prices that can be charged. For instance, when comparing fictive test results around input QoS test range interval $[A, B']$ where $B' > B$ ("$>$" means better) with a separate testing of $[B, B']$, then we can expect to receive inconsistent results. This is even more problematic for across service type comparisons, which mind the service usage preference of the user.

Complexity is at least a three-dimensional problem in operation, which involves technology, economics, but also users:

- "Simplicity" is an important factor from a technological point of view – reducing the complexity wherever possible, increasing the scalability whenever doable. For this reason, it is important to analyse the tradeoffs between profiting from QoE-aware service differentiation and rising complexities in parts of the system (e.g., client, access or core network) when doing so. In particular, strategic decisions like shifting complexity from the network to the clients (e.g., adaptive streaming) may strongly affect the technical complexity.
- On the business axis, we can differentiate in long term and short term effects: In the short run the generation of new revenue streams exceeding the costs of additional resources is in focus. In the long-run, the loyalty of customers may be influenced by adequate QoE levels. Both short and long term perspective have to be addressed in concert by appropriate QoE marketisation frameworks in order to obtain a sustainable QoE market configuration.
- On the user side, a high "usability" when offering QoE upgrades is crucial but difficult to obtain, as network quality is an experience good that can hardly be communicated in advance. In practice, experience simulators or past experiences of similar users may provide helpful indications on adequate quality and price combinations prior to the actual purchase and experience thereafter.

In general, a separate willingness-to-pay testing for network quality is more realistic than meaningful approximations, however, efforts should be dedicated to the better interrelation of individual test results and their projection to untested cases.

References

1 Peter Reichl, Patrick Maillé, Patrick Zwickl, and Andreas Sackl. A fixed-point model for QoE-based charging. In *Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking (FhMN)*, pages 33–38, 2013.

4 Working Groups

4.1 QoE theory and modeling

Sebastian Egger (AIT – Wien, AT)
Peter Reichl (Universität Wien – Wien, AT)

While the modeling of QoE has made significant advances over the last couple of years, currently existing models still lack an integration of user behavior aspects and user context factors along with the consideration of appropriate temporal scales. Therefore, during the discussions of this group, a comprehensive QoE and user behavior model has been developed, providing a framework which allows joining a multitude of existing modeling approaches under the perspectives of service provider benefit, user well-being and technical system performance. In addition, the role of a broad range of corresponding influence factors has been discussed, with a specific emphasis on user and context issues, and a series of related use cases have been identified which are suitable to validate the proposed model. During and after the discussions in this group, joint efforts resulted in a conceptual paper [1].

References

1 Peter Reichl, Sebastian Egger, Sebastian Möller, Kalevi Kilkki, Markus Fiedler, Tobias Hoßfeld, Christos Tsiaras and Alemnew Sheferaw Asrese. Towards a comprehensive framework for QoE and user behavior modelling. In *Proceedings of the 7th International Workshop on Quality of Multimedia Experience (QoMEX)*, May 2015.

4.2 QoE methodologies

Katrien De Moor (NTNU – Trondheim, NO)
Werner Robitza (TU Berlin – Berlin, DE)

The goal for this working group was to discuss challenges, problems and potential ways forward in the context of QoE methods and methodological approaches. The challenges that were identified during the initial presentation sessions were further clustered into 3 main and partly overlapping discussion topics: (1) the need to increase the ecological validity of QoE research inside and outside of the lab, (2) specific operationalization issues (measuring the "real" QoE), and (3) other issues regarding the "when" and how to measure QoE? For each of these topics, main issues and potential ways to address them in a better way were discussed.

4.2.1 Increasing ecological validity inside and outside the lab

Ecological validity refers to the extent to which a study design resembles and reflects the real-world situation and thus, the extent to which empirical findings can be generalized to real-world settings. In traditional (standardized) QoE studies, the ecological validity is very low and problematic in several ways:

- The results of such lab studies may be strongly biased (leading to "wrong" results and prediction models) because the stimuli and conditions are artificial and not necessarily representing the real-world situation.
- Test participants are very focused on quality degradations or specific properties of the system, as they are usually instructed and primed by the researcher to do so (e.g., in the briefing and instructions, in questionnaires). This is usually not the case in a real-life situation.
- It is very difficult to keep people engaged in a test which takes place in such an unnatural controlled lab situation. Yet, users' engagement is crucial in the context of QoE, so this poses a challenge which needs to be addressed.
- The current standards and recommendations are not adapted to ensure a higher ecological validity of research findings, yet they play a crucial role as they essentially define the terms of common understanding, methodologies and measures, which will be used by different stakeholders (from research, industry, ...) and for different purposes.

Without abandoning these current standards completely, there are certain aspects that could be changed relatively easily in order to work towards the goal of creating a more ecologically valid setting inside the lab. This is a relatively short-term goal which could be reached by e.g.,

- Introducing more immersive test paradigms: not repeating any content, because repetitions are unnatural, thus leading to boredom and increased attention to minuscule details or the test procedure itself. Immersive testing has already been shown to be practical and leading to less frustrated test participants [1].
- Having clear strategies to try to keep the attention high. For instance, give the participants a task that helps to create sufficient attention throughout the whole test (e.g., incentive per watched video, remembering details or replying to questions about the content afterwards and when correct answers are given, there is an additional reward, ...).
- Including longer duration and thus more representative test stimuli and using meaningful and real content (e.g., in audiovisual tests).
- Trying to make the evaluation process as unobtrusive as possible. This could be done by e.g., also capturing implicit feedback from users (e.g., through behavioral measures) and matching it with the captured explicit feedback (e.g., user ratings). A mismatch between both feedback sources could help to identify issues that threaten the validity (e.g., the observers were not paying attention, experimenter bias, ...).

The future vision (longer term) goes even further and includes:

- Development of new recommendations (or amendments to existing ones, such as ITU-T P.913) on how to investigate QoE in different real life environments (e.g., home, mobile, ...), with guidelines on appropriate methods to use, the combination of qualitative and quantitative approaches in an optimal way, tools that can be used for capturing behavior and monitoring technical parameters, etc.
- Evaluating QoE over longer time periods (cumulative or longitudinal QoE) instead of focusing on one particular moment in time (as in traditional test settings).
- Additional issues that have to be considered in this respect relate to e.g., privacy concerns and motivation (e.g., how to motivate people to participate in longer duration studies).

4.2.2 Operationalization issues

When it comes to the application of QoE, several issues were brought up:

- In the new, broadly-supported definition, QoE is defined as a degree of delight or annoyance, which is relative to a range of aspects (e.g., the user's expectations, current state, personality, ...) but the dominant measure (perceived overall quality) does not allow to capture delight or annoyance.
- The decision on which measures are most appropriate to include is depending on various issues, e.g., which modality/modalities? Which type of use case (e.g., mainly enjoyment vs. mainly utility-oriented?), Which target group (e.g., children, elderly people, ...).
- Depending on the modality there may be perceptual differences (e.g., age- or gender-based) but to gain deeper insights, we need to systematically gather and report more information about the participants and their characteristics: currently this is very limited.

What is needed?

- Complementary measures that operationalize the new QoE definition as well as guidelines on (1) how to use these measures in practice, and (2) on how to select the most relevant measures. Examples are e.g., behavioral measures (during experience, but also behavioral intentions after use), but also complementary self-report measures, such as Self Assessment Manikin (Valence, Arousal, Dominance), validated scales measuring abstract constructs such as user engagement, immersion, flow, expectations, mood, ...
- Guidelines on how to instruct and train test subjects (e.g., "don't overthink", which is really important when including e.g., measures of emotion, engagement, ...).
- In order to be able to really advance the understanding of how both stable and more dynamic human factors may influence or be related to QoE, we need to gather more information about the test subjects, so that we can describe the test panel better and look into different groups or segments (e.g., depending on affinity with technology, adopter profile, attitudes, personality traits, ...). There is also a strong need for guidelines on what to include, when and how, and how to use and report on this information later on. This topic was further discussed in the group discussion on user characteristics.
- Better guidelines on the number of participants that are needed: this strongly depends on the setup of the study and the goals (in some cases, a limited number of test subjects is enough, but in other cases it is not).
- Inclusion of qualitative questions in order to gain better insights into the "why" dimensions (and e.g., to check at the end of a test whether the "right" questions were included).

4.2.3 When and how should QoE be evaluated?

When should we measure QoE? This question comes up in several situations, both in a lab context and in real-life environments. When should a network operator do active or rather passive monitoring? When conducting a test, should we collect implicit or rather explicit feedback (or a combination of both)?
Specific questions were brought up:

- Should we measure QoE only when it is a complete "unknown"? Should we measure it only when a problem has occurred or rather pro-actively? Or rather in situations when a certain QoE-based decision has to be made (e.g. reduce resources while still guaranteeing an acceptance level of 80%)? How about in the context of next-generation services where tolerance levels are unknown?
- Is there an optimum point of asking during a subjective experiment or evaluation study?
- How to motivate and engage users to participate, without biasing them or unintendedly increasing certain acceptance thresholds?

- What are the most optimal incentives? By giving too many incentives, the risk is that people may be trying to simply "please" the experimenter. This was a problem e.g., in a study conducted in Uruguay (Plan Ceibal): because they received a free computer, the children involved in the study always rated the quality good despite intrinsically showing different behavior in the presence of bad network conditions.
- When including simple binary questions (e.g., "Is this quality level acceptable?"), there is a risk that internal quality threshold information gets lost with each further question.

We discussed several solutions and developed suggestions on how and when to measure:

- In general, it may be useful to consider first of all whether asking users for direct feedback could be avoided (e.g., by using other approaches, such as A/B testing). In some cases it may also be left up to the user, so that they can give feedback only when they want to (e.g., Skype).
- In the best case, the experience which is under investigation (e.g., game experience) should not be interrupted. In case explicit feedback is gathered, rather place it between two tasks.
- Another rule of thumb is to always try to lower the burden for the participant as much as possible, so for example in selection of measures to include: be selective and only include the constructs that you are most interested in (based on previous research, theory, ...).
- It may be useful (e.g., in a video quality test in the lab) to record rating and viewing behavior during a test and show it again to the test subject afterwards, while encouraging the subject to reflect on why he or she gave a certain score (did you see a change, how would you describe it, ...). This may provide valuable additional insights.
- Both in real-life and lab settings, it would be valuable to more explicitly inquire after personal priorities of users and customers of a service, as these priorities may differ a lot from user to user. However, gaining such richer information does not necessarily mean that only individual differences need to be considered. There may be patterns that can be studied for small groups of users and deeper insights in individual priorities could help to identify and to understand various user profiles/user groups, and to potentially derive different QoE models per profile or, at least, adjust the QoE provisioning procedure accordingly.
- When asking for explicit feedback related to QoE (e.g., in a real life setting), this should in an ideal case also have added value for the user or customer. However, this is not always easy to put into practice.
- It is also important to give users acknowledgments for their feedback, so that they know and believe that their feedback will really be considered and is useful.
- To increase the motivation and avoid biases, one strategy is to provide an intrinsic value instead of an extrinsic incentive, e.g., status (e.g., just mere participation), the value for the user is a functionality of the service itself (e.g., a free movie when evaluating a video on demand service).
- Involving users as early as possible may lower QoE issues with end product or service, and thus put a kind of "QoE by design" strategy into practice.

4.2.4 Conclusions

During the constructive and fruitful discussion during this group session, we identified a set of open issues and challenges related to the current methodological approaches and ways in which they could be adapted in order to increase the ecological validity of QoE research, put the new definition of QoE into practice and hereby enable new theory building

and better understanding of fundamental underlying relationships between QoE and its assumed influence factors. As follow-up work, bringing together practical guidelines and recommendations for more ecologically valid QoE research, and developing a decision tree for setting up experiments and selecting appropriate methods and measures, would be highly valuable.

References

1 Margaret H. Pinson, Andrew Catellier, and Marc Sullivan. A new method for immersive audiovisual subjective testing. In *Proceedings of the 8th International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, Jan. 2014.

4.3 User factors and QoE

Katrien De Moor (NTNU – Trondheim, NO)
Toni Mäki (VTT Technical Research Centre of Finland – Oulu, FI)
Werner Robitza (TU Berlin, Berlin, DE)

There are various reasons for considering user factors within QoE research. Regarding the economic and business values the following motivations were found important during our discussion:

- How to ensure customers are staying?
- How is the "threshold" for churning changing over time and impacted by rival services or adaptation over time?
- Incorporating user factors is important for marketing strategies, e.g., in customer segmentation.

From theoretical point of view following reasons to record and consider user factors were identified:

- User factors may help explain differences in ratings.
- It may be possible to create adaptive QoE models based on user factors (i.e., models targeted for different user groups).

Generally, internal reference models and internal criteria of human beings are influenced by previous experiences and differ among users. For example, the same rating behavior may be displayed but internal decision criteria may be different.

Practically, however it is still largely unresolved, how user factors can be integrated in the cycle of developing and conducting experiments, creating models and predictions, applying them in practice, and refining them for new services or new user groups.

4.3.1 Experiment lifecycle

The Figure 1 illustrates a typical flow of an experiment. The emerged ideas related to each phase are shortly discussed below (implicitly showing the focal points of discussion as well).

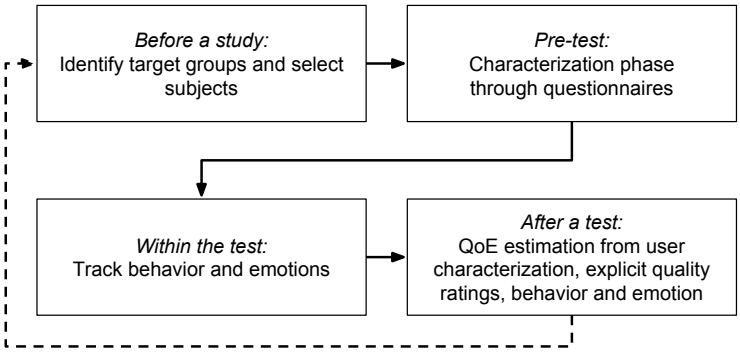


Figure 1 Experiment Lifecycle.

a) Before a study

The purpose of the study to be typically defines some ideal target groups from which the subjects can be selected. For example, in case of business oriented study, it is important to cover the real market segments. Here, we refer to marketing literature where a lot of existing information can be exploited.

b) Pre-test

Typically some demographic data is collected in QoE tests, including features like age, gender and expertise. Still, they are rarely even considered and typically not collected in standard fashion. There are numerous metrics or factors that could be considered:

- (Socio-)demographic factors such as age, gender and profession.
- Physiological factors such as long-term personality traits or short-term moods.
- Expectations which relate to various aspects such as motivations of users (gratification), attitudes towards technology or other people, and experience and relation to the service (e.g., frequent vs. casual gamer).
- Willingness to pay and spending behavior (how much and when does the user spend typically?)

At the moment, there is too little data, often lacking structure, to be able to determine which factors are influencing quality ratings significantly. Typical experiments use a small number of subjects, and both test stimuli and duration are short. All this makes it difficult to identify relevant factors. However, we think that the community should continue to encourage experimenters to continue to collect and report at least the typical factors, *in a systematic manner*, in order to gather at least descriptive and qualitative data.

In the long run, though, it would be highly desirable to identify and validate the important factors with quantitative results from tests with higher number of subjects and long-term duration. There are validated questionnaires in our field and related domains, which could serve as starting point. It may be necessary to define a new set of questionnaires or complement the existing ones. With understanding gained from such a study, we could come up with reduced and standard pre-test questionnaire.

It is also possible that the clustering of users based on user factors is sensitive to the service. Can we expect that they are the same for different services or not? Currently, such knowledge does not exist. One may be able to group similar media types, such as audio and video.

c) Within a test

Users have an internal reference of quality which they compare the currently experienced quality against. This internal reference varies per person, but may be clustered into groups. In addition to internal reference, subjects differ, e.g., in terms of sensitivity to stimuli and many physiological aspects.

Moving towards exploiting QoE, and going beyond perceived quality, user behavior was also discussed. Monitoring behavior during the tests may give insight of how people react to different quality levels.

Validity and reliability of rating. Each person assesses quality according the their internal reference and judgment process. So even if users agree on the rating, how they get there and because of what reasons, may differ. It is important to also note that these internal references change over time. So, how can we know do ratings present what we think they represent? How to create the picture of an internal reference of the observer?

There are several existing measures, for example: self-reports such as confidence ratings, recording rating behavior such as rating times, and using passive objective methods such as EEG measures, to identify internal cognitive thresholds, i.e., did the user notice anything?

Behavior. As with user factors, there are multitude of behaviors and metrics. To investigate the effect of QoE on behavior we need to include behavioral measures, but which ones? How to classify different types of behavior? Could the measures from related domains, such as UX, be exploited to complement QoE studies? The following issues were discussed:

- Two main types of behavior were identified: Adaptive behavior that may help in maintaining higher QoE (e.g. increasing pauses in conversation when the delay is high), or expressive behavior that a user may display, but that does not help to improve the QoE (e.g., aggressive clicking).
- Including behavioral measures (and observing user behavior) is possible to some extent in the lab, but may be more interesting in more realistic settings. Longer and realistic settings could enable us to track "default" behavior for certain users or user groups and then detect anomalies.
- Including behavioral patterns as part of user factors (i.e., user groups), would enable to create panels of users where behavioral patterns are better known. Such a capability would be highly valuable, e.g., for campaigns studying economic viability of services or features.
- Currently, most test paradigms enforce behavior in the sense of asking people for a quality rating in defined scenario, where the subject is told to imagine himself or herself performing a certain task. More ecologically valid tests would aim at giving realistic tasks and observing more "natural" user behavior.
- Also, the tests typically force people to focus on one service, and do not allow them to use other services or applications at the same time. How could we integrate the possibility of multi-tasking into a lab test? For example, allow them to do whatever they want at the side while doing a test and observing their behavior?

d) After a test

Typically, we would like to complement the existing QoE predictions with the collected user characteristics, behaviors and emotions to provide better accuracy. Understanding the relations between user factors, technical factors and behavior and emotions allows also creating prediction models related to behavioral aspects, such as willingness to pay.

4.3.2 Research questions and open issues

- Which user factors are actually the important ones?
- Can we use existing validated questionnaires to capture these factors or do we need our own set of validated questionnaires?
- What is the minimal set of factors that everyone should collect and report?
- What types of behavior are there and how they should be measured?
- How can we track behavior across multiple services and longer time intervals?

4.4 QoE management

Henning Schulzrinne (Columbia University – New York, US)
Martín Varela (VTT Technical Research Centre of Finland – Oulu, FI)

4.4.1 Architecture

We believe that many networked applications can benefit from following a QoE-enhanced architecture, shown below. In that architecture, the application, shown as "application", is observed by an application-specific observer function. The observer may, for example, gather information about video stalls, start-up latency or user interactions for a video playback applications, or audio gaps and speech behavior for a conferencing application. For web-based applications, many of these observations can be made on the web server, but JavaScript-heavy applications may also need local instrumentation. The applications, using operating system interfaces to be defined, also provide information about network characteristics (carrier, packet loss) to the application provider. The application is augmented by an advisor function that detects when the application is not performing well, provides an indication to the user, preferably indicating the underlying cause at an appropriate level of technical detail, and offer recommendation to the user on how to improve the experience. In some cases, the application can adjust the behavior automatically. The predictor function tries to anticipate problems, particularly for applications with long-lived sessions, such as video playback and interactive communication applications or multi-player games. For example, it might warn the user that the network conditions are unlikely to support a smooth application experience. The application provider filters the incoming application and low-level observations, possibly with the assistance of machine learning. The filtered information then may be visualized, but more importantly, feeds the advisor and predictor functions. Individual user observations may be combined by appropriate crowd-sourcing, e.g., correlating metrics within the same household, the same ISP or same computing platform.

4.4.2 QoE repair models

Traditionally, consumers notice degradation in QoE, contact their application or network provider by phone, and the help desk walks the customer through a series of manual steps, including short-term actions ("reboot") and longer-term decisions ("upgrade your service"). The help desk staff is essentially blind and limited to indirect observations. More recently, the help desk functionality has been enhanced by the ability to see some customer-specific metrics or install remote login tools. The long-term goal should be to avoid the need to call customer support – problems are detected, diagnosed and repaired automatically.

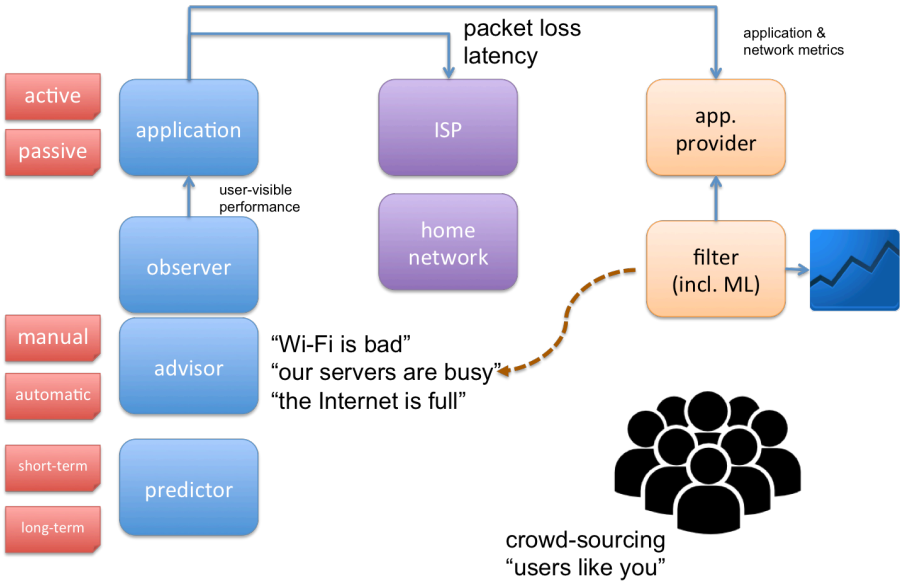


Figure 2 High-level architecture for QoE management.

4.4.3 Privacy concerns

Effective feedback loops may expose additional user data, such as user behavior, to the ISP or application provider. Thus, such systems must be designed to minimize information leakage to third parties, and ISPs and application providers must communicate clearly what information is gathered, how long it is retained in individualized or aggregate form and whether it is also used for purposes other than diagnostics.

4.4.4 Research questions and open issues

- **Intersection with other research disciplines.** Understood within a closed-loop context, the approach has strong overlap with other traditional and emerging research communities, such as classical network management, Software Defined Networking (SDN), smart data pricing and cross-layer design.
- **IWhat are the interfaces?** All key system components, including the operating system, home network components, the ISP, application and browser need suitable APIs, whether local (JavaScript) or HTTP-based. It is not yet clear what information should be exposed, to whom or at what time scale or granularity, or what functions need to be controllable.
- **What are the actuators?** Many of the components listed only allow users to set properties through configuration, command line or human-focused web interfaces, not APIs. For example, it would be useful if ISPs had interfaces that allowed authorized applications to increase the data cap or temporarily increase the connection speed. Home network gateways may need the ability to differentiate various users, e.g., provide priority to work-related applications.
- **What are the control loop properties?** Control loops needs to fail safe, so that applications continue to functions reasonably well even if some of the external functions cannot be observed or controlled. Control functions need to avoid adding to congested network and contribute to signaling storms. Feedback may be directed to more than one recipient in the control loop (e.g. ISP and application provider).

- **How can we support root cause analysis?** Today, determining the cause of intermittent QoE problems is often tedious and time consuming. What kind of information can components gather to simplify post-incident root cause analysis? Should there be an emulation capability so that users or programmers can easily replicate possible root causes, such as packet loss or intermittent connectivity? Like CCTV, better logging may not prevent the QoE crime, but help in apprehending the QoE perpetrator.
- **How can we anticipate and prevent problems?** Systems should be able to predict and anticipate flash crowds, for example, on the timescale of at least minutes, sufficient to marshal additional computing and network resources.
- **We need a new design-focused approach for QoE.** Research is needed to identify the appropriate overall system architecture that allows components to communicate in order to improve QoE. Standards bodies then need to define appropriate component-specific standards and data formats. Finally, both networking and software engineers need to be trained to towards a measurement-focused, whole-system approach of system design.

4.5 Monetization of QoE

Martín Varela (VTT Technical Research Centre of Finland – Oulu, FI)

4.5.1 Introduction

The goal of this working group was to discuss issues related to improving business (for any type of online service providers) by exploiting our knowledge of QoE. Given the group composition – with many participants from a networking background – the discussion had a distinct networking flavor, but OTT and cloud services services were also considered.

4.5.2 Main issues found...

a) In QoE-based business models

There are several key aspects to understand if we are to use QoE as a business-improving tool. The first one of those is how to convey what quality *feels like* to the end users. A lack of reference points for quality in most services (e.g., as "toll quality" was for the PSTN) presents a major roadblock for selling QoE to end users.

Secondly, from the operator's perspective, it is important to understand and measure the users' willingness to pay for the service, possibly at different quality levels. We should note here that the different quality levels might imply not just a different average MOS score, but rather differences in service dependability (consistency) or trustworthiness, for example. Metrics for these factors are currently under-developed topics in our community, and their impact may differ depending on the class of user, such as personal vs. professional use.

Thirdly, in competitive markets, the ability to segment users is limited, in theory, by the incremental cost of providing the improved quality (if a provider in a competitive market were to charge more, a competitor would underbid for the same quality step). Often, the cost of additional bandwidth is low once a network has been built, and costs for customer support and administrative overhead do not depend on usage. The ability to differentiate customer segments is strongest where customers have limited choices and where the need for quality strongly correlates with the ability to pay. Sometimes, providers also restrict

who can access low-cost, low-quality offerings, such as the low-bandwidth Internet access restricted to low-income families in the US. As another example for networks, prepay cellular packages may limit roaming, thus offering reduced coverage at a lower price point. Examples of market segmentation of applications by QoE include Netflix's UHD option, or Spotify's subscriber tiers. Some research [2] also suggests that quality-based market segmentation is possible.

On the technical side, there is a lack of standardized, well-engineered and robust measurement and monitoring tools for QoS and QoE, which are needed to successfully exploit QoE in a business context. The architecture proposed in the QoE Management working group could be a good starting point for addressing this gap.

b) Combining QoE and SLAs

QoE knowledge could be exploited in several places to facilitate SLAs:

- Facing the user (e.g., sell a consistently-guaranteed quality level);
- As an internal SLA by the service provider (e.g., use it to drive resource management so as to provide a certain quality level to the users);
- As a means of supporting SLAs in multi-provider scenarios (e.g., SaaS vendors who require guarantees from upstream providers).

Given the first issue discussed above on how to communicate the meaning of QoE to the users, the first option seems the hardest to work out currently. The second and third options could use QoE models as tools in determining which SLAs or resource management mechanisms would best suit the service provider's goals e.g., [1].

Another limiting factor related to SLAs is that in many contexts, they simply don't exist or are very poor. For example, user-facing network plans rarely have any service level guarantees (e.g., "*Up to* 100 Mbps", which in practice is never realized). Even in some commercial contexts, such as cloud hosting, SLAs can be almost non-existent (e.g., Amazon's AWS). Offering SLAs in these commercial contexts might also come with liability implications that require a careful risk analysis.

4.5.3 Discussion

a) Economic differentiators

ISPs mainly use speed as a low-level differentiator, while cellular providers compete on coverage and network technology (e.g., 4G). For enterprise users, reliability (and speed of recovery from failure), or sharing (virtual circuits vs. best effort) seem most promising.

For cloud services, there are more possibilities to differentiate quality offerings for different customer tiers, but they are – as can be expected – very service-dependent. For OTT video, for example, different quality levels and content availability could be a way to tier subscriptions.

b) Information disclosure

A common recurring topic during the discussion was that of properly communicating with the customers on quality and performance issues. The main issue related to this is about communicating what a certain level of QoE *feels like* to the users, if QoE-based differentiation is to be done at all.

The requirements for information disclosure are also different when dealing with end users, or with "edge providers" (e.g., SaaS providers), both in terms of metrics, and of language.

c) Public policy

When thinking about QoE-based differentiation, the issue of network neutrality appears very quickly in many cases. There are several examples where delivering a service with proper quality might require deals between the service and network providers involved, which could lead to neutrality issues (e.g., Netflix in the USA).

In some countries (e.g., Chile, Finland), telecom regulators have started imposing certain performance levels and/or availability of clearly defined metrics with which ISPs need to comply. This concept could be expanded upon to include QoE indicators for a group of services representative of what most users use, for example. Other regulators, such as Ofcom and the FCC, have promulgated voluntary or mandatory transparency requirements for ISPs, sometimes based on measurements ("Measuring Broadband America").

d) Non-ICT domains

An interesting parallel was drawn between the OTT services and ISPs relation and that of power generation and distribution in the Nordic countries. It would be interesting to study these in more detail, as the problems faced are remarkably similar in several aspects, and have been successfully solved in the case of power grids.

Other non-ICT domains, such as retail, might also provide interesting tools to apply to online services.

4.5.4 Research questions and open issues

- **In order to enable QoE-aware business models and SLAs, how can we convey the "meaning" of quality to the end users?** Three different levels of information disclosure were identified:

  End-users: A "QoE Emulator" would allow users to experience the different quality levels and make a decision based on them. This would allow to demonstrate how different services would be experienced at peak times. Another option is a crowd-sourced approach, where ratings of similar users under similar conditions might be used as guidelines.

  Professionals and regulators: For professionals, a MOS-like quantitative estimate may be sufficient, but may not reflect variability in time and across the service territory.

  Engineers: Multiple lower-level metrics, and if possible causal links between them and QoE, allow root cause analysis.

- **What to measure, and how to measure it?** From the ISP's point of view, most issues tend to occur in the "middle mile" which is shared by multiple users. For users, issues are often found in their home networks, which cannot be easily monitored by the providers. Measurements would then be needed in several points, including the end user's premises, the last mile, and the links between the POP and the actual services. The types and number of services to be measured in order to obtain representative results is also an open question. Once this has been solved, there remains the issue of communicating this to end-users in an understandable manner. Some ideas to deal with this could be a "food label"-like approach[1] that would put some technical aspects in terms understandable by users (e.g., 5-star ratings). An objective third party could collect the measurements

[1] Inspired by the nutritional information labels on food products.

and provide ratings for network and service providers available in a certain area, so that consumers would be able to gauge the differences (in terms of quality) between the different providers and their different plans.

- **How to identify different customer segments?** In this context, willingness-to-pay studies might be a good starting point, showing how users are willing to spend money on services at different qualities. Parallels drawn from other domains might also provide interesting insight, such as the case of Nordic countries' power grids mentioned above, for instance.

References

1 Pantelis A. Frangoudis, Aggeliki Sgora, Martín Varela, and Gerardo Rubino. Quality-driven optimal SLA selection for enterprise cloud communications. In *Proceedings of the IEEE ICC 2014 – Workshop on QoE-centric Network and Application Management (QoENAM)*, June 2014.

2 Andreas Sackl, Patrick Zwickl, and Peter Reichl. The trouble with choice: an empirical study to investigate the influence of charging strategies on content selection on QoE. In *Proceedings of the 9th International Conference on Network and Service Management (CNSM)*, Oct. 2013. DOI: 10.1109/CNSM.2013.6727850

4.6 QoE in new domains

Markus Fiedler (Blekinge Institute of Technology – Karlskrona, SE)

4.6.1 Starting points

The collection of topics and questions that were raised from the participants revealed interest in new domains for QoE. Initially, the areas of eHealth, education, and smart grids were identified, and complemented by issues regarding software, installation, billing, and security.

4.6.2 Leaving the QoE comfort zone of multimedia

The classical domain for QoE is multimedia experience with particular focus on audio and video, which is nicely reflected in the most recent, Dagstuhl-born and hedonic-centered definition of QoE as "degree of delight and annoyance" [1, 2, 3]. On the other hand, vendors, providers and operators are interested in QoE as a means of assessing and controlling the risk of user churn. Indeed, the risk of user annoyance and consequent churn nowadays exists in many ICT service domains. When users face suboptimal service performance, they may abandon the service, which makes providers, operators and in the end even vendors face unfortunate economical consequences. In particular, the introduction of new types of services has to be successful, or put the other way round, should not be "overshadowed" by QoE issues. Thus, from the viewpoints of vendors, providers and operators, QoE (and in particular its connection to churn, Service Level Agreements and business aspects) has become a focus area for both existing and emerging services. This poses the particular challenge to the QoE community of how the QoE definition, focusing on *well-being* (delight and annoyance), can be applied to any emerging kind of service, far beyond audio and video. However, purely focusing on well-being and QoE is not sufficient. It is evident that even aspects of *usability*, in particular regarding installation and usage, as well as *utility* need to be addressed in order to pave the way towards acceptance and success of new services.

4.6.3 Emerging areas

Areas showing an increased interest in QoE and related issues are amongst others:

- *Internet of Things* (IoT), with the challenge of implementing Machine-to-Machine (M2M) communication that satisfies the (human) end user and enables new Business-to-Business (B2B) relationships;
- *E-Health*, with the challenge of designing and implementing applications and services that are accepted by all stakeholders under rather complex usage and legal conditions;
- *E-Learning*, with the challenge of having services at hand that do not impede, but strongly support the learning process;
- *Smart data*, where a key value is found in the right information in the right place at the right time;
- *Serious games*, where the "gamification" of use cases (such as memory training) needs to be done in an appealing and convincing way;
- *Arts and culture*, where well-working and -perceived services open up for new channels to reach existing and new audiences.

All those areas have in common that well-being, usability and utility are key ingredients for successful service delivery.

4.6.4 Demands, net benefit, and utility

A review of demands in the different areas has shown that *time-based quality and consistency issues* dominate. One of the delegates pointed out that if one cannot provide the service in a consistent manner, one will face issues with experience and utility. Indeed, the approach in [4] introduces the *net benefit* as principal determinant of user behavior and – through user satisfaction – customer behavior. Net benefit is the result of *gross benefit* – impacted by fulfillment of needs and service quality – and *sacrifice* – impacted by service quality (again!), value of time and usage price. The ambivalent role of service quality as both enabler (if good) and disabler (if bad) can also be seen in utility values, which can be positive (gains) or negative (losses). Obviously, time plays a key role as sacrifice to be paid by the user (e.g. in the form of boredom or lost opportunities). Thus, waiting time issues translate into negative utility issues, which is counterproductive to the demand of positive utility for the successful implementation of new services, and these waiting times have to be modeled and evaluated. Related concepts from the area of Software Engineering, combining time, error, effort and usage, can contribute to this task.

4.6.5 Research questions and open issues

Given the rapid development of new fields with the need of QoE, usability and utility considerations, the following research questions emerged:

- Once a new field appears, how to develop a quality concept for that?
- How should QoE in those new domains be assessed?

There is an obvious need to deepen these issues, potentially in the form of another QoE-related Dagstuhl Seminar.

References

1 Dagstuhl Seminar 09192. http://www.dagstuhl.de/09192
2 COST Action IC 1003 QualiNet. http://www.qualinet.eu
3 Dagstuhl Seminar 12181. http://www.dagstuhl.de/12181
4 Kalevi Kilkki. *An Introduction to Communications Ecosystems*. 2012. http://kilkki.net

5 Concluding remarks

Katrien De Moor (NTNU – Trondheim, NO)

The main objectives of this third Dagstuhl seminar on Quality of Experience were to strengthen and go beyond the current understanding on QoE, and to push the transition from the assessment of QoE to the practical application of QoE knowledge and mechanisms. Keeping these objectives in mind and referring back to the detailed summaries from the different working groups above, we can conclude that significant steps forward were made. The fruitful discussions amongst the participants resulted in the identification of both points of crystallization concerning our current understanding of QoE and open questions and challenges for future joint endeavors.

The joint work on the fundamentals of QoE ("QoE theory and modeling") resulted in a comprehensive QoE and user behavior model, providing a high-level integrative framework which was to date still lacking. The discussions on "QoE methodologies" were centered around a number of caveats and challenges related to the traditional methodological approaches in QoE research and more specifically, on concrete proposals on how these could be better addressed in order to increase the ecological validity of QoE research. Similarly, in the group work on "User factors", and based on the limitations of current approaches, the discussion resulted in concrete suggestions on how to better take user factors into account (in terms of measuring and capturing user factors, incorporating these factors into QoE analyzes and exploiting the related insights in the application of QoE principles and insights) and in the identification of open questions that can steer future joint work on this matter.

Complementing and going beyond the research perspective, the fourth working group identified challenges related to "QoE Management". As a concrete and important step forward, a high-level and QoE-enhanced architecture was developed. In addition, a set of related research questions and requirements, including the need for a new design-focused approach for QoE, were discussed and put on the agenda. The business perspective and challenge of turning QoE-related knowledge into economic value, were the main focus of the discussions in the group on the "Monetization of QoE". Barriers to the development and implementation of QoE-aware business models and SLAs, as well as key issues that need to be tackled in order to overcome these barriers (such as e.g., the identification of different customer segments), were identified and discussed. Finally, the working group on "QoE in new domains" explored the relevance and application of QoE for fields that go beyond the "safe and familiar" multimedia domain. The concrete domains that were considered in the discussion include Internet of Things, E-Health, E-Learning, Smart data, Serious games and Arts and Culture. Each of these new domains brings along distinct challenges for research on QoE, which need to be further investigated and deepened in the future. By putting

these domains explicitly on the future QoE-agenda, we believe that the seminar may play a triggering and accelerating role in this respect.

During the wrap-up session on the last day, which featured the high-level summaries from the different working groups, the seminar in itself was also briefly evaluated. More specifically, every participant was asked to make a short statement about the seminar and its organization. From this short feedback session, it became clear that the group work, and the fact that the time for individual presentations was limited, was in general very much appreciated by the participants. However, it was also commonly repeated that there was perhaps not enough time for "digestion", reflection and further discussion due to the tight schedule and limited time (2.5 days). Similarly, it was argued that there should also be more time to report back to the whole group after the break-out discussions, and more time to take up some of the discussed issues further in plenum.

Altogether, we can look back at a successful and inspiring seminar, which triggered lively discussions and cross-disciplinary exchange, and which resulted in plans for joint follow-up activities in several of the discussion groups. 2.5 months after the seminar, a number of concrete outcomes can already be included in the report. For example, one paper has already been accepted for presentation at the IEEE ICC 2015-workshop on "Quality of Experience-based Management for Future Internet Applications and Services" [4], and three other joint papers that either directly originate from the discussions during the seminar, or have been inspired by them, have been accepted for presentation at the 7th International Workshop on Quality of Multimedia Experience (QoMEX 2015)[1, 2, 3]. In addition, many open issues, which will require deeper discussions (e.g., at potential future Dagstuhl seminars and at other venues), and which may lead to future collaboration between the participants, have been put on the agenda. In line with the previous Dagstuhl seminars on QoE, we are therefore confident that the Dagstuhl QoE community has been strengthened by the seminar and that several of the ideas discussed during this seminar will find their way into the literature, or become visible in another way in the future.

References

1 Jan-Niklas Antons, Sebastian Arndt, Katrien De Moor and Steffen Zander. Impact of perceived quality and other influencing factors on emotional video experience. In *Proceedings of the 7th International Workshop on Quality of Multimedia Experience (QoMEX)*, May 2015.

2 Tobias Hoßfeld, Poul E. Heegaard and Martín Varela. QoE beyond the MOS: Added value using quantiles and distributions. In *Proceedings of the 7th International Workshop on Quality of Multimedia Experience (QoMEX)*, May 2015.

3 Peter Reichl, Sebastian Egger, Sebastian Möller, Kalevi Kilkki, Markus Fiedler, Tobias Hoßfeld, Christos Tsiaras and Alemnew Sheferaw Asrese. Towards a comprehensive framework for QoE and user behavior modelling. In *Proceedings of the 7th International Workshop on Quality of Multimedia Experience (QoMEX)*, May 2015.

4 Martín Varela, Patrick Zwickl, Peter Reichl, Min Xie, and Henning Schulzrinne. Experience Level Agreements (ELA): the challenges of selling QoE to the user. In *Proceedings of the IEEE ICC 2015 – Workshop on Quality of Experience-based Management for Future Internet Applications and Services (QoE-FI)*, June 2015.

Participants

- Jan-Niklas Antons
TU Berlin – Berlin, DE
- Alemnew Asrese
Aalto University – Espoo, FI
- Katrien De Moor
NTNU – Trondheim, NO
- Philip Eardley
British Telecom R&D – Ipswich, GB
- Sebastian Egger
AIT – Wien, AT
- Markus Fiedler
Blekinge Institute of Technology – Karlskrona, SE
- Farnaz Fotrousi
Blekinge Institute of Technology – Karlskrona, SE
- Pantelis Frangoudis
INRIA Rennes – Bretagne Atlantique, FR
- Samuel Fricker
Blekinge Institute of Technology – Karlskrona, SE
- Juan Pablo González Rivero
Plan Ceibal – Montevideo, UY
- Poul Einar Heegaard
NTNU – Trondheim, NO
- Tobias Hoßfeld
Universität Duisburg-Essen – Essen, DE
- Kalevi Kilkki
Aalto University – Espoo, FI
- Eirini Liotou
National Kapodistrian University of Athens – Athens, GR
- Toni Mäki
VTT Technical Research Centre of Finland – Oulu, FI
- Sebastian Möller
TU Berlin – Berlin, DE
- Peter Reichl
Universität Wien – Wien, AT
- Miguel Ríos Quintero
TU Berlin – Berlin, DE
- Werner Robitza
TU Berlin – Berlin, DE
- Henning Schulzrinne
Columbia Univ. – New York, US
- Lea Skorin-Kapov
Univ. of Zagreb – Zagreb, HR
- Samira Tavakoli
Universidad Politécnica de Madrid – Madrid, ES
- Christos Tsiaras
Universität Zürich – Zürich, CH
- Astrid Undheim
Telenor Research and Development – Trondheim, NO
- Martín Varela
VTT Technical Research Centre of Finland – Oulu, FI
- Min Xie
Telenor Research – Fornebu, NO
- Patrick Zwickl
Universität Wien – Wien, AT

Report from Dagstuhl Seminar 15031

Understanding Complexity in Multiobjective Optimization

Edited by

Salvatore Greco[1], Kathrin Klamroth[2], Joshua D. Knowles[3], and Günter Rudolph[4]

1 **Università di Catania, IT,** salgreco@unict.it
2 **Bergische Universität Wuppertal, DE,** klamroth@math.uni-wuppertal.de
3 **University of Manchester, GB,** j.knowles@manchester.ac.uk
4 **TU Dortmund, DE,** guenter.rudolph@tu-dortmund.de

Abstract

This report documents the program and outcomes of the Dagstuhl Seminar 15031 Understanding Complexity in Multiobjective Optimization. This seminar carried on the series of four previous Dagstuhl Seminars (04461, 06501, 09041 and 12041) that were focused on Multiobjective Optimization, and strengthening the links between the Evolutionary Multiobjective Optimization (EMO) and Multiple Criteria Decision Making (MCDM) communities. The purpose of the seminar was to bring together researchers from the two communities to take part in a wide-ranging discussion about the different sources and impacts of complexity in multiobjective optimization. The outcome was a clarified viewpoint of complexity in the various facets of multiobjective optimization, leading to several research initiatives with innovative approaches for coping with complexity.

Seminar January 11–16, 2015 – http://www.dagstuhl.de/15031
1998 ACM Subject Classification G.1.6 Optimization, H.4.2 Types of Systems, I.2.6 Learning, I.2.8 Problem Solving, Control Methods, and Search, I.5.1 Models
Keywords and phrases multiple criteria decision making, evolutionary multiobjective optimization
Digital Object Identifier 10.4230/DagRep.5.1.96
Edited in cooperation with Richard Allmendinger

1 Executive Summary

Salvatore Greco
Kathrin Klamroth
Joshua D. Knowles
Günter Rudolph

Understanding complexity in multiobjective optimization is of central importance for the two communities, MCDM and EMO, and several related disciplines. It enables us to wield existing methodologies with greater knowledge, control and effect, and should, more importantly, provide the foundations and impetus for the development of new, principled methods, in this area.

We believe that a strong route to further progress in multiobjective optimization is a determination to understand more about the various ways that complexity manifests itself in multiobjective optimization. We observe that in several fields, ranging from engineering to medicine to economics to homeland security, real-world problems are very often characterized

by a high degree of complexity deriving from the presence of many competitive objectives to be optimized, many stakeholders expressing conflicting interests and the presence of many technical parameters being unstable in time and for which we have imperfect knowledge. These very complex problems require a specific methodology, mainly based on multiobjective optimization, that, using high computational capacities, takes into account robustness concerns and allows an effective participation of the several stakeholders in the decision process.

The seminar took place January 11th–16th 2015. The main goals of the seminar were the exploration and elucidation of complexity in three fundamental domains:

Focus 1: Complexity in preference

This topic is mainly concerned with elicitation, representation and exploitation of the preference of one or more users, for example: discovering and building preferences that are dynamic and unstable, group preference, complex structure of criteria,non-standard preferences, learning in multiobjective optimization.

Focus 2: Complexity in optimization

This topic is mainly concerned with the generation of alternative candidate solutions, given some set of objective functions and feasible space. The following topics are examples for the wide range of issues in this context: high-dimensional problems, complex optimization problems, simulation-based optimization and expensive functions, uncertainty and robustness, interrelating decision and objective space information.

Focus 3: Complexity in applications

An all-embracing goal is to achieve a better understanding of complexity in practical problems. Many fields in the Social Sciences, Economics, Engineering Sciences are relevant: E-government, Finance, Environmental Assessment, E-commerce, Public Policy Evaluation, Risk Management and Security issues are among the possible application areas.

During the seminar the program was updated on a daily basis to maintain flexibility in balancing time slots for talks, discussions, and working groups. The working groups were established on the first day in highly interactive fashion: at first each participant was requested to write her/his favorite topic on the black board, before a kind of collaborative clustering process was applied for forming the initial five working groups, some of them splitting into subgroups later. Participants were allowed to change working groups during the week, but the teams remained fairly stable throughout. Abstracts of the talks and extended abstracts of the working groups can be found in subsequent chapters of this report.

Further notable events during the week included: (i) a session devoted to discuss the results and the perspectives of this series of seminars after ten years of the first one, (ii) a hike within a time slot with worst weather conditions during the week, (iii) a presentation session allowing us to share details of upcoming events in our research community, and (iv) a wine and cheese party made possible by a donation of UCL's *EPSRC Centre for Innovative Manufacturing in Emergent Macromolecular Therapies* represented by Richard Allmendinger.

Outcomes

The outcomes of each of the working groups can be seen in the sequel. Extended versions of their findings will be submitted to a Special Issue on "Understanding Complexity in Multiobjective Optimization" in the *Journal of Multi-Criteria Decision Analysis* guest-edited by the organizers of this Dagstuhl seminar.

This seminar resulted in a very insightful, productive and enjoyable week. It has already led to first new results and formed new cooperation, research teams and topics. In general, the relations between the EMO and MCDM community were further strengthened after this seminar and we can expect that thanks to the seminar a greater and greater interaction will be developed in the next few years.

Acknowledgements. Many thanks to the Dagstuhl office and its helpful and patient staff; huge thanks to the organizers of the previous seminars in the series for setting us up for success; and thanks to all the participants, who worked hard and were amiable company all week. In the appendix, we also give special thanks to Salvatore Greco as he steps down from the organizer role.

2 Table of Contents

3 Overview of Talks

3.1 Preference learning in EMO: Complexity of preference models

Jürgen Branke (University of Warwick, GB), Salvatore Corrente (Università di Catania, IT), Salvatore Greco (Università di Catania, IT), Roman Słowiński (Poznan University of Technology, PL), and Piotr Zielniewicz (Poznan University of Technology, PL)

Joint work of Branke, Jürgen; Corrente, Salvatore; Greco, Salvatore; Słowiński, Roman; Zielniewicz, Piotr
Main reference J. Branke, S. Corrente, S. Greco, R. Słowiński, P. Zielniewicz, "Using Choquet Integral as Preference Model in Interactive Evolutionary Multiobjective Optimization," WBS Working Papers, Warwick Business School, 2014.
URL http://wrap.warwick.ac.uk/64234/

When learning user preferences from user interactions, one usually has to make a decision on the nature of the preference model to be learned. There is a trade-off: if the preference model is too simplistic (say, linear), it is unlikely to be able to represent perfectly the user's preferences expressed in interactions. On the other hand, if the preference model is too versatile, a lot of preference information is required from the user to narrow down the model's parameters to a useful degree, i.e., such that the preference relation implied by the model is sufficiently richer than the dominance relation. In this talk, we will survey the literature on preference learning in EMO with a special focus on the complexity of the preference model used. We will then move on to some of our recent work where the complexity of the preference model is increased adaptively.

3.2 Computational Complexity in Multi-objective (Combinatorial) Optimisation

Matthias Ehrgott (Lancaster University, GB)

In combinatorial optimisation, the study of worst case complexity of problems is very important. Researchers have also considered the worst case complexity of multi-objective versions of combinatorial optimisation problems. Most results are thoroughly unexciting and negative: MOCO problems have been shown to be NP-hard (their decision versions), #P-hard (the related counting versions, and to have exponentially many efficient solutions and non-dominated points. This is true for multi-objective versions of very easy polynomially solvable, even trivial, single objective combinatorial optimisation problems. This begs the question whether worst case complexity in the standard sense is the right framework for discussing complexity of MOCO problems. On the other hand, recent results in multi-objective linear programming show the theoretical (but far from practical) polynomial solvability of MOLP, and the possibility of computing non-dominated extreme points in MOLP with polynomial delay. Is polynomial delay a better framework and the best one can hope for? Maybe for specific instances? Is there any hope for any polynomiality results? This brief presentation is intended to encourage discussion of these issues which have been largely ignored in multi-objective optimisation to date.

3.3 Variable ordering structures – what can be assumed?

Gabriele Eichfelder (TU Ilmenau, DE)

Main reference G. Eichfelder, "Variable Ordering Structures in Vector Optimization," Springer, 2014.
URL http://www.springer.com/mathematics/book/978-3-642-54282-4

In some real-world applications in multi-objective optimization it cannot be assumed that there is a partial ordering in the image space, i.e. that there exists a binary relation which is reflexive, transitive and compatible with the linear structure of the space. Instead, preferences may vary depending on the current information. This can be modeled by an ordering map which associates sets of improving (or deteriorating) directions with each element of the image space or of the pre-image space. Depending on the point of view (i.e. preference or domination) different optimality concepts are discussed in the literature. In this talk we give some motivating applications and a basic introduction to this topic. We present the various ways given in the literature to model a variable ordering structure and the different optimality concepts which are derived. We collect some basic properties which are often assumed for obtaining theoretical and numerical results. Limitations of the current concepts are also pointed out. This talk aims to be the base for a discussion on how variable ordering structures can be modeled, which assumptions on an ordering map seem to be reasonable, and which optimality concepts are considered to be most practically relevant.

3.4 An Open Problems Project for Set Oriented and Indicator-Based Multicriteria Optimization

Michael Emmerich (Leiden University, NL)

Main reference SIMCO – Open problems webpage
URL http://simco.gforge.inria.fr/doku.php?id=openproblems

In September 2013 the 'Set-oriented and Indicator Based Multicriteria Optimization Open Problems Project' (SIMCO-OPP) was launched during the Lorentz Center Workshop on Multicriteria Optimization in Leiden University, The Netherlands, in order to collect exact results on algorithms and open questions in indicator based multi-criteria optimization. The SIMCO-OP Project maintains a collection of registered positive (exact) results and questions related to problems such as multi-criteria sorting and searching, computation of multi-criteria performance indicators, gradient computations, convergence times of problem-algorithm pairs, and optimal subset computation problems. Computational complexity results are a major theme and state-of-the-art results for the known computational complexity bounds for a large number of problems are maintained.

In our talk, which is related to the topic 'complexity in optimization', an overview of the SIMCO-OP Project will be given, including a brief introduction to its scope and the structure of result records in the repository. The aim is to invite participants to use the repository and to contribute to it by, for instance, registering new published results that they find or that come to their attention. The presentation will also highlight selected open problems on computational complexity of algorithms in multicriteria optimization.

3.5 Pareto-front approximation statistics

Carlos M. Fonseca (University of Coimbra, PT)

Joint work of Fonseca, Carlos M.; Grunert da Fonseca, Viviane; Guerreiro, Andreia P.; López-Ibáñez, Manuel; Paquete, Luis

In this talk, the variation of Pareto-front approximations across multiple multiobjective optimisation runs is considered from a statistical point of view. The attainment function methodology [1] is briefly described as a means of capturing important aspects of algorithm behaviour, such as location, variability, and dependence, through the estimation of the moments of the set-distribution of the corresponding outcome approximation sets. Complexity issues [2] concerning the computation, visualisation, and size of the moment estimates, as the number of objectives, number of runs, and size of the approximations grow are highlighted.

Acknowledgments. This work was partially supported by iCIS (CENTRO-07-ST24-FEDER-002003).

References

1 V. Grunert da Fonseca and C. M. Fonseca. The attainment-function approach to stochastic multiobjective optimizer assessment and comparison. In *Experimental Methods for the Analysis of Optimization Algorithms* (T. Bartz-Beielstein, M. Chiarandini, L. Paquete, M. Preuss, eds.), ch. 5, pp. 103–130, Springer, 2010.

2 C. M. Fonseca, A. P. Guerreiro, M. López-Ibáñez, and L. Paquete. On the computation of the empirical attainment function. In *Evolutionary Multi-Criterion Optimization. 6th Int'l Conf., EMO 2011* (R. H. C. Takahashi, K. Deb, E. F. Wanner, and S. Greco, eds.), vol. 6576 of *LNCS*, pp. 106–120, Springer, 2011.

3.6 Complex combinatorial problems with heterogeneous objectives

Andrzej Jaszkiewicz (Poznan University of Technology, PL)

An important source of complexity in multiobjective combinatorial optimization that is often overlooked are heterogeneous objectives. By heterogeneous we understand objectives that differ from the point of view optimization, i.e. the tasks of single objective optimization of particular objectives differ significantly.

The objectives may differ by difficulty in optimization. They may of different difficulty from computational complexity point of view, e.g. optimization of some objectives may correspond to simple problem while optimization of other objectives may correspond to NP-hard problems. Probably even more common are differences in practical difficulty, i.e. optimization of some objectives may require much more or less steps of an evolutionary or more generally metaheuristic algorithm. Differences in difficulty may result from different mathematical form of the objective functions, but even if the mathematical form is the same, different objectives may correspond to instances of different difficulty. For example for classical CO problems like TSP or set covering it is well known that various classes of instances, like Euclidean, random, clustered, are of different difficulty.

Another, even more complex aspect, of heterogeneous objectives is that they may require different optimization algorithms, or different operators used in the algorithms, to get very

good results. For example, very different recombination operators may perform best for particular objectives.

Quick literature review shows that most theoretical papers focus on problems with homogeneous objectives, while most papers about practical applications of MOCO describe problems with heterogeneous objectives. Thus, naturally existing algorithms are well adapted to homogeneous objectives only.

3.7 Bridging the Gap between Theory and Application in Evolutionary Multi-Objective Optimization

Yaochu Jin (University of Surrey, GB)

This talk aims to bridge the gap between the current hot topics in evolutionary multi-objective optimization (MOO) and the urgent demands from real-world optimization. We will show that, while solving multi-objective optimization problems (MOPs) with a large number of objectives, often known as many-objective optimization, MOPs having a high-dimensional decision space (large-scale optimization) and MOPs having a very complex Pareto front have been very popular in academia, industry is more concerned with complexity in formulating the optimization problems, choosing the right decision variables, defining the most important objectives, and handling different constraints in the conceptual, design and verification phases. We will also point out that some assumptions in the present many-objective optimization and dynamic optimization research are unrealistic; the results are practically of little value, or even misleading. In addition, handling uncertainty and reducing the computational complexity in evaluating the quality of the designs are extremely important in dealing with real-world problems. As a result, incorporating a priori knowledge in various forms will be critical for handling the time constraint and performance requirement in real-world optimization. In the presentation, several application examples from industry, including design of vehicles, natural gas terminals and steel-making processes will be used to illustrate the real-world challenges in multi-objective optimization.

3.8 Machine Decision Makers: From Modeling Preferences to Modeling Decision Makers

Manuel López-Ibáñez (Free University of Brussels, BE)

Joint work of López-Ibáñez, Manuel; Knowles, Joshua D.
Main reference M. López-Ibáñez, J. D. Knowles, "Machine decision makers as a laboratory for interactive EMO," in Proc. of the 8th Int'l Conf. on Evolutionary Multi-Criterion Optimization (EMO'15), LNCS, Vol. 9019, pp. 295–309, Springer, 2015; pre-print available as IRIDIA Technical Report No. TR/IRIDIA/2014-016.
URL http://dx.doi.org/10.1007/978-3-319-15892-1_20
URL http://iridia.ulb.ac.be/IridiaTrSeries/link/IridiaTr2014-016.pdf

Quantitative assessment of any method involving human decision-makers (DMs) is difficult due to the need for a DM from whom preference information is elicited.Not only it is complex to characterize the properties of a human DM, but the DMs considered during experimentation may not have the same characteristics nor the same motivation than the ones for which the method is ultimately designed. Most studies that simulate DMs typically consider them

as no more than a utility function. In a few cases, noise is added to this utility function to simulate human mistakes. Such a simple model cannot hope to capture the complexity of human psychological biases. At the same time, it neglects the existence of common characteristics in human DMs that are independent of a particular preference. The existence of such commonalities is more evident when considering DMs within a particular application scenario that may be, for example, risk-averse, risk-seeking or exploratory. Nonetheless, there are several works in the literature that have tried to simulate realistic DMs. In particular, T. J. Stewart proposed simulation models of various cognitive biases and factors that deviate from the ideal model ("non-idealities"), and studied their effect on multi-criteria decision-making (MCDM) methods such as goal programming, aspiration-based methods and additive value functions. Recently, López-Ibáñez and Knowles have applied this simulation model to evaluate an interactive evolutionary multi-objective optimization (EMO)algorithm. This talk discusses the concept of a machine decision-maker as are-usable, parametric, and general model of a realistic DM that can be used to analyze the effect of human factors and other non-idealities on interactive MCDM/EMO algorithms. The ultimate goal is that machine DMs would motivate the development of methods that are able to cope with various human cognitive biases and other non-idealities. Moreover, given enough data about past human interactions, it could be possible to learn the parameters of machine DMs in order to adapt them to particular application scenarios. Theories and results from psychology of judgment and decision-making, behavioral economics, and cognitive science should guide the construction of machine DMs. Nevertheless, there are still many open research questions on how to build, configure and use machine DMs in the context of interactive MCDM/EMO algorithms.

3.9 Sources of Computational Challenges in Multiobjective Optimization

Kaisa Miettinen (University of Jyvaskyla, FI)

Joint work of Luque, Mariano; Ruiz, Francisco; Ruuska, Sauli; Sindhya, Karthik; Steponavice, Ingrida

In this talk we, discuss some reasons why multiobjective optimization problems may be computationally expensive and challenging to solve. We also propose some ways of tackling the challenges. The main focus is on simulation-based optimization. It is important to keep in mind that reliable models are required for optimization but, on the other hand, optimization enables taking full advantage of high-quality models. A high accuracy easily introduces a high computational cost and this implies a need for balancing between accuracy and cost.

In simulation-based optimization we need different tools for handling complexity. When objective and constraint functions depend on the output of simulation models, function evaluations may be time-consuming, which introduces computational cost as a challenge. And as real life problems typically involve several conflicting objectives to be optimized simultaneously, methodological support for decision makers is important in identifying the most preferred solution. This necessitates preference information from the decision maker. Simulation models may have a black box nature and, thus, properties of functions involved may be unknown. For example, global optimization is needed when the convexity of the problem cannot be assumed and this typically increases the computational cost. Finally, stochasticity may have to be taken into account, for example, because the output of simulation

models may be random vectors with unknown distributions. Handling this requires sampling the output which increases the computational cost.

We outline a three-stage approach which has been proposed in [1] for solving computationally challenging multiobjective optimization problems involving black-box models and stochasticity. In the pre-decision making stage, a set of Pareto optimal solutions is generated based on which a computationally less expensive surrogate problem is formed. The decision maker can solve the surrogate problem in the decision making stage with an interactive method because of low computing times. Finally, in the post-decision making stage, the final solution of the surrogate problem is projected to the Pareto optimal set of the original problem. The three-stage approach is applied in [1] when solving a joint design and operation problem of a paper plant. Here, the PAINT method [2] is used to generate the surrogate problem and the decision maker solves the problem with the interactive NIMBUS method [3].

We also discuss further method development challenges including high dimensions in decision and objective spaces, need of robustness, different forms of uncertainty, multilevel problems, user interface design and the importance of usability, the added value offered by different disciplines like visual analytics, new devices and platforms enabling a better utilization of the strengths of humans and computers and the potential of hybrid methods where elements of different types of methods are combined.

We conclude by outlining the interactive E-NAUTILUS method [4] for computationally expensive problems, which combines the three-stage approach and the philosophy of the NAUTILUS method [5]. In NAUTILUS, solutions of consecutive iterations improve all objectives and, thus, only the final solution is Pareto optimal. In this way, the decision maker can make a free search for the most preferred solution without e.g. anchoring.

References

1 I. Steponavice, S. Ruuska, and K. Miettinen. A Solution Process for Simulation-based Multiobjective Design Optimization with an Application in Paper Industry. *Computer-Aided Design*, 47:45–58, 2014.

2 M. Hartikainen, K. Miettinen, and M. M. Wiecek. PAINT: Pareto Front Interpolation for Nonlinear Multiobjective Optimization. *Computational Optimization and Applications*, 52(3):845–867, 2012.

3 K. Miettinen and M. M. Makela. Synchronous Approach in Interactive Multiobjective Optimization. *European Journal of Operational Research*, 170(3):909–922, 2006.

4 A. B. Ruiz, K. Sindhya, K. Miettinen, F. Ruiz, and M. Luque. E-NAUTILUS: A Decision Support System for Complex Multiobjective Optimization Problems based on the NAUTILUS Method. Submitted.

5 K. Miettinen, P. Eskelinen, F. Ruiz, and M. Luque. NAUTILUS Method: An Interactive Technique in Multiobjective Optimization based on the Nadir Point. *European Journal of Operational Research*, 206(2):426–434, 2010.

3.10 Understanding and managing complexity in real-case applications

Silvia Poles (Noesis Solutions, BE)

This presentation will be divided into two parts. In the first part we list all the possible sources of complexity in real-case applications and we analyze how these sources can affect the achievement of a solution in terms of time and effort.

Among other sources of complexity, we can mention the difficulty in integrating external solvers (e.g. simulation software) or the evaluation of time consuming functions (e.g. CFD codes) in the optimization process. Another difficulty can be a limited number of possible function evaluations, this limit is very common when dealing with time consuming functions. As other sources of complexity we can have a highly dimensional problem, a highly constrained problem, an optimization problem many conflicting objectives, or a problem with highly non-linear responses.

In the second part of the presentation we discuss about proposed solutions for managing complexity in real-case applications. For example, we show the use of a "process integration for design optimization" (PIDO) tool such as Optimus for the easy integration of different external solvers into a single platform. We will demonstrate the use of design of experiment approaches for reducing the problem dimension, and the use of models (meta-models) for reducing the number of evaluations of time consuming function. Eventually, the use of hybrid algorithms or a task list of methods will be proposed as an approach for highly non-linear problems. In this second part, all the proposed solutions will be supported by real-case multiobjective optimization problems.

3.11 Perspectives on the application of multi-objective optimization within complex engineering design environments

Robin Purshouse (University of Sheffield, GB)

Joint work of Purshouse, Robin; Giagkiozis, Ioannis; Fleming, Peter

Multi-objective optimization has experienced significant growth as a research field over the last few decades. However there exist very few published examples where multi-objective optimization methods have been used within a real decision-making context for engineering products or services. Whilst this dearth of evidence may be due to disincentives surrounding publication, it may also support a hypothesis that formal methods for multi-objective optimization are incongruent with *in situ* organisational practices and, as a result, are simply not used. This presentation will review the existing optimization frameworks that have attempted to account for the complexity in engineering design environments. Most of these frameworks have arisen in the field of multi-disciplinary design optimization (MDO), and include architectures such as collaborative optimization and analytical target cascading. The presentation will also highlight some of the key challenges as yet unaddressed by the MDO community; specifically: (1) how to handle the asynchronous distributed nature of the engineering design environment to ensure right-first-time design; (2) how to allocate resources to compromise-seeking activities in an environment of shared design variables and conflicting product requirements.

3.12 Advancing Many-Objective Robust Decision Making Given Deep Uncertainty

Patrick M. Reed (Cornell University, US)

Main reference J. D. Herman, P. M. Reed, H. B. Zeff, G. W. Characklis, "How Should Robustness Be Defined for Water Systems Planning under Change?" Journal of Water Resources Planning and Management, 2015.
URL http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000509

This talk will demonstrate the many-objective robust decision making (MORDM)framework on a severely challenging real-world application where we are working to facilitate improved coordination across four independent U.S. cities seeking to maintain their region's supply reliability and financial stability given increasingly severe droughts. MORDM combines massively parallel many objective evolutionary optimization under uncertainty (benchmarked on more than 500,000compute cores) with recent decision theoretic work in the area of robust decision making (RDM). MORDM makes extensive use of interactive visual analytics, to facilitate negotiated group decisions and to provide insights on key system uncertainties. In contexts such as urban water supply planning, nontrivial conceptual as well as computational challenge due to the structural uncertainties associated with defining complex management problems (e.g., choosing objectives, management decisions, planning horizons, representations of preferences, etc.) as well as the challenges associated with predicting the impacts of actions (e.g., imperfect knowledge of system dynamics, external forcings, or environmental thresholds). Often in complex infrastructure systems, modelled processes are impacted by deep (or Knightian) uncertainties. Deep uncertainties emerge when planners are unable to agree on or identify the full scope of possible future events including their associated probability distributions. RDM is used in the second stage of this framework to determine the robustness of tradeoff alternatives to deeply uncertain future conditions and facilitates decision makers' selection of promising candidate solutions. MORDM tests each solution under the ensemble of future extreme states of the world (SOWs). Global sensitivity methods are used to identify what assumptions and system conditions pose many-objective performance vulnerabilities if candidate Pareto approximate alternatives are selected.

3.13 Tutorial on Large-Scale Multicriteria Portfolio Selection Leading Up to Difficulties Obstructing Further Progress

Ralph E. Steuer (University of Georgia, US)

Joint work of Hirschberger, Markus; Steuer, Ralph E.; Utz, Sebastian; Wimmer, Maximilian; Qi, Yue
Main reference M. Hirschberger, R. E. Steuer, S. Utz, M. Wimmer, Y. Qi, "Computing the Nondominated Surface in Tri-Criterion Portfolio Selection," Operations Research, 61(1):169–183, 2013.
URL http://dx.doi.org/10.1287/opre.1120.1140

This is oriented toward algorithms for solving large-scale multicriteria portfolio selection problems and visualization methods for conveying the nondominated set. The basic formulation here is a multiple objective linear program other than for one or more of the objectives being quadratic. It is necessary to compute the entire nondominated set because in portfolio selection users are usually not able to select an optimal solution until after seeing that everything else is worse. There are two basic ways of solving for the nondominated set. One

is to compute exactly the nondominated set by means of a parametric quadratic programming procedure. The other is to compute a dotted representation of the nondominated set via an e-constraint strategy, but because of covariance matrix difficulties this is not always as easy as it may seem.The simplest portfolio case is a QL problem: two objectives (one quadratic, one linear). In this case the nondominated set graphs as a frontier consisting of a connected collection of curved arcs each coming from a different parabola. With additional objectives, the nondominated set becomes a surface consisting of a connected collection of curved patches each coming from a different paraboloid. While we can also compute the patches of QLL and QLLL problems, problem size drops dramatically. Now we are beginning to see QQL and QQLL problems proposed, but no one knows how to deal with them yet. Whereas it is a struggle to graph the nondominated surfaces of tri-criterion portfolio selection problems, how to employ visualization techniques with the nondominated sets of problems with four criteria is a major challenge. Then there are other types of difficult problems including the cardinality constrained portfolio selection problem.

3.14 Distributed MCDM under partial information

Margaret M. Wiecek (Clemson University, US)

Main reference B. Dandurand, M. M. Wiecek, "Distributed computation of Pareto sets," to appear.

Technological advances and globalization of the world create a need for multiobjective optimization-based decision making for large-scale systems. Such systems are characterized by a number of subsystems and various science or engineering disciplines that demand for specific expertises and multiple teams working in different geographical locations. Subsystems and disciplines are involved in the decision making process as interconnected elements in the physical as well as conceptual sense. The participating teams do not have access to the optimization subproblems of the other teams but may exchange limited information about their own current decisions. Because information flow between the subproblems is limited and requires periodic updating, direct solution approaches are available only at the subproblem level, and not at the level of the entire system.For example, in a large international corporation decisions are made under multiple objectives locally in each country so that the entire corporation performs globally at its best. In a military environment, multiteam planning takes place and multiple missions are executed under partial information due to constraints in the communication bandwidth or due to required communication latencies. Complex engineering design problems involve a system-level design problem and component-level design subproblems that correspond to different design-team organizational structures and require disparate solution methodologies and software interfaces. This decision making scenario requires the development of mathematical models and distributed solution methodologies that are able to capture the presence of different interconnected entities making decisions for different subsystems based on the criteria originated in multiple disciplines.The state-of-art analyses for distributed solution approaches such as the alternating direction method of multipliers (ADMM) and the block coordinate descent (BCD) method had been developed in the context of problem decomposition originating in a single objective setting and are not immediately applicable to multiobjective programs (MOPs). Applying certain scalarization techniques well-suited for nonconvex MOPs, the decomposable MOP is reformulated into a single objective problem (SOP) but the decomposability is not preserved

and the SOP is not suitable for the application of ADMM. Furthermore, coupling between the subproblems makes BCD in its current form likewise inadequate. To address these challenges to distributed multiobjective optimization, existing theory is extended for 1) iterative augmented Lagrangian coordination techniques and 2) the block coordinate descent method. Based on this study, a Multi-Objective Decomposition Algorithm (MODA) is developed for the distributed generation of efficient solutions to nonconvex decomposable MOPs. MODA is applied to a bilevel automotive design problem that is formulated as a collection of two subproblems including a vehicle-level subproblem and component-level subproblem. Numerical results of the implementation are presented showing the MODA capability of exploring the tradeoffs generated by the multiple criteria at each level.

4 Working Groups (WGs)

4.1 Modeling Behavior-Realistic Artificial Decision-Makers to Test Preference-Based Multiple Objective Optimization Methods (WG1)

Jürgen Branke, Salvatore Corrente, Salvatore Greco, Miłosz Kadziński, Manuel López-Ibáñez, Vincent Mousseau, Mauro Munerato, and Roman Słowiński

4.1.1 Introduction

Traditionally, in Multiple Objective Optimization (MOO), two separate methodological streams have been developed: evolutionary and interactive ones [2]. On the one hand, the role of Evolutionary MOO (EMO) is to approximate the entire Pareto front. On the other hand, Interactive MOO (IMO) deals with identification of the most preferred solution. IMO techniques require participation of a Decision Maker (DM) who is expected to provide her subjective preference information. The recent trend in MOO consists in merging the interactive and evolutionary approaches (for reviews, see [2, 8, 3]). This is achieved by integrating preference information into the EMO algorithms already during their optimization runs. The appealing effect of such integration consist in focusing the search on the area of the Pareto front which is most suitable to the DM.

Whenever DM preferences are used for guiding the search in MOO methods, the theoretical analysis [4] and experimental assessment of such algorithms is challenging, because it requires setting up a test environment that includes a model of the DM's behavior. Traditionally, artificial DMs have been simulated as a pre-defined value (utility) function for decision making. For example, the two user's functions used in an experimental setting in [3] assumed either linear weighting or a Tchebycheff-like aggregation of the objectives. In some other works, uncertainty of the DM interacting with an algorithm has been modeled by adding noise to an assumed function. In any case, the underlying model of an artificial DM is not known to a tested algorithm, but rather used to derive preference information that is subsequently provided at the method's input.

By contrast, the literature in (multiple criteria) decision making clearly identifies several cognitive biases, psychological phenomena, and inaccuracies occurring at the stage of problem modeling. Obviously, these highly affect preference elicitation and interactive decision making.

Thus, the simple models of DMs most commonly used in the literature for testing IMO algorithms neglect the richness of human behavior and aggregate into a random component a variety of factors that should be rather modeled individually. The important factors that we identified are discussed in the following section. Altogether, they contribute to the idea of implementing a *machine DM* that would take into account the "true" criteria and "true" preference modified appropriately so that to approximate the behavior of real-world DMs.

4.1.2 Modeling cognitive biases, psychological phenomena, and inaccuracies of a machine Decision Maker

We call a machine DM, a model of DM biases and other factors that influence the interaction of the DM with the algorithm by modifying the true criteria and the true preference considered by the DM. This model does not actually specify the criteria and preferences considered by the DM, although different models (different machine DMs) may require them to satisfy certain characteristics. We decided to extend the machine DM from [7], which is based on previous work by Stewart [10], by modeling additional cognitive biases. Stewart [10] assumes a true preference function inspired by prospect theory, that is, a weighted sum of sigmoidals, and the biases modeled are omission of objectives, mixing of objectives and noise. We discuss these phenomena along with the newly considered ones in the following subsections.

Omission of criteria

Omission of criteria consists in neglecting by the algorithm some of the criteria that are internally considered by the DM [10, 11]. For example, attributes of the problem that are modeled as constraints might be considered criteria by the DM. As noted by Stewart [11], the selection of the q criteria among m true ones (with $q < m$) can be conducted as follows:

- assign to each criterion g_j a uniformly generated weight w_j;
- order the criteria from the least ($rank = 1$) to the most important ($rank = m$);
- select q criteria randomly with probabilities proportional to the ranks of criteria so that less important objectives have a higher probability of being omitted.

In this scenario, the machine DM derives its preferences from the m-objective space, whereas the algorithm is allowed to refer the $q < m$ objectives only, i.e.:

$$\vec{g} \in \mathbb{R}^m \quad \text{(DM)} \quad \Rightarrow \quad \vec{\hat{g}} \in \mathbb{R}^{q<m} \quad \text{(algorithm)}. \tag{1}$$

Inversely, the machine DM may neglect some of the m criteria known to the algorithm by constructing its preferences on the basis of $q < m$ criteria only. In this case:

$$\vec{g} \in \mathbb{R}^{q<m} \quad \text{(DM)} \quad \Rightarrow \quad \vec{\hat{g}} \in \mathbb{R}^m \quad \text{(algorithm)}. \tag{2}$$

This bias can be modeled analogously to the previous one.

Mixing of criteria

Even if the criteria internally considered by the DM are preferentially independent, they may have been inadvertently corrupted when modeling the problem by mixing them in such a way that violates preferential independence [6, 10, 11]. Stewart suggests to obtain the new criteria in the following way [11]:

$$\hat{g}_k = (1 - \gamma)g_j + \gamma g_{j+1}, \tag{3}$$

where $\gamma \in [0, 1]$ is a mixing parameter.

In the same spirit, even if the criteria have been defined so that to satisfy the requirement of preferential independence, one may introduce interaction components to the machine DM's value function. For example, Greco et al. [5] considered two particular types of such components corresponding to "bonus" and "penalty" values for positively or negatively interacting pairs of criteria. A bonus is added to (a penalty is subtracted from) the comprehensive value if a pair of criteria is in a positive (negative) synergy for performances of the considered alternatives on the two criteria. These effects may be considered as mutual strengthening or mutual weakening effects, respectively, which are both easily integratable into the model of a machine DM.

Mental fatigue

A great share of MOO methods require the DM to provide the preference information incrementally. On the one hand, this allows both avoiding the necessity of dealing with a large set of preference information pieces already at the initial stages of the interaction as well as controlling the impact of each piece of information (s)he supplied on the delivered results. On the other hand, a lengthy preference elicitation process may result in a mental fatigue of the DM. Such fatigue is defined as a temporary inability to maintain maximal cognitive performance from prolonged periods of cognitive activity (in our case, answering questions that would guide the search) [http://en.wikipedia.org/wiki/Fatigue_(medical), last accessed: 10/03/2015]. Obviously, its onset depends upon an individual DM, but in general it is considered to be gradual. Thus, we have decided to model it as a noise factor $\sigma(k)$ that depends on the number of queries (k) to the DM. We found an exponential model $\sigma_0 \cdot e^{\alpha \cdot k}$ as appropriate for this purpose. Note that a closely related cognitive bias may consist in modeling mistakes of the machine DM just by negating or inverting the preferences derived from its model at random intervals.

Bounded rationality

The limited abilities of the DMs concerning information manipulation and computation have been accounted in the literature within the extensive studies on *bounded rationality* [9]. Indeed, the observed real-world decisions often violate the normative principles according to which all the relevant information should be taken into account. Various phenomena indicating that only a limited part of the available information is accounted in practical decision problems have been framed within so called *decision strategies* or *choice heuristics*. Using reverse engineering, these heuristics can be used for modeling the behavior of a machine DM with bounded rationality. For example, we may refer to:

- the satisficing heuristic [9] which (1) considers the solutions one after another, in a random way, (2) compares the value on each criterion of the current solution to a predefined level, and (3) selects the first alternative which passes this test; this procedure may potentially neglect a large part of the solution set;
- the elimination by aspect heuristic [12] which compares all solutions to a pre-defined aspiration level at each criterion starting from the most important one until a single alternative remains; thus defined, this approach considers a limited number of criteria.

Anchoring

Anchoring is a cognitive bias that describes the common human tendency to rely too heavily on the first piece of information offered (the "anchor") when making decisions. During decision

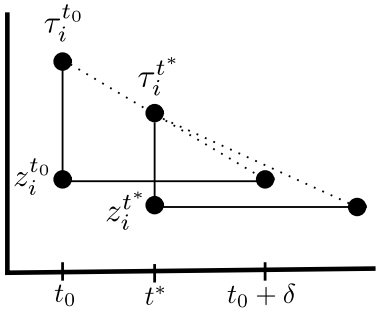


Figure 1 Dynamic model with delayed adjustment of reference point.

making, anchoring occurs when individuals use an initial piece of information to make subsequent judgments [http://en.wikipedia.org/wiki/Anchoring, last accessed: 10/03/2015].

There are two levels of anchoring: a psychological or judgmental level, where there is no notion of gains or losses, and a reference-based level, where the DM defines her reference point according to earlier interactions and resists changing it. As a particular case of anchoring, we considered shifting the reference levels at each interaction according to the median value of each criterion for solutions shown to the DM (or the best solution found). However, we concluded that such a shift may have different interpretations depending on whether dynamic changes in true preference are desirable or not. Thus, we considered two models, where $U()$ is the true preference of the DM and $\hat{U}()$ is the perceived preference that determines the interaction with the algorithm:

- **Static (stable) model**, where interaction does not change the true reference point. In this model, anchoring means that interaction shifts the perceived reference point in $\hat{U}()$ from the true reference point in $U()$. The goal of the algorithm is to minimize the error with respect to the true (static) preference.
- **Dynamic model**, where interaction allows the DM to adjust her reference point (learn), that is, reference point changes in the true preference $U(\vec{z})$. In this model, anchoring means a resistance to change in $\hat{U}()$, when $U()$ changes. The goal of the algorithm is to minimize error with respect to the true preference at the last iteration. Such dynamic model may be treated as an example of evolving DM preferences, when the internal model of the DM changes as a result of the interaction with an algorithm.

We also tentatively discussed an additional *dynamic model with delayed adjustment of reference point* (Fig. 1), where the reference point is updated as:

$$\tau_i^{t^*} = \tau_i^{t_0} + (z_i^{t_0} - \tau_i^{t_0}) \cdot \frac{t^* - t_0}{\delta}$$

where $\delta > 0$ is a delay in adjusting preferences (anchoring). In this model, the goal is to minimize the error with respect to the true preference model at the last iteration $+ \delta$.

Loss aversion

The best solution identified so far in the course of an interaction with the MOO method may be treated by the DM as a reference point. When further exploring the objective space, the DMs tend to collate the newly constructed solutions with her actual reference. Such comparisons may be affected by a loss aversion bias, which implies that the impact of a difference on a criterion is greater when that difference is evaluated as a loss than when the same difference is evaluated as a gain [13]. Such asymmetry in perception of gains and

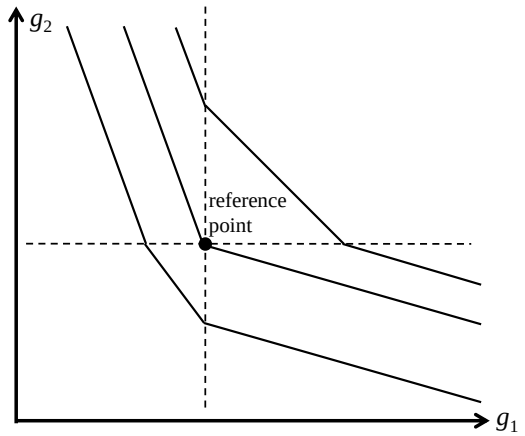


Figure 2 Exemplary indifference curves illustrating loss aversion with respect to the reference point.

losses with respect to the reference point $R = [r_1, \ldots, r_j, \ldots]$ may be easily modeled by transforming the DM's true function u_j in the following way:

$$R_j(x_J) = \begin{cases} u_j(x_j), & \text{if } x_j \geqslant r_j \\ \lambda_j u_j(x_j) - (1-\lambda_j)u_j(r_j), & \text{if } x_j < r_j, \end{cases} \tag{4}$$

where λ_j is a coefficient of loss aversion for criterion g_j. In Figure 2, we provide exemplary indifference curves illustrating the application of thus defined transformation to a two-objective additive linear value function. These isoquants demonstrate that, when observing improvements on both objectives, the perception of the DM is unchanged, whereas the loss in performance at one objective negatively affects the overall quality of the solution from the point of the DM.

4.1.3 Conclusions and future work

The assumption that a true, not directly observable, preference exists is controversial on itself. One consequence of such a model is that, when attempting to avoid biases that distort this function, we are basically telling the DM that her behavior is somehow wrong. We recognize that this is a contentious issue, however, for simulation purposes, the existence of such a true preference is a useful working hypothesis which enables the analysis of how different biases affect interactive algorithms.

When modeling the machine DM, we can draw inspiration from previous literature on theoretical models and empirical studies with human DMs in (multiple criteria) decision making, behavioral economics, judgmental psychology, and cognitive science. In this perspective, we feel that a thorough analysis of how DMs actually behave may gain yet another stream of applied research. That is, apart from having a relevant practice of preference elicitation and designing efficient preference elicitation procedures, we may design the procedures for deriving preference information to be provided at the input of tested algorithms.

Since the ultimate goal of modeling machine DM consists in using them for analysis and comparison of different methods, their models should be extended to various types of preference information, interaction and true preference models, in order to achieve as much generality as possible. During a group discussion, we decided to focus on how to model DM biases in the context of ranking and pairwise comparisons of solutions, nonetheless, we

agreed that it is a worthwhile goal to understand how to simulate DM biases in the context of various types of interaction and preference information, including aspiration levels (goal programming), reference points, trade-offs, select 1 of n, sorting into categories, scoring, intensities of preferences, order of objectives, and desirability functions.

Our plan is to apply the proposed machine DM to NEMO-Choquet [1], which is an interactive evolutionary multiple objective algorithm based on the Choquet integral. Our intuition is that NEMO-Choquet should be able to cope with various biases, such as the mixing of objectives. In the medium term, we wish to do experiments that examine the trade-off between number of questions and quality of information, which decreases because of the fatigue, with respect to different types of questions (pairwise vs. ranking vs. aspiration levels vs. ...). Future machine DMs should also simulate more biases such as bounded rationality heuristics in order to simulate more realistic behaviors.

References

1. J. Branke, S. Corrente, S. Greco, R. Słowiński, and P. Zielniewicz. *Using Choquet integral as preference model in interactive evolutionary multiobjective optimization.* Technical report, WBS, University of Warwick, 2014.
2. J. Branke, K. Deb, K. Miettinen, and R. Słowiński (eds.). Multi-objective Optimization: Interactive and Evolutionary Approaches. *Lecture Notes in Computer Science*, vol. 5252. Springer, Heidelberg, Germany, 2008.
3. J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz. Learning value functions in interactive evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 19(1):88–102, 2015.
4. D. Brockhoff, M. López-Ibáñez, B. Naujoks, and G. Rudolph. Runtime analysis of simple interactive evolutionary biobjective optimization algorithms. In *Parallel Problem Solving from Nature*, PPSN XII, *Lecture Notes in Computer Science*, vol. 7491, pages 123–132. Springer, Heidelberg, Germany, 2012.
5. S. Greco, V. Mousseau, and R. Słowiński. Robust ordinal regression for value functions handling interacting criteria. *European Journal of Operational Research*, 239(3):711–730, 2014.
6. R. L. Keeney. Analysis of preference dependencies among objectives. *Operations Research*, 29:1105–1120, 1981.
7. M. López-Ibáñez and J. D. Knowles. Machine decision makers as a laboratory for interactive emo. In *Evolutionary Multi-criterion Optimization*, EMO 2015. *Lecture Notes in Computer Science*, Springer, Heidelberg, Germany, 2015. To appear.
8. L. Rachmawati and D. Srinivasan. Preference incorporation in multiobjective evolutionary algorithms: A survey. In *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)*, pages 3385–3391, 2006.
9. H. A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1955.
10. T. J. Stewart. Robustness of additive value function methods in MCDM. *Journal of Multi-Criteria Decision Analysis*, 5(4):301–309, 1996.
11. T. J. Stewart. Evaluation and refinement of aspiration-based methods in MCDM. *European Journal of Operational Research*, 113(3):643–652, 1999.
12. A. Tversky. Choice by elimination. *Journal of Mathematical Psychology*, 9(4):341–367, 1972.
13. A. Tversky and D. Kahneman. Loss aversion in riskless choice: a reference-dependent model. *The Quarterly Journal of Economics*, 106(4):1039–1061, 1991.

4.2 Computational Complexity (WG2)

Dimo Brockhoff, Matthias Ehrgott, José Rui Figueira, Luis Martí, Luís Paquete, Michael Stiglmayr, and Daniel Vanderpooten

4.2.1 Introduction

As discrete multiobjective optimization is more and more applied in practice, the necessity arises to categorize both computationally easy and computationally difficult problems. This asks for a thorough complexity theory and analysis of discrete multiobjective optimization problems. Some classical complexity concepts have their limitations when applied to multiobjective optimization, since almost all non-trivial multiobjective optimization problems are NP-hard and intractable.

To the best of our knowledge there are only few publications on the topic of computational complexity for multicriteria optimization problems in general ([5, 1, 2, 3]). In [3], different notions of complexity (depending on the solution concept) are proposed and their interrelations are analyzed. However, there is a wide range of articles investigating the complexity issues of several multiobjective optimization problems and/or their approximability.

Some properties determining the complexity of a problem are related to the decision space, like total unimodularity or other polyhedral properties of the feasible set, others are related to the objective space, like the cardinality of $f(X)$ and its dominance structure. Particularly, the construction scheme, usually applied to show intractability of a problem, uses a very special dominance structure, in which all points are pairwise nondominated.

4.2.2 Dominance Structure

Consider the following biobjective integer problem with a cardinality constraint:

$$\begin{aligned} \min f(X) = &\begin{pmatrix} p^1 X \\ p^2 X \end{pmatrix} \\ \text{s.t. } &\|X\|_1 = \ell \\ &X \in \{0,1\}^n \end{aligned} \tag{P1}$$

where $\|\cdot\|_1$ denotes the 1-norm. Note that $(X, \leq)$ is a strict partially ordered set (i.e. a *poset*), where $\leq$ denotes the component-wise ordering. We build a Hasse diagram of $(X, \leq)$ *via* the cover relation with an implied upward orientation, that is,

1. If $f(x_i) \leq f(x_j)$ holds in the poset, then the point corresponding to x_j appears lower in the drawing than the point corresponding to x_i.
2. The edge between the points corresponding to any two elements x_i and x_j of the poset is included in the drawing if and only if x_i covers x_j or x_j covers x_i, with respect to the given cover relation.

We are particularly interested in relating the (dominance) structure of a given instance of Problem (P1), via the Hasse diagram of the set $\{x_1, x_2, \ldots, x_n\}$, with the cardinality of the efficient set.

Let graph $H = (V, E)$, with vertex set $V = \{x_1, \ldots, x_n\}$ and edge set E, correspond to the graph representation of the Hasse diagram of an instance of Problem (P1). Note that

graph H may be disconnected. We denote by $V_i \subseteq V$, $1 \leq i \leq k$, the vertex set of the i-th connected component of graph H, and by $V^* \subseteq V$ the set of all vertices that are minimal elements in the Hasse diagram. We obtain a directed graph $G = (V \cup \{x_0\}, A)$ by performing the following transformation in graph H:

1. For each edge $\{x_i, x_j\} \in H$, $f(x_i) \leq f(x_j)$, create an arc (x_i, x_j) in G
2. For each vertex $x_i \in V^*$, create an arc (x_0, x_i) in G
3. For each vertex $x_i \in V^*$ and $x_j \in V$, for which it holds that $f(x_i) \not\leq f(x_j)$, create arcs (x_i, x_j) and (x_j, x_i)

Note that this directed graph is connected, may contain cycles and is rooted in x_0. An upper bound on the number of efficient solutions for an instance of Problem (P1) is given by the number of distinct paths of size $\ell + 1$ in G, starting in vertex x_0. In the following we illustrate this transformation on some particular cases and give the upper bound on the number of efficient solutions.

A trivial example is given in the left plot of Figure 3, for which it holds that

$$f(x_i) \leq f(x_j) \iff i < j$$

The graph transformation is given in the right plot of Figure 3, for which there is only one efficient solution. The second example is given in the left plot of Figure 4 with two connected components. The graph transformation is given in right plot, where the dashed arcs correspond to arcs of type 3. In this case, we have $O(\ell^2)$ efficient solutions. A generalization of the previous example is to consider k connected components, each of which with the same structure as that described in the first example. In this case, the number of efficient solutions is $O(\ell^k)$.

Another example is given in left plot of Figure 5, which connects the two connected components from the example in the left plot of Figure 4. In this case, the number of efficient solutions is $O(2^\ell)$. A generalization of the previous example, which can be obtained by connecting k connected components, gives $O(k^\ell)$ efficient solutions.

Another example is given in left plot of Figure 6, which consists of a binary tree, and corresponding transformation in the right plot. An upper bound on the number of efficient solutions is given by the number of binary trees of size ℓ, which corresponds to the Catalan number $C_\ell = \frac{1}{\ell+1}\binom{2\ell}{\ell}$. Finally, left plot of Figure 7 shows a worst case example with the corresponding transformation in the right plot. An upper bound on the number of efficient solutions is given by the $O(n^\ell)$.

4.2.3 Using Total Unimodularity in Multiobjective Optimization

Total unimodularity (TU) is a well known and widely investigated property to identify a certain class of easy-to-solve optimization problems in single objective integer programming (see e. g. [4]). The special polyhedral structure of totally unimodular problems allows to use linear programming to solve integer problems. So the question arises: *Is total unimodularity also useful in multiobjective optimization?*

▶ **Definition 1.** An $m \times n$ integral matrix A is *totally unimodular (TU)* if the determinant of each square submatrix of A is equal to 0, 1 or -1.

▶ **Property 2** (C.f. [4])**.** *If A is TU and b is integral, then linear programs of forms like $\{\min cx : Ax \geq b, x \geq 0\}$ have integral optima, for any c and thus can be solved using (continuous) linear programming.*

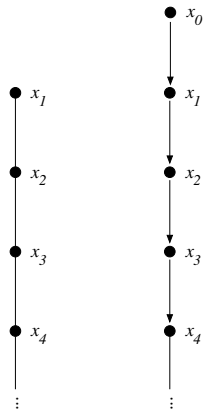


Figure 3 The first example (left) and the graph transformation (right).

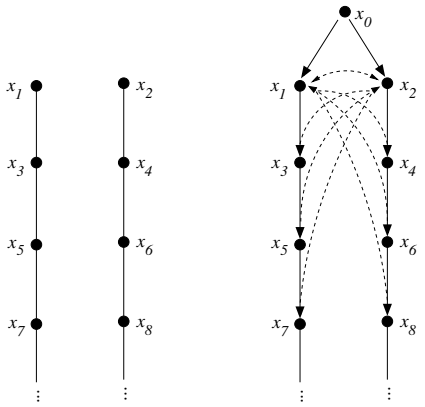


Figure 4 The second example (left) and the graph transformation (right).

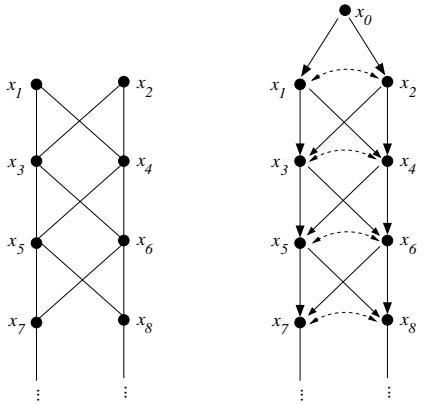


Figure 5 The third example (left) and the graph transformation (right).

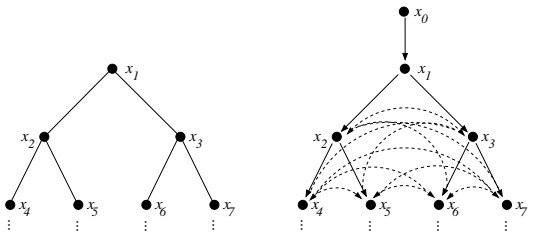


Figure 6 The fourth example (left) and the graph transformation (right).

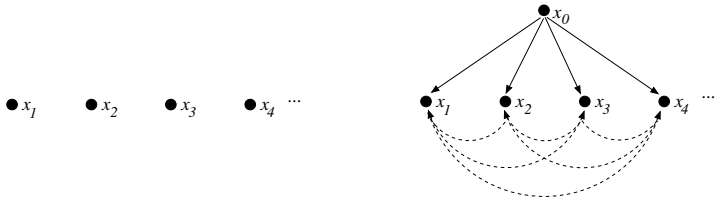


Figure 7 The fifth example (left) and the graph transformation (right).

▶ **Proposition 3.** *The non-dominated set of the following biobjective integer problem*

$$\begin{aligned} \min \begin{pmatrix} c^1 x \\ c^2 x \end{pmatrix} \\ s.t.\ A x = b \\ x \geq 0 \end{aligned} \tag{BOIP}$$

where A is TU can be enumerated in polynomial delay, *when c^2 is*

- *a unit vector (in polynomial time)*
- *any row A^i of A.*

Moreover, all non-dominated points of (BOIP) *are supported.*

Proof. The problems

$$\begin{aligned} \min\ & c^1 x \\ \text{s.t. } & A x = b \\ & x \geq 0 \quad \text{(integer)} \end{aligned}$$

can be solved in polynomial time, since A is TU. The constraint matrices of the ε-constraint problem, namely

$$A' = \begin{pmatrix} & A & & 0 \\ & & & \vdots \\ 0 & \dots & 1 & 1 \end{pmatrix} \qquad \text{or} \quad A' = \begin{pmatrix} & A & & 0 \\ & & & \vdots \\ & A^i & & 1 \end{pmatrix}$$

are then also TU, c.f. [4], page 540. Consequently, the corresponding ε-constraint problems

$$\begin{array}{lll} \min & c^1 x & \\ \text{s.t.} & A' x' = \begin{pmatrix} b \\ \varepsilon \end{pmatrix} & \iff \\ & x' \geq 0 & \end{array} \qquad \begin{array}{lll} \min & c^1 x & \\ \text{s.t.} & A x = b & \\ & c^2 x \leq \varepsilon & \\ & x' \geq 0 & \text{(integer)} \end{array} \tag{EC}$$

can also be solved in polynomial time using linear programming.

There is still to show, that all nondominated points are supported, i. e. every nondominated point can be obtained by solving a weighted-sum scalarization

$$\begin{aligned} \min\ & \lambda c^1 x + (1-\lambda)\, c^2 x \\ \text{s.t. } & A x = b \\ & x \geq 0 \quad \text{(integer)} \end{aligned} \tag{WS}$$

for a value of $\lambda \in [0, 1]$.

We can reformulate (WS) and interpret it as the Lagrange dual of (EC) relaxing the constraint $c^2 x \leq \varepsilon$:

$$\begin{aligned} \min\ & c^1 x + \mu\,(c^2 x - \varepsilon) \\ \text{s.t. } & A x = b \\ & x \geq 0 \quad \text{(integer)} \end{aligned} \tag{LD}$$

with Lagrange multiplier $\mu \geq 0$. Furthermore we apply the result ([4], page 329), that strong duality holds, i. e. $\exists \mu \geq 0$: the optimal values of (EC) and its Lagrange dual (LD) coincide, iff

$$\text{conv}\{x \in \mathbb{R}^n : Ax = b, c^2 x \leq \varepsilon\} = \text{conv}\{x \in \mathbb{R}^n : Ax = b\} \cap \{x \in \mathbb{R}^n : c^2 x \leq \varepsilon\}.$$

This equality holds since all considered polyhedra are TU, and thus we can conclude that every nondominated point can be obtained by weighted sum scalarization with a certain value of $\lambda \in [0,1]$. ◀

Since all the non-dominated points are supported the problem can be solved even more efficiently using dichotomic-search.

Application to the Transportation Problem

Let I be the set of suppliers with capacities to deliver up to an amount of s_i product units, and J be the set of customers with demands d_j. As in the single objective transportation problem we consider the minimization of transshipment costs and additionally we introduce a second objective of the form given in Proposition 2, which corresponds to the minimization (or maximization) of product flow between one supplier and one customer (a) or the minimization (or maximization) of the number of units provided by one supplier (b).

a)
$$\min \sum_{ij} c_{ij}\, x_{ij}$$
$$\min\ x_{12}$$
$$\text{s.t.} \sum_i x_{ij} = d_j \qquad \forall j \in J$$
$$\sum_j x_{ij} \le s_i \qquad \forall i \in I$$
$$x_{ij} \ge 0 \text{ (integer)} \quad \forall (i,j) \in I \times J$$

$$\min \sum_{ij} c_{ij}\, x_{ij}$$
$$\text{s.t.} \sum_i x_{ij} = d_j \qquad \forall j \in J$$
$$\sum_j x_{ij} \le s_i \qquad \forall i \in I$$
$$x_{12} \le \varepsilon$$
$$x_{ij} \ge 0 \text{ (integer)} \quad \forall (i,j) \in I \times J$$

1	…	1							$= d_1$
			1	…	1				$= d_2$
						1	…	1	$= d_3$
1			1			1			$\le s_1$
	1			1			1		$\le s_2$
		1			1			1	$\le s_3$
0	1	0	0	0	0	0	0	0	$\le \varepsilon$

b)
$$\min \sum_{ij} c_{ij}\, x_{ij}$$
$$\min \sum_j x_{1j}$$
$$\text{s.t.} \sum_i x_{ij} = d_j \qquad \forall j \in J$$
$$\sum_j x_{ij} \le s_i \qquad \forall i \in I$$
$$x_{ij} \ge 0 \text{ (integer)} \quad \forall (i,j) \in I \times J$$

$$
\begin{aligned}
\min \quad & \sum_{ij} c_{ij}\, x_{ij} \\
\text{s.t.} \quad & \sum_{i} x_{ij} = d_j \qquad \forall j \in J \\
& \sum_{j} x_{ij} \le s_i \qquad \forall i \in I \\
& \sum_{j} x_{1j} \le \varepsilon \\
& x_{ij} \ge 0 \text{ (integer)} \quad \forall (i,j) \in I \times J
\end{aligned}
$$

1	…	1							$= d_1$
			1	…	1				$= d_2$
						1	…	1	$= d_3$
1			1			1			$\le s_1$
	1			1			1		$\le s_2$
		1			1			1	$\le s_3$
1	0	0	1	0	0	1	0	0	$\le \varepsilon$

Acknowledgments. Luís Paquete acknowledges support by iCIS (CENTRO-07-ST24-FEDER-002003).

References

1. V. Emeličhev and M. Kravtsov. Completeness of vector discrete optimization problems. *Cybernetics and Systems Analysis*, 5(30):693–699, 1994.
2. V. Emeličhev and V. Perepeliza. Complexity of vector optimization problems on graphs. *Optimization*, 22(6):906–918, 1991.
3. K. Fleszar, C. and Glaßer, F. Lipp, C. Reitwießner, and M. Witek. The Complexity of Solving Multiobjective Optimization Problems and its Relation to Multivalued Functions. *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 18, page 53, 2011.
4. G. Nemhauser and L. Wolsey. *Integer and Combinatorial Optimization*, Wiley-Interscience, New York, NY, USA, 1988.
5. P. Serafini. Some considerations about computational complexity for multi objective combinatorial problems. In J. Jahn and W. Krabs, editors, *Recent Advances and Historical Development of Vector Optimization*, volume 294, *Lecture Notes in Economics and Mathematical Systems*, pages 222-232. Springer Berlin Heidelberg, 1987.

4.3 Heterogeneous Functions (WG3)

Gabriele Eichfelder, Xavier Gandibleux, Martin Josef Geiger, Johannes Jahn, Andrzej Jaszkiewicz, Joshua D. Knowles, Pradyumn Kumar Shukla, Heike Trautmann, and Simon Wessing

4.3.1 Introduction

Observing the literature on real-world multiobjective optimization problems, one might notice that many practical applications exhibit considerable heterogeneity regarding the involved objective functions. This working group collected examples of such problems, characterized the kind of heterogeneity that may be found, and identified suitable benchmarks and potential challenges for respective optimization algorithms.

4.3.2 An example

Let $f^1, f^2 : \mathbb{R}^n \to \mathbb{R}$ be nonlinear (objective) functions and let $f^3 : \mathbb{R}^n \to \mathbb{R}$ be a linear objective function. Moreover, let $\Omega \subseteq \mathbb{R}^n$ be the constraint set. Based on these let us consider two multi-objective optimization problems:

$$\min\left(f^1(x), f^2(x)\right), \quad \text{s.t. } x \in \Omega \text{ and} \tag{P1}$$

$$\min\left(f^1(x), f^3(x)\right), \quad \text{s.t. } x \in \Omega. \tag{P2}$$

It is clear that both (P1) and (P2) are classified as nonlinear multi-objective optimization problems. If one applied a *weighted sum method* the scalarized single-objective function remains nonlinear. Therefore, there is no added difficulty (or simplicity) due to the heterogeneity of the objectives in (P2) compared with (P1). *Homotopy-based methods* [13], on the other hand, can use the linearity of objective f^3 in an efficient way, and therefore, (P2) can be solved using such methods in an easier way (compared to the nonlinear problem (P1)). (P2) can also be easier to solve using *population-based heuristics*. A well-known example is the benchmark problem ZDT1 (or ZDT4) where NSGA-II first finds the individual minima of the first, linear objective function, and then spreads along the efficient front.

4.3.3 Motivating Applications

Multiobjective capacitated arc routing problem. Lacomme et al. [19] and Mei et al. [23] consider the multiobjective version of capacitated arc routing problems (CARP). These find application in optimization of salting and removing the snow in the winter or in waste collection by a fleet of vehicles. They consider two objectives, namely the total cost (time) of the routes, which is related to minimization of the total operational costs, and the makespan, i. e., the length of the longest route, which is related to the satisfaction of the clients. Clearly the two objectives differ by mathematical form – sum or maximum of the routes' costs. This difference may also influence the landscapes of these objectives and thus influence their practical difficulty. Consider for example the typical insertion or swap moves for CARP. Such moves modify two routes at a given step. In order to improve the makespan objective the longest route has to be improved, so it has to be one of the modified routes. This means that there are in general less potential moves that could improve this objective and local search may stop at a local optimum very fast. For the total cost objective, on the other hand, many moves may result in an improvement. Please note that this situation is similar to the optimization of either linear (weighted sum) or Chebycheff scalarizing functions. The latter type of functions use a maximum operator. Jaszkiewicz [17] observed that linear functions are easier to optimize than Chebycheff ones in the framework of a multiple objective genetic local search algorithm.

Multiobjective chemical formulation problem. Based on communications with Unilever plc., Allmendinger and Knowles motivated their recent work on heterogeneous evaluation times of objectives [1] using an example from a chemical formulation problem: "We wish to optimize the formulation of a washing powder, and our two objectives are washing excellence and cost. In this case, [...] assessing washing excellence may be a laborious process involving testing the powder, perhaps on different materials and at different temperatures. By contrast, the cost of the particular formulation can be computed very quickly by simply looking up the amounts and costs of constituent ingredients and performing the appropriate summation". Earlier work by the same authors [2] stated that heterogeneous evaluation times could be associated with other lengthy experimental processes such as fermentation, or might occur

because of a need for subjective evaluations from experts. In both studies (ibid.), the authors consider a variety of algorithmic approaches to handling objectives with different "latency", including use of pseudofitness values, and techniques based on interleaving evaluations on the slower and the faster objective(s).

Multiobjective traveling salesman problem with profits. Jozefowiez et al. [18] consider the multiobjective traveling salesman problem with profits. The two objectives are minimization of the tour length and maximization of the collected profits. The tour does not have to include all nodes. TSP with profits is a well known combinatorial problem with multiple applications [10]. Although it is multiobjective by nature, it is usually solved by aggregation of the two objectives, which not only differ by mathematical form but also have different domains. The tour length depends on both the selected cities and the chosen tour, while the profit depends only on the selected cities. Furthermore, the two objectives correspond to two different classes of combinatorial problems. The authors used two sets of moves. The first set optimizes the tour while the second set modifies the set of visited nodes. An interesting observation is that the higher the number of selected nodes, the more difficult is the related TSP subproblem, i. e., optimization of the tour.

Multi-objective optimization in the Lorentz force velocimetry framework. Lorentz force velocimetry (LFV) is an electromagnetic non-contact flow measurement technique for electrically conducting fluids. It is especially suited for corrosive or extremely hot fluids (glass melts, acidic mixtures, etc) that can damage other measurement setups [30]. The magnetic flux density B is produced by permanent magnets and an electrically conducting (σ) fluid moves with a velocity v through a channel. The magnetic field interacts with the fluid and eddy currents develop. The resulting secondary magnetic field acts on the magnet system. The Lorentz force F_L breaks the fluid and an equal but opposite force deflects the magnet system, which can be measured. It holds that $F_L \sim \sigma \cdot \bar{v} \cdot \bar{B}^2$. Fluids with a small electrical conductivity produce only very small Lorentz forces. Thus, a sensitive balance system is necessary for measurement. This limits the weight of the magnet system (we use the magnetization M as surrogate) and causes external disturbances to have a high influence on the force signal. In order to increase the force to noise aspect ratio, the objective function has to take into account two conflicting goals: maximize the Lorentz force and minimize the magnetization.

$$\begin{gathered}\min \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} -F_L(\Phi, \Theta, M) \\ \sum_{k=1}^{8} M_k \end{pmatrix} \\ \text{such that} \\ \Phi_i \in [-\pi, \pi],\ i = 1, \ldots, 8, \\ \Theta_j \in [0, \pi],\ j = 1, \ldots, 8, \\ M_k \in [0, 10^6],\ k = 1, \ldots, 8\end{gathered}$$

The Lorentz force is thereby calculated by a time consuming (20–120 s) simulation run while the magnetization can be calculated analytically. In the above optimization problem $\Phi \in \mathbb{R}^8$ and $\Theta \in \mathbb{R}^8$ describe the direction of the magnetization vector. Both functions are assumed to be smooth. The derivatives of the second objective can be easily determined while already the first derivative of the first objective can only be approximated by numerical differentiation. As this requires in general many functional evaluations, it should be avoided. The second objective is even linear and also the feasible set is a linearly constrained set (there are only box constraints). The first objective is nonlinear and has locally optimal solutions which are not globally optimal.

Portfolio optimization. The portfolio optimization problem is formulated as a bi-criterion optimization problem, where the reward (mean of return) of a portfolio is maximized, while the risk (variance of return) is to be minimized. Practical portfolio optimization problems use extensions to the Markowitz model, and these often use several risk measures, e. g., quantile-based risk measures [3]. These measures replace variance in the standard mean-variance model, thus leading to an entire family of mean-risk portfolio selection models. This makes the problem heterogeneous as the first objective is linear and the second objective has stochastic terms. Many other practical portfolio optimization formulations even use a tri-objective problem so as to find trade-offs between risk, return, and the number of securities in the portfolio [4], which is even more heterogeneous (continuous, stochastic, and integer-valued functions are involved). An overview on extended Markowitz models for further reading can be found in [29]. Conditional values at risk and satisficing constraints can also be incorporated.

Multi-objective inventory routing. The *inventory routing problem* (IRP) describes a generalization of the classical vehicle routing problem (VRP), in such that delivery volumes, i. e., the quantities of the products delivered to customers in a logistics network, are considered to be additional variables. While early research on this problem can be traced back to the 1980s [9], it has only recently been investigated in its true formulation as a multi-objective problem [12]. The bi-objective formulation of [12] introduces two objectives: the inventory levels held by the customers in the network are to be minimized (a typical consideration in just-in-time logistics), and the costs for transporting the goods to the customers are minimized. Obviously, the two criteria are in conflict with each other. A decision support system for this biobjective IRP is visualized in [16]. There, it could be observed that the minimization of the inventory levels is of lower practical difficulty than the minimization of the routing costs. The reasoning behind this is based on the observation that delivery volumes simply are the setting of a single variable value for each customer, and the subsequently held inventories are directly affected by the amount of delivered products. However, the solution of the resulting VRP is difficult even for small data-sets, and in practical cases with reasonable running time restrictions, only (meta-)heuristics appear to be promising solution approaches [15].

Interventional radiology in medical engineering. An essential component of interventional radiology is the procedure of minimally invasive therapeutic interventions, for example in the vasculature. Since the line of sight is interrupted, the interventional material used in these procedures, e. g., catheters, guide wires, stents, and coils, are tracked by imaging techniques. In this application we consider the deformable 3D-2D registration for CT. With the considered method the patient motion during the intervention can be corrected. Only such a procedure can reconstruct artifact-free volumes showing the true position of the interventional material. A bicriterial approach is taken in [11], which is based on raw data and adapts the position of the prior volume immediately to the position included in the raw data without a reconstruction. One objective is the sum of squared differences in raw data domain and the other is a regularization term which originates from physical models for fluids and diffusion processes. An application of a gradient method to this bicriterial problem would require the solution of an implicit differential equation for the computation of a gradient direction. In order to reduce the inhomogeneity of the objectives the bicriteria optimization is done in an alternating manner. The raw data fidelity is minimized by a conjugate gradient descent and the resulting vector fields are then convolved with Gaussian kernels to realize regularization. This alternation between the two objectives is only possible using a special linking term combining both objectives. With this technique one gets the required images with high quality in a faster way.

4.3.4 Aspects of Heterogeneity

Functions of multi-objective problems may differ in several, usually interconnected aspects, of which the following could be identified:

Scaling. An objective function's range of values may be quite different from the range for other objective functions of the problem.

Landscape. Objective functions may differ quite considerably in basic features, as their degree of multi-modality, presence of plateaus, separability, or smoothness. An even richer description can be achieved by calculating empirical summary characteristics such as fitness-distance correlation, auto-correlation, or the numerous features developed under the term exploratory landscape analysis (ELA) [24]. These require evaluating a space-filling sample drawn from the domain of the multi-objective problem. Such features may be less intuitive than theoretical properties, but nonetheless they are designed to correspond to the practical performance of heuristic optimization methods, and thus provide valuable information about the function. However, current ELA features are designed for individual objectives and the design of specific features capturing the multiobjective problem characteristics, like e.g. front shape, local fronts etc., is still an open research topic. The relationship between the individual ELA features and multiobjective problem characteristics would be very helpful in assessing the influence of objective heterogeneity.

Evaluation time. Each objective or constraint function of a multi-objective problem may take a different amount of time to evaluate. These differences may result from different theoretical complexity of the functions, different size of the domain of the functions (see Domains below), or other differences. In practical problems, the heterogeneity of evaluation times could be large, for example if one objective function was a simple sum while the other one was evaluated by conducting a physical experiment [1, 2]. A further point related to evaluation time is that some functions may be computed more quickly if another solution, whose function value is known and differing in the values of a small number of decision variables, is available. In some cases the ability to evaluate efficiently the objective functions by computing the difference (or delta) from an existing solution is very important (e. g. in symmetric TSP) for local search methods.

Theoretical and practical difficulty. Some functions may be more or less difficult to optimize in terms of the number of solutions that must be explored in order to find an optimum (e. g., using a local search or other iterative search method). Differences in practical difficulty between the objectives could be a result of different theoretical complexity of the functions, or different domain sizes, or different properties of the fitness landscape.

Domains. Let us consider the binary relation "intersects with" between all pairs of domains of the objective functions and constraints as a graph. This graph may have only one connected component, or there would be no conflict between some of the functions. However, the domains do not necessarily have to be completely identical, either. This holds especially for constraints, which usually concern only a subset of the variables. Consequently, not all functions have to be defined on variables of the same data type.

Parallelization. Each objective function could have different restrictions regarding the amount of parallelization. E. g., some objective functions might require physical equipment or software licenses, which restrict the number of function evaluations that can be executed in parallel.

Problem class. It may be known that some objective belongs to a different problem class than another. Examples are the aforementioned TSP and shortest path.

Analytic form vs. black box. Some objective function may be available in analytic form, while another may be available only as a black box. This usually implies that the evaluation time differs considerably between the objective functions (see above). Moreover, while for the analytic functions the derivatives can be calculated, they can only be approximated for black-box functions using numerical differentiation.

Determinism. Some objective functions of a problem may be stochastic, while others might be deterministic.

4.3.5 Benchmarks

For investigating this topic in controlled experiments, "artificial" benchmark problems are a useful tool. Here we argue which existing benchmarks exhibit heterogeneity and how even more heterogeneous ones could be constructed.

Continuous benchmarks. In the area of evolutionary multi-objective optimization a large number of continuous test instances are collected in [14]. These have different landscapes as for instance one objective is linear and the second one is highly nonlinear. This is used to create convex, non-convex, mixed convex-concave, and multi-modal problems. The objectives in ZDT, SZDT, RZDT, and WFG test problem instances are heterogeneous. One of the test functions is linear (or piecewise linear) while the other objective(s) are highly nonlinear and multi-modal. DTLZ test problem instances, on the other hand, use similar objective functions (using sine and cosine terms) and hence are not heterogeneous at first sight. They might differ in terms of ELA features, however. Simple benchmark functions like e.g. the Schaffer or Binh problems are homogeneous, though. Instances with differing evaluation times can be easily constructed by inserting a time delay in the respective functions. Moreover, noise can be added to a subset of the objectives in order to address heterogeneity in terms of determinism as discussed above.

KP benchmarks. We carried out some preliminary experiments to construct heterogeneous discrete problems. The bi-objective unidimensional 01 knapsack problem (KP) was used as a basis for these investigations. Its objective is to optimize $\vec{f} = (\max \sum_{j=1}^{n} c_j^1 x_j,$ $\max \sum_{j=1}^{n} c_j^2 x_j)^T$ under the side constraints $\sum_{i=1}^{n} w_j x_j \leq \omega$ and $x_j \in \{0, 1\}$. Four families (A/B/C/D) of instances are already provided by the MOCOlib [25]. Among them are family A, where c_j^1, c_j^2 are randomly generated for $i = 1, \ldots, n$ $(1 \leq c_j^1, c_j^2 \leq 100)$, and family C, which contains patterns (plateaus where l_i is the length and v_i is the value) created by choosing v_i randomly in $\{1, \ldots, 100\}$, $c_1^1 = c_2^1 = \ldots = c_{l_1}^1 = v_1$, and $c_{l_1+1}^1 = c_{l_1+2}^1 = \ldots = c_{l_1+l_2}^1 = v_2$. In [8] it was observed that the patterns tend to make the MOCO problem harder to be solved. So, our preliminary impression is that the patterns provide a way to introduce a form of heterogeneity in functions.

We also constructed some new families by combining different existing ones, e. g., by taking objective 1 and resource constraint from family A and objective 2 from family C. This way, we obtained five new families, called AC, AL, AZ1, AZ12, and AZ3. In preliminary experiments with a solver taken from [5, 6], the comparison of results obtained on A, AZ12, and AZ3 indicated that the presence of "null" plateaus seems to affect the performance of the solver negatively. More research on this topic shall follow.

Constraint satisfaction benchmarks. Max-SAT-ONE [28, 22] is an example of a bi-criterion constraint satisfaction problem with objectives heterogeneous in their (assumed) computational complexity class. The first objective is NP-hard, while the other objective is a simple sum over variables and is hence linear.

Max-SAT-ONE is a relative of the logical Satisfiability (SAT) problem, an archetypal decision problem with a central role in theoretical computer science as the first to be proved NP-Complete [7]. In an instance of the SAT problem a number c of logical clauses involving a number n of Boolean variables are presented. The problem is to determine whether there is an assignment to the variables that satisfies all the clauses. The optimization form of the problem, known as MAX-SAT, is also well-known. The problem, the subject of intensive research for a number of years, follows the same form as SAT but for the objective, which is now to maximise the number of satisfied clauses. The problem is NP-hard, and examples of techniques developed for the problem can be found in [20, 27].

Max-SAT-ONE has been studied in the context of constraint programming [22] and decomposition methods in multiobjective optimization [28]. The first objective is that of MAX-SAT, while the second one is to maximize the number of variables with an assignment of TRUE. This leads to a discrete Pareto front with at most n distinct Pareto optimal points.

TSP benchmarks. One of the possibilities is to use a MOCO problem with objectives defined mathematically in the same way, but with different distribution of parameters. Paquete [26] and Lust and Teghem [21] proposed a set of travelling saleperson (TSP) instances with various classes of objective functions:

- Euclidean instances: the costs between the edges correspond to the Euclidean distance between two points in a plane, randomly located from a uniform distribution.
- Random instances: the costs between the edges are randomly generated from a uniform distribution.
- Clustered instances: the points are randomly clustered in a plane, and the costs between the edges correspond to the Euclidean distance.

They also proposed mixed instances: the first cost comes from the Euclidean instance while the second cost comes from the random instance. They observed some differences in behavior of the multiobjective algorithms for these instances. The Lin-Kerninghan heuristic used in the first phase required significantly more time for random than for Euclidean instances. The Pareto local search used in the second phase was on the other hand faster on Euclidean instances due to much lower number of efficient solutions. The time performance of mixed instances was in between in both cases.

The above mentioned multiobjective traveling salesman problem with profits [18] is an interesting candidate for discrete benchmark problem with heterogeneous objectives. It is relatively simple in definition, based on well studied TSP problem, and contains several aspects of heterogeneity – different mathematical definitions, different difficulty, different domains.

4.3.6 Conclusions and Outlook

Our study suggests that heterogeneity between the objectives of a multiobjective optimization problem is both common and yet little understood (or even considered) in the literature. We have made a modest start on providing motivating examples and beginning a characterization of this complex feature. There seems to be a rich vein to investigate further, and much work to do in proposing and testing suitable methods.

References

1 R. Allmendinger, J. Handl, and J. D. Knowles. Multiobjective optimization: When objectives exhibit non-uniform latencies. *European Journal of Operational Research*, 243(2):497–513, 2015.

2 R. Allmendinger and J. D. Knowles. Hang on a minute: Investigations on the effects of delayed objective functions in multiobjective optimization. In *Evolutionary Multi-Criterion Optimization*, volume 7811 of *Lecture Notes in Computer Science*, pages 6–20. Springer, 2013.

3 K. P. Anagnostopoulos and G. Mamanis. Multiobjective evolutionary algorithms for complex portfolio optimization problems. *Computational Management Science*, 8(3):259–279, 2011.

4 K. P. Anagnostopoulos and G. Mamanis. A portfolio optimization model with three objectives and discrete variables. *Computers and Operations Research*, 37(7), 1285–1297, 2010.

5 A. Cerqueus, X. Gandibleux, A. Przybylski, and F. Saubion. Efficacité des heuristiques de branchement pour le branch-and-bound multi-objectif: vers une gestion plus dynamique. *ROADEF 2014: 15e Conférence ROADEF de la société Française de Recherche Opérationnelle et Aide à la Décision*, Bordeaux, France, 2014.

6 A. Cerqueus, A. Przybylski, and X. Gandibleux. Surrogate upper bound sets for bi-objective bi-dimensional binary knapsack problems. *European Journal of Operational Research*, 2015. To appear.

7 S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

8 F. Degoutin and X. Gandibleux. Un retour d'expériences sur la résolution de problèmes combinatoires bi-objectifs. *Journée "Programmation Mathématique Multiobjectifs"* (PM20), Angers, France, 2002.

9 A. Federgruen and P. Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 32, no. 5, 1019–1037, 1984.

10 D. Feillet, P. Dejax, M. Gendreau. Traveling salesman problems with profits. *Transportation Science*, 39(2):188–205, 2005.

11 B. Flach. *Rekonstruktion mit Vorinformation für die Niedrigstdosis-CT-Fluoroskopie in der Interventionellen Radiologie*. Ph. D. thesis, University of Erlangen-Nürnberg, Erlangen, 2014.

12 M. J. Geiger and M. Sevaux. The biobjective inventory routing problem – problem solution and decision support. In: Network Optimization, Volume 6701 of *Lecture Notes in Computer Science*, pages 365–378. Springer, 2011.

13 C. Hillermeier. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach.* Birkhäuser Basel, 2001.

14 S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.

15 S. Huber and M. J. Geiger. Swap Body Vehicle Routing Problem: A Heuristic Solution Approach. In: Computational Logistics: Proceedings of the 5th International Conference (ICCL 2014), Volume 8760 of *Lecture Notes in Computer Science*, pages 16–30, Springer, 2014.

16 S. Huber, M. J. Geiger and M. Sevaux. Simulation of Preference Information in an Interactive Reference Point-Based Method for the Bi-Objective Inventory Routing Problem. *Journal of Multi-Criteria Decision Analysis*, 22(1-2):17–35, 2015.

17 A. Jaszkiewicz. Genetic local search for multiobjective combinatorial optimization. *European Journal of Operational Research*, 137(1):50–71, 2002.

18 N. Jozefowiez, F. Glover, and M. Laguna. Multi-objective Meta-heuristics for the Traveling Salesman Problem with Profits. *Journal of Mathematical Modelling and Algorithms*, 7(2):177–195, 2008.

19 P. Lacomme, C. Prins, and M. Sevaux. A genetic algorithm for a biobjective capacitated arc routing problem. *Computers and Operations Research*, 33(12):3473–3493, 2006.

20 J. Larrosa, F. Heras, and S. de Givry. A logical approach to efficient MAX-SAT solving. *Artificial Intelligence*, 172(2):204–233, 2008.

21 T. Lust and J. Teghem. Two-phase Pareto local search for the biobjective traveling salesman problem. *Journal of Heuristics*, 16(3):475–510, 2010.

22 R. Marinescu. Exploiting problem decomposition in multi-objective constraint optimization. In Principles and Practice of Constraint Programming – CP 2009, volume 5732 of *Lecture Notes in Computer Science*, pages 592–607. Springer, 2009.

23 Y. Mei, K. Tang, and X. Yao. Decomposition-Based Memetic Algorithm for Multiobjective Capacitated Arc Routing Problem. *IEEE Transactions on Evolutionary Computation*, 15(2):151–165, 2011.

24 O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph. Exploratory landscape analysis. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation (GECCO '11)*. ACM, pp. 829–836, 2011.

25 MOCOlib: The MOCO Numerical Instances Library. http://www.mcdmsociety.org/MCDMlib.html

26 L. Paquete. *Stochastic local search algorithms for multiobjective combinatorial optimization: methods and analysis*. Ph. D. thesis, FB Informatik, TU Darmstadt, 2005.

27 M. G. C. Resende, L. S. Pitsoulis, and P. M. Pardalos. Approximate solution of weighted MAX-SAT problems using GRASP. *Satisfiability problems*, 35:393–405, 1997.

28 E. Rollón and J. Larrosa. Bucket elimination for multiobjective optimization problems. *Journal of Heuristics*, 12(4-5):307–328, 2006.

29 M. C. Steinbach. Markowitz Revisited: Mean-Variance Models in Financial Portfolio Analysis. *SIAM Review*, 43(1), 31–85, 2001.

30 D. Terzijska, M. Porcelli, and G. Eichfelder. Multi-objective optimization in the Lorentz force velocimetry framework. In Book of digests & program / OIPE, *International Workshop on Optimization and Inverse Problems in Electromagnetism 13*, Delft, pp. 81–82, 2014.

4.4 Visualization in Multiobjective Optimization (WG4)

Carlos M. Fonseca, Carlos Henggeler Antunes, Renaud Lacour, Kaisa Miettinen, Patrick M. Reed, and Tea Tušar

4.4.1 Introduction

Visualization is useful and needed for many purposes in multiobjective optimization. Roughly speaking, we can identify three main uses for visualization: as a tool for analyzing either sets of solutions or individual solutions, as part of decision support in applying interactive methods, and as a tool for analyzing performance of algorithms. One can say that visualization itself has multiple objectives. On one hand, visual representations or graphics should be easy to comprehend so that no relevant information is lost but, on the other hand, no additional,

unintentional information should be included as a byproduct. Whichever way visualization is used, it is closely connected to graphical user interface design.

People are familiar with e.g. simple bar charts or pie diagrams and understanding them is not regarded demanding. However, as soon as the dimensions or the amount of the information to be visualized increases [34], there are many challenges involved. In both developing visualizations and interpreting them, one should avoid introducing biases like having unintentional meanings associated to colors (which may be culture-dependent) or assignment of axes to represent different dimensions of the information being visualized. It may not be possible to generate self-explanatory visualizations but cognitive training is needed. Overall, contextual awareness of all parties involved is important.

As mentioned, visualization has many purposes and has a lot to offer for various needs. Surveys of visualization techniques for multiobjective optimization and multiple criteria decision making are given in [21, 25, 29, 43]. The work [29] also contains many further references relevant for visualization. When analyzing sets of solutions or individual solutions, it is beneficial to exploit the geometric structure of Pareto front approximations or the connection between the decision and the objective space. As part of decision support, new ways can be provided to decision makers for directing the progress of interactive methods. When analyzing the performance of algorithms, one can exploit recent algorithm advances. Furthermore, new technologies like virtual or augmented reality and 3-D printers etc. give a new meaning to visualization.

In what follows, we briefly consider different visualization problems and tools and provide further references.

4.4.2 Open problems

Visualization is generally seen as a powerful means of conveying information to humans by harnessing the strong information processing capabilities associated with human vision and cognition. However, while humans are usually quite comfortable with two-dimensional data, effective visualization rapidly becomes more difficult as the number of dimensions increases, due to a combination of human and technical factors. Indeed, multidimensional data visualization is intimately tied to the features and limitations of the computing and display technologies available and to the cognitive limitations associated with the high numbers of dimensions and high volumes of data.

In this section, some of the challenges posed by visualization in the context of multiobjective optimization are discussed, from selecting the information to be visualized, through to the visual representation of individual solutions and of families of solutions, possibly under uncertainty, and finally to user interaction.

Selecting information to be visualized

Large amounts of high-dimensional data impose both a computational burden (on equipment) and, more importantly, a cognitive burden on users that may simply render visualization ineffective or even impossible. Therefore, information (data and/or dimensionality) reduction techniques are often required, the goal being to provide the user with a sufficiently accurate representation of the data which highlights the most relevant features without introducing unintended artifacts. In practice, as elaborate representations are often used to embed multiple dimensions into two- or three-dimensional representations, cognitive training is often required before users can usefully process such representations.

Particularly in multiobjective optimization, users (as decision makers) are usually concerned both with the values and with the relations among the various data points. Appropriate notions of Pareto front approximation quality and of the relative importance of the different objectives are, therefore, required. Once such concepts are established, selecting the information to be visualized reduces to solving the corresponding computational problems, which is often another challenge in itself, especially as the number of objectives increases.

Identifying representative sets. For many continuous, discrete or mixed discrete-continuous multiobjective optimization problems, one can compute a large set of Pareto optimal or nearly Pareto optimal solutions. In such a case, it becomes more difficult to compare solutions in the decision space and identify the possible tradeoffs between them in the objective space.

For visualization, one can follow two approaches, possibly combined, to deal with the large size of a set of computed solutions. First, given a set of solutions, one can identify a subset of solutions which (1) has a smaller size and (2) satisfies some quality criterion. Two widely used criteria are the hypervolume indicator, to be maximized, and the additive or multiplicative ε-coverage. Satisfying both (1) and (2) seems to be difficult above the bi-objective case (see e.g. [6] for the ε-coverage criterion), but optimal subset selection can now be performed rather efficiently based on the hypervolume indicator and ε-dominance [7, 22], but also uniformity and coverage [46].

Second, one can try to group together solutions that behave similarly in the decision space or in the objective space. Clustering techniques are used to this end. In the objective space, the Euclidean distance is relevant for quantitative objectives. In the decision space, the distance between solutions has to be chosen according to the context, in order for the result of the clustering to be meaningful. An interesting recent approach, found in [45], attempts to generate clusters of solutions that are "compact and well separated" both in the decision space and in the objective space. To this end, two validity indexes are defined to minimize intracluster distances and maximize intercluster distances, in the decision and objective spaces, respectively. Therefore the clustering problem is itself a bi-objective problem for which one seeks compromise solutions, i.e. clusters of solutions that are good both in the decision and objective spaces..

Reducing objective space dimensionality. Dimensionality reduction is a fundamental task in data visualization. In particular, all data must be projected on two dimensions to be displayed e.g. on a computer screen. Such a physical limitation may be alleviated to some extent by 3D display technologies (whether stereoscopic, volumetric or holographic) and/or by resorting to animation, but ultimately the number of dimensions that can be used directly is low.

At a more abstract level, the number of visualization axes may be extended further, e.g. by associating them with properties of the different graphical objects displayed, as with bubble charts. In addition to the cognitive training required to interpret such representations, a fundamental limitation of those techniques is that the representation of a point may then occlude that of another point. Thus, the amount of data displayed, as discussed in the previous subsection, the number of dimensions actually represented, or both, need to be reduced.

As pointed out by Brockhoff and Zitzler [8], there are two distinct approaches to dimensionality reduction: feature extraction and feature selection. Considering the visualization of the objective space in multiobjective optimization, feature extraction consists in producing a (small) set of arbitrary axes, possibly by non-linearly combining the original objectives, so as to represent the given data as well as possible. Principal component analysis and maximum

variance unfolding have been used for objective reduction [41], and are are examples of such techniques. Unfortunately, although they may preserve certain types of relationships in the data, dominance relationships are usually not preserved [38], and unwanted biases may be introduced in the representation of Pareto front approximations. The cognitive burden imposed on the decision maker, who has to accommodate additional, artificial, objectives, and deal with potentially misleading dominance information, is also increased.

A feature selection approach to objective space dimensionality reduction, on the other hand, consists in selecting a subset of the original objectives to be visualized allowing dominance information to be more strictly preserved. Since conflicting objectives are at the heart of multiobjective optimization, it is natural to see non-conflicting objectives as good candidates to be discarded. More specifically, objectives that do not affect the set of Pareto optimal solutions are termed redundant, and can be safely omitted from the optimization, although they may be of semantic interest to the decision maker [2].

Several interpretations of what conflicting objectives are have been proposed. For Purshouse and Fleming [38], there is conflict between two objectives when improvement in one objective leads to deterioration with respect to the other. A similar view is adopted also by other authors [23, 41], although the actual definition of conflict may vary. Brockhoff and Zitzler [8], on the other hand, define conflict as a relation between sets of objectives, based on the structure of the corresponding weak Pareto dominance relations. Since such a notion of conflict is often too strict, they extend it using the concept of ε-dominance to arrive at a measure of degree of conflict, and at a subset selection formulation of objective reduction.

Assigning objectives to visualization axes. Prior to visualization, one needs to decide how to map the objectives to visualization axes. While this might seem straightforward and is often done implicitly by assigning objectives to visualization axes in their existing order, many visualization methods are order-sensitive and produce significantly different visualizations for different arrangements of objectives. Consider, for example, bubble charts, parallel coordinates [18], radar charts (or star plots) [10], radviz [17], interactive decision maps [24], hyper-space diagonal counting [1], heatmaps [37], hyper-radial visualization [11] and prosections [43].

A lot of research on this topic, called also axes (re-)ordering, has been devoted to parallel coordinates. Assigning the objectives to parallel coordinates so that a similarity measure is maximized is an NP-complete problem [4]. While the similarity between adjacent objectives/axes seems to be the focal point of most work, our working group has agreed that in multiobjective optimization we often need to show conflicts between objectives. These conflicts can be difficult to observe if similarity between adjacent objectives is enforced. We are also missing more research of other methods such as bubble charts, where one needs to determine which of the objectives is going to be represented with color (or size), and discussions on how decision maker's preferences influence such choices.

Visualizing solutions and surfaces

The illustration of solutions and surfaces in objective space plots is a valuable tool not just for dynamically elucidating the progress of the algorithms but also exploiting the results in applications (e.g. enabling to shed light on some features of the problem). Meaningful graphical displays should offer domain experts information about the range of Pareto optimal solutions and the assessment of the trade-offs between the competing objectives, thus conveying relevant information to aid the selection of a final recommendation or a reduced set of solutions for further screening. Human information processing strongly relies on

visual processes to deal with large amounts of data and unveil patterns that lead to sounder decisions, thus minimizing cognitive effort.

Different types of plots, namely scatter plots, are used to visualize 2-D and 3-D (approximations of) Pareto optimal sets. These plots are quite informative in 2-D problems and in most cases provide useful information in 3-D problems, although in this case visualization challenges may already arise due to the complexity of the surfaces or sets of solutions to be displayed. Additional information may be portrayed in scatter plots using size or color (e.g. bubble charts). Whenever the problem has more than three objective functions, sometimes projection techniques may be of help, but no general technique exists offering straightforward visualization in higher dimensions ensuring clarity, intuitiveness, and intelligibility. Dimension reduction approaches to obtain 2-D or 3-D mappings include, among others, self-organizing maps [20] and interactive decision maps [24], which attempt to highlight different features of data under analysis. Self-organizing maps are unsupervised neural networks that generate a mapping of the high dimension data into cells (array of nodes) usually in 2-D, which may then be clustered according to some similarity measure. Nodes are associated with weighting vectors (one vector per node), which are sorted and adapted such that similar data are mapped to the closest node. Interactive decision maps approximate the feasible objective set (and the objective points dominated by it) by developing frontiers of bi-objective slices that display as "topographical" maps (i.e. avoiding intersections). Animation, or its snapshots, can be used to deal with problems with more than three objectives.

Techniques aimed at encompassing the whole information include parallel coordinates [18] and heatmaps [37]. In parallel coordinate plots each evaluation dimension is visualized on a vertical axis and each data point is represented as a line connecting the corresponding values on those axes. A high number of dimensions to be visualized using parallel coordinate plots and too many data points may result in a dense and unclear view. Heatmaps organize data in rows (solutions) and columns (objective function or other feature under analysis). An extensive use of color is made to convey information of the matrix elements, although some color schemes are criticized with the arguments of the lack of a natural perceptual ordering or erroneous perceptions of color gradients due to color changes between matrix cells. In some way, enabling the interchange of rows or columns may circumvent these issues.

In general, in projection schemes there is no Pareto dominance preserving mapping from a higher- to a lower-dimensional space, i.e. erroneous dominance relations may appear in lower-dimensional displays that are not present in original points. [43] use the projection of a section ("prosection") to visualize 4-D points in 3-D in an intuitive manner, in such a way that the shape, range and distribution of points are reproduced.

Empirical attainment functions (EAF) [15, 14], which are associated with the probabilistic distribution of the (approximation of) Pareto sets, can also be used to offer visualizations, by using cutting planes to cut through the 3-D objective space of the EAF values and display the resulting intersections in 2-D [44].

Visualizing uncertainty

Uncertainty is an inherent characteristic of real-world problems arising from multiple sources of distinct nature. It is generally unfeasible that mathematical and decision aid models could capture all the relevant inter-related phenomena at stake, get through all the necessary information, and also account for the changes and/or hesitations associated with the expression of the decision makers' preferences. In addition to structural uncertainty associated with the knowledge about the overall system being modeled, input data (model coefficients and parameters) may suffer from imprecision, incompleteness or be subject to changes.

Preferences are often ill-specified at the outset of a decision support process and they should be progressively strengthened through experimentation and learning for increasing the confidence in a final recommendation. Once a final solution is selected for execution, the decision variables may drift from the computed values, e.g. in case they are associated with some components of an engineering design project. That is, uncertainty may arise in the model structure, in the mathematical model coefficients, in preferences and in the behavior of the decision variables after implementation.

Therefore, it is of utmost importance to provide decision makers with robust conclusions. The concept of robust solution is not uniformly defined in the literature but it is generally linked to the guarantee of a certain degree of "immunity" to data perturbations and adaptive capability (or flexibility) regarding an uncertain future and ill-shaped preferences, leading to an acceptable performance even under a plausible set of unfavorable conditions [12, 5, 19]. In this setting, uncertainty visualization techniques may have a role in influencing decisions and decision makers' confidence in the recommendations to be adopted. Several approaches have been proposed to communicate to decision makers the effects of uncertainty upon solution quality, which should be tailored to the context of the study in order to convey useful information to decision makers. In general, uncertainty adds further dimensions to the visualization to display the uncertain outcomes. How this is operationalized also depends on how uncertainty is captured (e.g., probability distribution functions, fuzzy sets, intervals).

The most common approach is juxtaposition, i.e. providing a visualization of the effects of uncertainty in a separate display together with a crisp representation. This approach can be extended by means of a toggle capability, thus enabling swapping between the crisp representation and the uncertain ones possibly controlled by a "perturbation" parameter. This technique also enables to offer a dynamic view of the uncertain outcomes associated with model coefficients or preferences uncertainty. For instance, dynamic bars juxtaposed to a crisp solution representation may help in distinguishing the relationship between desirable and undesirable outcomes according to uncertainty parameters and user-defined thresholds of acceptability. Overlay techniques are also used on top of the visualization of crisp outcomes, by combining different types of displays. Colors can be added to represent degrees of uncertainty associated with solutions.

These approaches can be combined with pictorial solution displays, for instance in network optimization problems in which solutions can be shown on a (real or schematic) map. In these cases both value uncertainty and positional uncertainty would be at stake as information to be conveyed to decision makers. Some approaches have been specifically designed to assist decision makers in dealing with uncertainty visualization in multiobjective optimization models. In this context, the main issues stated in the previous sections regarding visualization in high dimensional spaces apply, with even more significance since additional dimensions should be taken into account. The hyper-space diagonal counting method maps the n-dimensional Pareto front to two- or three- dimensional data thus enabling to visualize it in a succinct way [1]. However, as reported in [35] this method does not preserve all neighborhoods when collapsing the n-dimensional data onto two or three dimensions, i.e. different grouping schemes of the n-dimensional neighborhoods may lead to different visualizations. In [35], the hyper-radial visualization method proposed in [11] is used to incorporate into the analysis random and epistemic uncertainty associated with preference choices.

Visualization in Interactive Solution Processes

In interactive multiobjective optimization methods, a solution pattern is formed and repeated so that a decision maker iteratively takes part and provides preference information to direct

the solution process (see, e.g. [26, 27, 33, 39]). Her/his preference information is used to generate one or more new Pareto optimal solutions. In this way, only those Pareto optimal solutions that are interesting to the decision maker are generated and the decision maker considers a small set of Pareto optimal solutions at a time. If visualization is used to provide preference information or feedback to guide the interactive method, we can call it visual steering.

It is important to utilize visualization in both providing preference information and analyzing the solutions generated. This should enable the decision maker to express one's wishes of what kind of solutions are more desirable and understand the consequences of these preferences and compare the solutions obtained. Different methods utilize different preference information and offer different types of information to the decision maker which naturally means that different visualizations are needed. The dimensions and the complexity of the problem also set their own requirements on the visualization techniques to be applied.

As mentioned in the introduction, visualization is closely connected to graphical user interface design. Examples of studies in user interface design for the WWW-NIMBUS implementation [31] of the interactive NIMBUS methods [32] are given in [30, 40]. Further example of user interface design are given in [42] for the interactive Pareto Navigator method [13] and in [16] involving heatmap visualizations and particle swarm optimization.

Different people prefer different visualizations and, thus, it is desirable not to use only one but different visualizations that the decision maker can compare and combine or switch between them. The objective space typically has a lower dimension than the decision space and that is why the consideration of preferences often takes place in the objective space but the connection between the two spaces can be important. It may, e.g. be necessary to consider the corresponding solution in the decision space to be able to evaluate the goodness of a solution in the objective space. In this, the visualizations in the decision space are typically problem-specific whereas objective vectors can be visualized with problem-independent visualizations. Further research is still needed to enable steering in both decision and objective spaces.

One can think of taking full advantage of visualization in connection of interactive methods from at least two perspectives: starting from what the method needs or starting from what visualization techniques can offer. This will likely lead to new method development and new software implementations. Examples of software implementations including visualizations are Grapheur [9], IND-NIMBUS [28, 36] and iMOLPe [3].

When visualization and the interaction are successful, it may also lead to reformulating the problem instead of solving it. The challenge of making the most of the expertise of the decision maker is crucial in the success of applying interactive methods. Visualizations can be a key in not only expressing preference information but enabling insight gaining and learning.

4.4.3 Tools

Many tools for visualization of multidimensional data exist. This working group wanted to to provide a (non-exhaustive) list of tools that are especially suitable for visualization in multiobjective optimization and are often used by the researchers in this field.

Free and/or open-source software

Visualization Tool Kit (VTK) (http://www.vtk.org/) is a software system for 3-D computer graphics, image processing and visualization. The VTK library is used by a number of

scientific data visualization tools, such as Mayavi (http://code.enthought.com/projects/mayavi/), ParaView (http://www.paraview.org/) and VisIt (https://wci.llnl.gov/simulation/computer-codes/visit).

A separate (non VTK-based) environment for scientific computation, data analysis and data visualization is SCaVis (http://jwork.org/scavis/). The data visualization tool XmdvTool (http://davis.wpi.edu/xmdv/) supports a variety of interaction modes and tools, including brushing, zooming, panning, and distortion techniques, and the masking and reordering of dimensions.

Two powerful high-level programming languages that include numerical computation and optimization in addition to visualization are Scilab (http://www.scilab.org/) and GNU Octave (https://www.gnu.org/software/octave/).

Some basic tools that can be used to produce publication quality figures comprise gnuplot (http://www.gnuplot.info/), matplotlib (http://matplotlib.org/), GLE (http://www.gle-graphics.org/), and PGFPlots (http://pgfplots.sourceforge.net/). When producing such plots, the online tool ColorBrewer (http://colorbrewer2.org/) can be used to help select good color schemes.

Proprietary software

Tools, such as Optimus (http://www.noesissolutions.com/Noesis/), modeFRONTIER (http://www.esteco.com/modefrontier/), OptiY (http://www.optiy.eu/) and DecisionVis (https://www.decisionvis.com/) use advanced and interactive visualization methods to aid the engineering design and optimization process.

MATLAB (http://www.mathworks.com/products/matlab/) is a high-level language and interactive environment that can be used for optimization as well as visualization of multidimensional data.

Finally, Trade Space Visualizer (http://www.atsv.psu.edu/) is a data visualization program designed to help users explore multidimensional data sets to discover relationships between features.

Note that while these tools are generally not free, some offer free academic licenses.

Acknowledgments. In the working group discussions in Dagstuhl, Ralph E. Steuer presented interesting visualization challenges. Carlos Henggeler Antunes and Carlos M. Fonseca acknowledge support by iCIS (CENTRO-07-ST24-FEDER-002003).

References

1 G. Agrawal, C. L. Bloebaum, and K. Lewis. Intuitive design selection using visualized n-dimensional Pareto frontier. In *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, pages 1813:1–1813:14. American Institute of Aeronautics and Astronautics, 2005.

2 P. J. Agrell. On redundancy in multi criteria decision making. *European Journal of Operations Research*, 98:571–586, 1997.

3 M. J. Alves, C. Henggeler Antunes, and J. Clímaco. Interactive MOLP explorer–A graphical-based computational tool for teaching and decision support in multi-objective linear programming models. *Computer Applications in Engineering Education*, 23(2):314–326, 2015.

4 M. Ankerst, S. Berchtold, and D. A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 52–60. IEEE, 1998.

5 C. Barrico and C. Henggeler Antunes. An evolutionary approach for assessing the degree of robustness of solutions to multi-objective models. *Studies in Computational Intelligence*, 51(2007):565–582, 2007.

6 C. Bazgan, F. Jamain, and D. Vanderpooten. Approximate Pareto sets of minimal size for multi-objective optimization problems. *Operations Research Letters*, 43(1):1–6, 2015.

7 K. Bringmann, T. Friedrich, and P. Klitzke. Two-dimensional subset selection for hypervolume and epsilon-indicator. In *GECCO 14 Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, pages 589–596, 2014.

8 D. Brockoff and E. Zitzler. Objective reduction in evolutionary multiobjective optimization. *Evolutionary Computation*, 17(2):135–166, 2009.

9 M. Brunato and R. Battiti. Grapheur: A software architecture for reactive and interactive optimization. In Christian Blum and Roberto Battiti, editors, *Learning and Intelligent Optimization*, pages 232–246. Springer, 2010.

10 J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Wadsworth, 1983.

11 P.-W. Chiu and C. L. Bloebaum. Hyper-radial visualization (HRV) method with range-based preferences for multi-objective decision making. *Structural and Multidisciplinary Optimization*, 40(1–6):97–115, 2010.

12 K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evolutionary computation*, 14(4):463–494, 2006.

13 P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 23:211–227, 2010.

14 C. M. Fonseca, A. P. Guerreiro, M. López-Ibáñez, and L. Paquete. On the computation of the empirical attainment function. In Ricardo H. C. Takahashi, Kalyanmoy Deb, Elizabeth F. Wanner, and Salvatore Greco, editors, *Proceedings of the 6th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2011*, volume 6576 of *Lecture Notes in Computer Science*, pages 106–120. Springer, 2011.

15 V. Grunert da Fonseca, C. M. Fonseca, and A. O. Hall. Inferential performance assessment of stochastic optimisers and the attainment function. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and D. Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, EMO 2001*, volume 1993 of *Lecture Notes in Computer Science*, pages 213–225. Springer, 2001.

16 J. Hettenhausenay, A. Lewis, and S. Mostaghim. Interactive multi-objective particle swarm optimization with heatmap-visualization-based user interface. *Engineering Optimization*, 42(2):119–139, 2010.

17 P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. DNA visual and analytic data mining. In *Proceedings of the IEEE Conference on Visualization*, pages 437–441. IEEE, 1997.

18 A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and its Applications*. Springer, 2009.

19 Y. Jin and J. Branke. Evolutionary optimization in uncertain environments – A survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317, 2005.

20 T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences. Springer, 2001.

21 P. Korhonen and J. Wallenius. Visualization in the multiple objective decision-making framework. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Słowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 195–212. Springer, 2008.

22 T. Kuhn, C. M. Fonseca, L. Paquete, S. Ruzika, and J. Figueira. Hypervolume subset selection in two dimensions: Formulations and algorithms. Technical report, Technische Universität Kaiserslautern, 2014.

23 A. López Jaimes, C. A. Coello Coello, and D. Chakraborty. Objective reduction using a feature selection technique. In *GECCO 08 Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pages 673–680, 2008.

24 A. Lotov, V. A. Bushenkov, and G. K. Kamenev. *Interactive Decision Maps: Approximation and Visualization of Pareto Frontier*. Kluwer Academic Publishers, 2004.

25 A. Lotov and K. Miettinen. Visualizing the Pareto frontier. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Słowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 213–243. Springer, 2008.

26 M. Luque, F. Ruiz, and K. Miettinen. Global formulation for interactive multiobjective optimization. *OR Spectrum*, 33:27–48, 2011.

27 K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

28 K. Miettinen. IND-NIMBUS for demanding interactive multiobjective optimization. In Tadeusz Trzaskalik, editor, *Multiple Criteria Decision Making '05*, pages 137–150. The Karol Adamiecki University of Economics in Katowice, 2006.

29 K. Miettinen. Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum*, 36(1):3–37, 2014.

30 K. Miettinen and K. Kaario. Comparing graphic and symbolic classification in interactive multiobjective optimization. *Journal of Multi-Criteria Decision Analysis*, 12(6):321–335, 2003.

31 K. Miettinen and M. M. Mäkelä. Interactive multiobjective optimization system WWW-NIMBUS on the Internet. *Computers & Operations Research*, 27:709–723, 2000.

32 K. Miettinen and M.M. Mäkelä. Synchronous Approach in Interactive Multiobjective Optimization. *European Journal of Operational Research*, 170:909–922, 2006.

33 K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to multiobjective optimization: Interactive approaches. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Słowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 27–57. Springer, 2008.

34 G.A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–87, 1956.

35 A. M. Naim, P.-W. Chiu, C. L. Bloebaum, and K. E. Lewis. Decision-making support under uncertainty using preference ranges : The PRUF method. In *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference Conference*, pages 6087:1–6087:12, 2008.

36 V. Ojalehto, K. Miettinen, and T. Laukkanen. Implementation aspects of interactive multiobjective optimization for modeling environments: The case of GAMS-NIMBUS. *Computational Optimization and Applications*, 58(3):757–779, 2014.

37 A. Pryke, S. Mostaghim, and A. Nazemi. Heatmap visualisation of population based multi objective algorithms. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007*, volume 4403 of *Lecture Notes in Computer Science*, pages 361–375. Springer, 2007.

38 R. C. Purshouse and P. J. Fleming. Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In C. M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003*, volume 2632 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 2003.

39 F. Ruiz, M. Luque, and K. Miettinen. Improving the computational efficiency of a global formulation (GLIDE) for interactive multiobjective optimization. *Annals of Operations Research*, 197(1):47–70, 2012.

40 P. Saariluoma, K. Kaario, K. Miettinen, and M. M. Mäkelä. Mental contents in interacting with a multiobjective optimization program. *International Journal of Technology and Human Interaction*, 4(3):43–67, 2008.

41 D.K. Saxena, J.A. Duro, A. Tiwari, and K. Deb. Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *IEEE Transactions on Evolutionary Computation*, 17(1):77–99, 2012.

42 S. Tarkkanen, K. Miettinen, J. Hakanen, and H. Isomäki. Incremental user-interface development for interactive multiobjective optimization. *Expert Systems with Applications*, 40:3220–3232, 2013.

43 T. Tušar and B. Filipič. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 2014. In press, doi:10.1109/TEVC.2014.2313407.

44 T. Tušar and B. Filipič. Visualizing exact and approximated 3D empirical attainment functions. *Mathematical Problems in Engineering*, 2014, 2014. Article ID 569346, 18 pages.

45 T. Ulrich. Pareto-set analysis: Biobjective clustering in decision and objective spaces. *Journal of Multi-Criteria Decision Analysis*, 20(5-6):217–234, 2013.

46 D. Vaz, L. Paquete, C. M. Fonseca, K. Klamroth, and M. Stiglmayr. Representation of the non-dominated set in biobjective combinatorial optimization. Technical Report BUW-IMACM 14/06, Bergische Universität Wuppertal, 2014.

4.5 Multiobjective Optimization for Interwoven Systems (WG5)

Hisao Ishibuchi, Kathrin Klamroth, Sanaz Mostaghim, Boris Naujoks, Silvia Poles, Robin Purshouse, Günter Rudolph, Stefan Ruzika, Serpil Sayın, Margaret M. Wiecek, and Xin Yao

4.5.1 Introduction

Complex systems' optimization is responsive to the demand that computational sciences solve more and more complex problems. A complex system is defined to be a natural or engineered system that is difficult to understand and analyze because it may (1) involve interactions among many phenomena; (2) have multiple and dissimilar components or subsystems that may be connected in a variety of ways and as a whole exhibit one or more properties not obvious from the properties of the individual parts; (3) be characterized by noncomparable and conflicting criteria. Indeed, many entities of interest to humans are complex systems. In the literature, these systems are also referred to as interwoven systems or systems of systems [23]. Natural complex systems such as human body, oceans, climate, and many more have been around since ever and their understanding have been of great significance to people. Energy or telecommunication infrastructures, manufacturing systems, service sector systems are examples of man-made or engineered complex systems. In the modern global world, man-made systems become more and more widespread and omnipresent, and therefore of growing importance to the society.

For complex systems, the overall decision-making goal is to harmonize local requirements and goals to attain the objectives required of the entire system. The overall system performance depends on the interactions and synergy of all its parts, and human preferences are not always captured in the mathematical model. In the presence of multiple components and criteria, a unique decision optimal for the system does usually not exist but rather many or

even infinitely many decisions are suitable. Because of the synergy among the components, the overall system performance is not implied by the simple sum of their performances but is enriched by the synergy among them. Furthermore, when the complex system achieves an "optimal" solution, the system may not have been optimized as a whole because its overall mathematical model may not exist, or if it exists, it makes computations prohibitively expensive. In this case, a solution to the complex system is achieved by optimizing only its components and coordinating their optimal solutions. In effect, due to the features and demands of complex systems, decision-making for them requires tools originated from multiobjective optimization that additionally account for the coupling among the components and the coordination of subsystem optimal solutions into an overall system optimal solution.

Literature on complex systems with multiple criteria is rather limited. The first studies in multiobjective complex problems are undertaken for hierarchical systems in [19, 20, 21, 14, 28, 22] and later continued in [12, 3]. Large-scale hierarchical multiobjective systems are studied in [20, 21, 14]. Other papers propose (i) decomposition of the original problem modeled as one integrated multiple objective problem (MOP) into a collection of smaller-sized sub-problems, for which the development of a solution procedure becomes a more manageable task, and (ii) coordination of the solutions of the sub-problems to obtain the solution of the original problem. A large number of such approaches exists for specific applications in the management sciences, engineering, and multidisciplinary optimization (see [19, 29] among many others). Other papers deal with decomposition and coordination due to a large number of criteria in the original problem [18, 14, 2, 12, 3]. Finally some papers study objective decompositions from a predominantly mathematical perspective [30, 24, 5, 25].

Multidisciplinary design optimization (MDO) had been developed within the engineering community to coordinate results of various disciplines involved in design. The MDO focus has been to either encapsulate disciplinary optimizations into subproblems that are coordinated by a super-optimizer or use sensitivity information to relate the effect of one disciplinary optimization on another. Multiobjective optimization has been introduced to strengthen MDO techniques attempting to deal with noncomparable and conflicting design objectives that are characteristic for each design discipline. Numerous papers present applications of multiobjective MDO in various areas of engineering design, however, formal methodologies such as Multiobjective Collaborative Optimization [29, 27], Multiobjective Concurrent Subspace Optimization (CSSO) [16, 15] and a bilevel method [35] are also proposed.

The discipline-based decomposition of a system, the driving force for MDO, has also been replaced with other types of decomposition such as scenario-based or object-based decomposition, each leading to studying a collection of multiobjective problems. If a system performs in multiple scenarios and each of them is driven by different objective functions, the resulting collection represents a set of multiobjective problems where each of them models the performance of the system in a scenario. Refer to [8, 31, 32] for multiscenario multiobjective optimization in engineering design. An effort to quantify trade-offs between disciplines or scenarios is undertaken in [6, 7]. Physical or object-based decomposition leads to studying a system composed of subsystems and components that can interact with each other in various ways, which additionally increases the complexity of the overall problem. A collection of multiobjective problems naturally emerges because each of the elements may perform according to multiple criteria. Calculation of the Pareto sets of such complex systems is studied in [9, 10, 11].

In this preliminary study, we consider interwoven systems that can be modeled as two interacting subsystems, each modeled as a multiobjective optimization problem (cf. Sec. 4.5.2). The goal is to develop an initial mathematical model of this system and approaches to its optimization. Several examples of such interwoven systems are presented in support of

the proposed modeling paradigm (cf. Sec. 4.5.3). Notions of optimality that recognize the overall system optimality as well as local subsystem optimality are introduced, cf. Sec. 4.5.4. Optimization-based solution approaches are proposed as the composition architectures allowing the computation of the optimal solutions (cf. Sec. 4.5.5). Finally in Sec. 4.5.6, connections of the proposed approach to other areas of optimization and systems science are discussed.

4.5.2 Model

A simple yet non-trivial setup of an interwoven system consists of three parts: two subsystems and the interaction between them. The subsystems come in the form of the following optimization subproblems.

Subproblem 1:

$$\begin{aligned} &\min\ f_1(x_0, x_1, y_{21}) \\ &\text{s.t.}\ \ g_1(x_0, x_1, y_{21}) \leqq 0 \\ &\qquad x_0 \in X_0,\ x_1 \in X_1 \end{aligned}$$

and Subproblem 2:

$$\begin{aligned} &\min\ f_2(x_0, x_2, y_{12}) \\ &\text{s.t.}\ \ g_2(x_0, x_2, y_{12}) \leqq 0 \\ &\qquad x_0 \in X_0,\ x_2 \in X_2 \end{aligned}$$

where $X_i \subseteq \mathbb{R}^{n_i}$ for $i = 0, 1, 2$ and $y_{21} \in \mathbb{R}^{n_{21}}$, $y_{12} \in \mathbb{R}^{n_{12}}$ for some $n_0, n_1, n_2, n_{21}, n_{12} \in \mathbb{N}$. Each subproblem has objective functions $f_i : \mathbb{R}^{n_0} \times \mathbb{R}^{n_i} \times \mathbb{R}^{n_{ji}} \to \mathbb{R}^{p_i}$, and constraint functions $g_i : \mathbb{R}^{n_0} \times \mathbb{R}^{n_i} \times \mathbb{R}^{n_{ji}} \to \mathbb{R}^{q_i}$, $i, j = 1, 2$, $i \neq j$, for some $p_i, q_i \in \mathbb{N}$. Note that Subproblems 1 and 2 share some common decision variables $x_0 \in X_0$ while they also comprise model-specific decision variables x_1 and x_2, respectively.

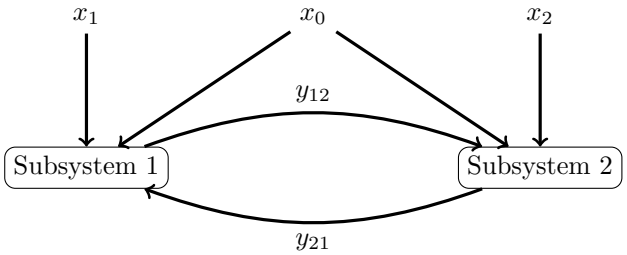
The interaction between the subsystems is modeled with *linking functions* ℓ_1 and ℓ_2 that yield the values of the *linking variables* y_{21} and y_{12}. The interaction is then typically expressed by a system of *interaction equations*:

$$y_{12} = \ell_1(x_0, x_1, y_{21}) \qquad \text{and} \qquad y_{21} = \ell_2(x_0, x_2, y_{12}).$$

where $\ell_i : \mathbb{R}^{n_0} \times \mathbb{R}^{n_i} \times \mathbb{R}^{n_{ji}} \to \mathbb{R}^{r_i}$ for some $r_i \in \mathbb{N}, i, j = 1, 2, i \neq j$. This system of interaction equations has the form of *implicit representation* of linking variables y_{12} and y_{21} by means of linking functions ℓ_1 and ℓ_2. However, an explicit representation of y_{12} and y_{21} of the following form may also exist:

$$\begin{aligned} y_{12} &= y_{12}(x_0, x_1, x_2) \\ y_{21} &= y_{21}(x_0, x_1, x_2) \end{aligned}$$

A graphical exemplification of this setup is given in the following figure:



Feasibility of decision variables may refer to either of the two subsystems or to the interwoven system. This observation motivates the following definitions.

▶ **Definition 1.** A solution (x_0, x_i, y_{ji}) $(i \neq j)$ is said to be *i-subsystem feasible* if $x_0 \in X_0$, $x_i \in X_i$, $g_i(x_0, x_i, y_{ji}) \leqq 0$ and y_{ji} satisfies the interaction equations $y_{ji} = \ell_j(x_0, x_j, y_{ij})$ for some $x_j \in X_j$, where $y_{ij} = \ell_i(x_0, x_i, y_{ji})$.

▶ **Definition 2.** A pair of solutions (x_0, x_1) and (x_0, x_2) is said to be *multisystem feasible* if they are feasible for each system respectively and x_0, x_1, x_2 satisfy the system of interaction equations given by the implicit representation.

In other words, given (x_0, x_1, x_2) such that (x_0, x_1) and (x_0, x_2) are multisystem feasible, the resulting $(x_0, x_1, x_2, y_{21}, y_{12})$ is feasible for the two subproblems.

4.5.3 Examples

An interwoven system consists of interacting subsystems. In some areas of human activity, the susbsystems are developed independently from each other. For example, in engineering design the subsystems of an automotive vehicle such as an engine or a suspension are designed by different companies. Even that each company's designers anticipate that these subsystems will work together within one system (vehicle), the subsystem designs are carried out with limited or even without any information about the future interaction between the subsystems. In other applications, such as location of facilities, subsystems were not even meant to work together when they were being developed but later, due to new circumstances, they necessarily start to interact with each other as an interwoven system.

A countless number of interwoven systems are encountered in daily life and numerous examples can be identified e. g., in traffic systems, multidisciplinary design, or evacuation planning to name just some areas. Nonetheless, some comprehensible examples shall be listed in the following for the sake of intended exemplification of the proposed model.

An Academic Example

Let $X_i = \mathbb{R}$, $i = 0, 1, 2$ and, let $x_i, y_{12}, y_{21} \in \mathbb{R}$, $i = 0, 1, 2$. The scalar-valued objective functions f_1 and f_2 of the subproblems are defined as

$$f_1(x_0, x_1, y_{21}) = x_0^2 + x_1^2 y_{21}$$
$$f_2(x_0, x_2, y_{12}) = (x_0 - 5)^2 + x_2^2 y_{12}$$

The values of the linking variables y_{21} and y_{12} are specified by the following linking functions ℓ_1 and ℓ_2:

$$y_{12} = 2x_0 - 3x_1 + y_{21} = \ell_1(x_0, x_1, y_{21})$$
$$y_{21} = -x_0 + 4x_2 - y_{12} = \ell_2(x_0, x_2, y_{12})$$

For this problem, the following explicit representations can be calculated:

$$y_{21} = -\frac{3}{2}x_0 + \frac{3}{2}x_1 + 2x_2$$
$$y_{12} = \frac{1}{2}x_0 - \frac{3}{2}x_1 + x_2.$$

Integrated Location Problem

Let a finite set of customer locations $A = \{a_1, \ldots, a_M\}$ be given in the plane $\mathbb{R}^2$. Suppose that some group of decision makers, referred to as DM 1, wants to locate an airport at a location $x_1 \in X_1 \subseteq \mathbb{R}^2$. Suppose that for some given weights $w_m^1 \geq 0$, $m = 1, \ldots, M$, the sum of weighted distances between the customers and the airport is to be minimized. Another group of decision makers, say DM 2, wants to locate a hospital at a location $x_2 \in X_2 \subseteq \mathbb{R}^2$ which should (among others) also serve the same set of customers. Given some weights $w_m^1 \geq 0$, $m = 1, \ldots, M$, the maximum weighted distance between the customers and the hospital is to be minimized. The hospital acts as a repulsive facility for the airport (due to noise) which is expressed by some weight $-\lambda_2 < 0$. The airport acts as an attractive facility for the hospital since emergencies occurring at the airport have to reach the hospital quickly. This aspect is modeled by a weight $\lambda_1 > 0$. Staff of the airport and of the hospital will jointly use a service facility, e. g., providing childcare, which has to be located at a location $x_0 \in X_0 \subseteq \mathbb{R}^2$.

The resulting interwoven system can again be specified by identifying the two subproblems corresponding to the two subsystems and by expressing the linking functions.

Subproblem 1: Location of the Airport

$$\begin{aligned}
&\min\ f_{11}(x_0, x_1, y_{21}) = \sum_{m=1}^{M} w_m^1 d(x_1, a_m) - \lambda_2 d(x_1, y_{21}) \\
&\min\ f_{12}(x_0, x_1, y_{21}) = d(x_0, x_1) \\
&\text{s.t.}\ \ x_0 \in X_0,\ x_1 \in X_1
\end{aligned}$$

Subproblem 2: Location of the Hospital

$$\begin{aligned}
&\min\ f_{21}(x_0, x_2, y_{12}) = \max\left\{ \max_{m=1,\ldots,M} w_m^2 d(x_2, a_m);\ \lambda_1 d(x_2, y_{12}) \right\} \\
&\min\ f_{22}(x_0, x_2, y_{12}) = d(x_0, x_2) \\
&\text{s.t.}\ \ x_0 \in X_0,\ x_2 \in X_2
\end{aligned}$$

Interaction Equations. The interaction equations are given by the linking functions ℓ_1 and ℓ_2 as

$$\begin{aligned}
y_{12} &= \ell_1(x_0, x_1, y_{21}) := x_1 \\
y_{21} &= \ell_2(x_0, x_2, y_{12}) := x_2,
\end{aligned}$$

which is again an explicit representation of the linking variables.

Traveling Thief Problem

The traveling thief problem (TTP) consists of two well-known, interacting combinatorial subproblems, the Traveling Salesman Problem (TSP) and the Knapsack Problem (KP). This interaction can be described as follows.

Subproblem 1: Knapsack Problem. A subset of m items numbered $1, \ldots, m$ has to be packed into a knapsack. Each item has a value $b_j \geq 0$ and a weight $w_j \geq 0$, $j = 1, \ldots, m$. The knapsack has a limited capacity Q and it is filled by a thief who wants to maximize the total (additive) value of the items packed while not exceeding the knapsack's capacity. Using binary decision variables $z_1, ..., z_m$, the problem can be modeled as follows:

$$\max f_1(z, b) = \sum_{j=1}^{m} b_j z_j$$
$$\text{s.t. } \sum_{j=1}^{m} w_j z_j \leq Q$$

The solution of the knapsack problem is a binary vector called picked items $z = (z_1, \ldots, z_m)$. Each element z_j, $j \in \{1, \ldots, m\}$ is a binary variable being 1 if the corresponding item is picked and 0 otherwise.

Subproblem 2: Traveling Salesman Problem. The TSP subproblem is one of the classic NP-hard optimization problems. In this problem, there are n cities and the distances between the cities are given by a distance matrix $D = (d_{ij})$ (d_{ij} is the distance between city i and j, $i, j = 1, \ldots, n$). There is a salesman who must visit each city exactly once and minimize the time of the complete tour. While it is usually assumed that the speed v of the salesman is constant throughout every tour, we consider also varying velocities $v(\pi_i)$ of the salesman that depend on the last visited city $\pi_i \in \{1, \ldots, n\}$. Then the TSP subproblem can be formulated as:

$$\min f_2(\pi, v) = \sum_{i=1}^{n} \frac{d(\pi_i, \pi_{i+1})}{v(\pi_i)}$$
$$\text{s.t. } \pi = (\pi_1, \pi_2, \ldots, \pi_n) \in \mathbb{P}_n,$$

where we set $\pi_{n+1} := \pi_1$ to simplify notation. Here, $\mathbb{P}_n$ denotes the set of all permutations of the set $\{1, \ldots, n\}$. The solution of the TSP subproblem is called a tour $\pi = (\pi_1, \cdots, \pi_n)$ where π_i is the i^{th} visited city.

Interaction Equations. In the TTP, there are two objectives, namely maximizing the total value of the knapsack and minimizing the total travel time. We assume that each item is located at one city and that the traveling thief can decide to pick an item or not when visiting the respective city. The more items the thief has picked, the lower his travel speed becomes. In other words, the velocity $v(\pi_i)$ after leaving city π_i depends on the items picked so far. This is modeled using two interconnecting variables:

1. The speed $v(\pi_i)$ of travel when leaving city π_i is related to the knapsack's current weight at city π_i, $i = 1, \ldots, n$:

$$v(\pi_i) = \ell_{1,\pi_i}(z, \pi, b) := v_{\max} - \left(\frac{v_{\max} - v_{\min}}{Q}\right) \sum_{k=1}^{i} \sum_{j=1}^{m} a_j(\pi_k) w_j z_j.$$

The parameter $a_j(\pi_k)$ is equal to 1 if item j is located in city π_k, and zero otherwise. $v_{\max}$ and $v_{\min}$ are the maximum and minimum velocity of the thief, respectively, and Q is the capacity of knapsack. According to this formulation, the speed of the thief decreases when the weight of the knapsack increases, i.e., the speed captures the impact of the KP on the TSP.

2. The value b_j, $j = 1, \ldots, m$, of the picked item j drops over time. In fact, the final value of the item at the end of the travel is not the same as its value when the thief picked the item. This value is dependent on travel time:

$$b_j = \ell_{2,j}(\pi, v) := b_j^{\text{init}} - \rho_j T_j(\pi, v)$$

where $T_j(\pi, v)$ is the time between the moment when item j located at city π_k (i.e., $a_j(\pi_k) = 1$) is picked and the end of the tour:

$$T_j(\pi, v) = \sum_{i=k}^{n} \frac{d(\pi_i, \pi_{i+1})}{v(\pi_i)}.$$

Moreover, ρ_j is a rate of decline in the value of b_j so that $b_j \geq 0$ for all possible values of T_j. The time-dependent value of the items captures the impact of the TSP on the KP.

4.5.4 Notions of Optimality

It is of interest to establish a concept of optimality for the interwoven system presented above. Note that such a concept could recognize all three parts of the system or just a subset of them. We propose three notions of optimality depending on the level of engagement of each subsystem in the overall system.

Assuming that each subsystem would like to perform best to the common good of both subsystems, we define cooperative Pareto solutions that are feasible for both systems.

▶ **Definition 3.** A multisystem feasible solution $(x_0, x_1, x_2, y_{12}, y_{21})$ is said to be *cooperative Pareto optimal* if it is Pareto optimal with respect to all objective functions.

Under the scenario that each subsystem would like to operate at its best for itself regardless of the values of the linking variables passed from the other system, we define individually Pareto-optimal solutions for each system.

▶ **Definition 4.** A solution (x_0, x_1, y_{21}) is said to be *individually Pareto optimal for Subsystem 1* if it can be extended to a multisystem feasible solution $(x_0, x_1, x_2, y_{12}, y_{21})$ and if there is no other multisystem feasible solution $(x'_0, x'_1, x'_2, y'_{12}, y'_{21})$ such that

$$f_1(x'_0, x'_1, y'_{21}) \leq f_1(x_0, x_1, y_{21}). \tag{5}$$

A solution (x_0, x_2, y_{12}) is said to be *individually Pareto optimal for Subsystem 2* if it can be extended to a multisystem feasible solution $(x_0, x_1, x_2, y_{12}, y_{21})$ and if there is no other multisystem feasible solution $(x'_0, x'_1, x'_2, y'_{12}, y'_{21})$ such that

$$f_2(x'_0, x'_2, y'_{12}) \leq f_2(x_0, x_2, y_{12}). \tag{6}$$

The third notion of optimality reflects that both systems might want to perform at their best simultaneously even if one of them could perform better while the other system is ignored.

▶ **Definition 5.** A pair of multisystem feasible solutions (x_0, x_1, y_{21}) and (x_0, x_2, y_{12}) are said to be *mutually Pareto optimal* if there is no other pair of multisystem feasible solutions (x'_0, x'_1, y'_{21}) and (x'_0, x'_2, y'_{12}) such that

$$\begin{pmatrix} f_1(x'_0, x'_1, y'_{21}) \\ f_2(x'_0, x'_2, y'_{12}) \end{pmatrix} \leq \begin{pmatrix} f_1(x_0, x_1, y_{21}) \\ f_2(x_0, x_2, y_{12}) \end{pmatrix} \tag{7}$$

4.5.5 Composition Approaches

We discuss some possible ways of composing the interwoven subsystems.

Biobjective All-in-One System

This approach imposes the least additional structure upon the interwoven system while composing it by bringing together the two subsystems in a natural biobjective way as follows.

$$\begin{aligned}
&\min \begin{pmatrix} f_1(x_0, x_1, y_{21}) \\ f_2(x_0, x_2, y_{12}) \end{pmatrix} \\
&\text{s.t. } g_1(x_0, x_1, y_{21}) \leqq 0 \\
&\quad g_2(x_0, x_2, y_{12}) \leqq 0 \\
&\quad x_0 \in X_0,\ x_1 \in X_1,\ x_2 \in X_2
\end{aligned}$$

where $y_{12} = \ell_1(x_0, x_1, y_{21})$ and $y_{21} = \ell_2(x_0, x_2, y_{12})$.

The term biobjective is used in relation to the two subsystems involved. Note that if f_1 or f_2 is a vector-valued function, the number of objectives in the above formulation will be more than two. Therefore, in general, this is a multiobjective optimization formulation. The Pareto-optimal solutions to this multiobjective problem can be considered as the solutions to the interwoven system.

As an example, consider again the academic example introduced in Section 4.5.3. The corresponding biobjective all-in-one system is in this case given by

$$\begin{aligned}
&\min f_1(x_0, x_1, y_{21}) = x_0^2 + x_1^2 y_{21} \\
&\min f_2(x_0, x_2, y_{12}) = (x_0 - 5)^2 + x_2^2 y_{12} \\
&\text{s.t. } y_{21} = -\frac{3}{2}x_0 + \frac{3}{2}x_1 + 2x_2 \\
&\quad y_{12} = \frac{1}{2}x_0 - \frac{3}{2}x_1 + x_2.
\end{aligned}$$

An approximation of the nondominated set of this all-in-one system is illustrated in Figure 8. The points shown are obtained by sampling feasible solutions and filtering for dominated points.

Bilevel All-in-One System

In some situations, the interactions between the two subsystems may be modeled in a hierarchical way. In such cases, a bilevel programming framework may best describe the composed system. Such a composition does not need to utilize the variables x_0.

$$\begin{aligned}
&\min f_1(x_0, x_1, y_{21}) \\
&\text{s.t. } g_1(x_0, x_1, y_{21}) \leqq 0 \\
&\quad y_{21} = \ell_2(x_0, x_2, y_{12}) \\
&\quad x_0 \in X_0,\ x_1 \in X_1 \\
&\quad x_2 \in \arg\min f_2(x_0, x_2, y_{12}) \\
&\qquad \text{s.t. } g_2(x_0, x_2, y_{12}) \leqq 0 \\
&\qquad\quad y_{12} = \ell_1(x_0, x_1, y_{21}) \\
&\qquad\quad x_2 \in X_2.
\end{aligned}$$

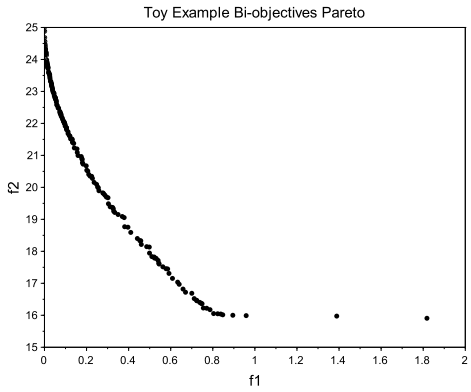


Figure 8 Approximation of the images of the cooperative Pareto solutions of the all-in-one system for the academic example introduced in Section 4.5.3.

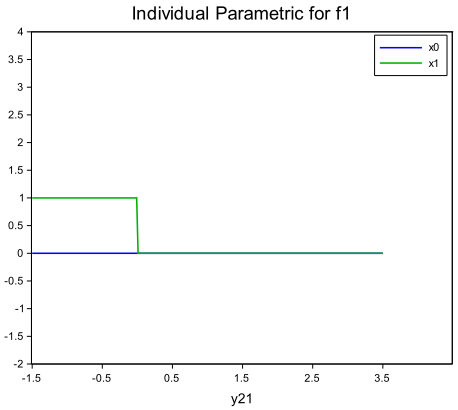


Figure 9 Dependence of the upper level variables x_0, x_1 on the output y_{21} of the lower level problem for the academic example introduced in Section 4.5.3.

In this bilevel problem the objective functions f_1 and f_2 can be scalar and/or vector-valued. Solutions to the interwoven system are solutions to this (possibly multiobjective) bilevel programming problem. The optimal (Pareto-optimal) solutions to this bilevel problem can be considered as optimal (Pareto-optimal) solutions to the interwoven system.

Considering again the academic example problem introduced in Section 4.5.3, Figure 9 shows the dependence of the upper level variables x_0 and x_1 from the value of the linking variable y_{21} that directly depends on the optimal solution x_2 of the lower level problem.

Individual Systems with Parameterized Interactions

The two subsystems may be decoupled by letting each subsystem treat the linking variables as parameters. The idea can be expressed as follows.

$$\min_{x_0,x_1} f_1(x_0, x_1; y_{21})$$

s.t. $g_1(x_0, x_1; y_{21})$ where $y_{21} \in [t_L, t_R]$ is a parameter.

$$\min_{x_0,x_2} f_2(x_0, x_2; y_{12})$$

s.t. $g_2(x_0, x_2; y_{12})$ where $y_{12} \in [u_L, u_R]$ is a parameter.

It is anticipated that best solutions to the two subsystems will be found by solving the subproblems independently. However, since the two subsystems must agree on the linking variables y_{21} and y_{12} and the common variable x_0, a solution mechanism that ensures such a conversion must be employed. This can lead to quite different implementations of the individual systems approach. Below, we describe two possibilities.

Analytical Target Cascading. Analytical target cascading (ATC) is a hierarchical, multilevel multidisciplinary methodology to optimize complex engineering design problems, see, for example, [17]. A system is hierarchically decomposed into individual design problems at each level, possibly in multiple subproblems. Once a higher-level problem is solved, solutions are propagated (cascaded) as targets to the lower-level and then solved at that level. The new solutions (responses) are in turn passed back up to the higher level. The solution process continues iteratively until solutions at every level are within a tolerance level of or as close as possible to the desired targets.

The approach is described in Algorithm 1 below in the form of a pseudocode for the simpler case of the interwoven system without the global variable x_0. However, more general cases can also be addressed with ATC. In the presented case, the linking variable y_{12} is treated as a target being sent down from subproblem 1 to subproblem 2, while the other linking variable y_{21} acts as a response to be send up from subproblem 2 to suproblem 1. In particular implementations, acceptable stopping criteria must be specified.

Algorithm 1 ATC for two subproblems without global variables

Initialize y_{21}^0
$k \leftarrow 0$
repeat
  $k \leftarrow k+1$
  $x_1^k \leftarrow \arg\min_{x_1}\{f_1(x_1, y_{21}^{k-1}) | g_1(x_1, y_{21}^{k-1}) \leqq 0\}$
  $y_{12}^{k-1} \leftarrow \ell_1(x_1^k, y_{21}^{k-1})$
  $x_2^k \leftarrow \arg\min_{x_2} f_2(x_2, y_{12}^{k-1}) | g_2(x_2, y_{12}^{k-1}) \leqq 0\}$
  $y_{21}^k \leftarrow \ell_2(x_2^k, y_{12}^{k-1})$
until solution (x_1^k, x_2^k) is of acceptable quality

Bayesian Updates. The parameters in the individual systems can be treated as uncertain, although more accurately, these are “presently undetermined" for the particular subsystem in consideration. Since as in the above case, this is true for both subsystems, an iterative optimization mechanism must be employed to converge to a joint solution. At the start of the optimization process, subsystem 1 would formulate a prior distribution for the uncertainty in y_{21} – this could be uninformative or based on some initial estimates coming from subsystem 2. Subsystem 2 does the same. Then either hierarchically or simultaneously, some information sharing that leads to belief updates takes place. The iterative process is repeated until an acceptable solution is obtained.

4.5.6 Connection to Other Disciplines

Game Theory

The structure of interwoven systems can encompass game theoretic models by modeling subsystem behavior. This can be achieved by incorporating anticipation of other subsystem's

responses into the objective function or the value function. It is also possible to incorporate different information sharing strategies of subsystems via linking variables. When subsystems exhibit some hierarchical structure as modeled in the bilevel formulation introduced in Section 4.5.5 (Bilevel All-in-One System), a Stackelberg game is relevant. When all subsystems are considered as simultaneous players, a Nash game may be implied [13, 33].

Robust Design

The structure defined in Section 4.5.5 (Bayesian Updates) can relate to robust optimization if a robustness-related metric is chosen as the value function in the sense that each subsystem's individual solution can be regarded as a robust solution where the interaction with the other subsystem is modeled as uncertain. In particular, each subsystem of an interwoven system could treat the interaction variable as an *uncertain* parameter although the interaction variables are *undetermined* rather than uncertain. This concept is embodied in the notion of *Type II Robust Design* [4] when interpreted in the context of coupled variables in a distributed problem [1].

Co-Evolutionary Algorithms

Studies in co-evolutionary computation investigate how separate subpopulations solve their own subproblems as a means of solving the complete problem – an approach known as *c*ooperative-co-evolution [26]. The sub-populations exchange their information at certain intervals, e.g., at a certain number of generations in an evolutionary algorithm. Although subpopulations try to optimize their own objectives, they have to cooperate to solve the overall problem [34].

4.5.7 Summary and Outlook

In this report a mathematical model for an interwoven system consisting of two subproblems was introduced. Different concepts defining the optimal performance of such an interwoven system were proposed, and the relation to associated multiobjective and bilevel optimization models was discussed. Several existing optimization methodologies were suggested as tools for generating optimal solutions to interwoven systems.

This research raises a variety of challenging and interesting questions. This includes generalizations to interwoven systems with more than two subproblems, an in-depth analysis of the similarities and the differences between different notions of optimality and between the associated optimization models, and the development and critical evaluation of efficient solution methods. The example problems mentioned in this report may serve as a first benchmark for such approaches.

References

1 J. K. Allen, C. Seepersad, H. J. Choi, and F. Mistree. Robust design for multiscale and multidisciplinary applications. *Journal of Mechanical Design, Transactions of the ASME*, 128(4):832–843, 2006.

2 H. P. Benson and E. Sun. Outcome space partition of the weight set in multiobjective linear programming. *J. Optim. Theory Appl.*, 105(1):17–36, 2000.

3 R. Caballero, T. Gomez, M. Luque, Miguel F., and R. Ruiz. Hierarchical generation of Pareto optimal solutions in large-scale multiobjective systems. *Computers and Operations Research*, 29:1537–1558, 2002.

4 W. Chen, J.K. Allen, K.-L. Tsui, and F. Mistree. A procedure for robust design: Minimizing variations caused by noise factors and control factors. *Journal of Mechanical Design, Transactions of the ASME*, 118(4):478–485, 1996.

5 M. Ehrgott and S. Nickel. On the number of criteria needed to decide Pareto optimality. *Math. Methods Oper. Res.*, 55(3):329–345, 2002.

6 A. Engau and M. M. Wiecek. 2D decision making for multi-criteria design optimization. *Structural and Multidisciplinary Optimization*, 34(4):301–315, 2007.

7 A. Engau and M. M. Wiecek. Interactive coordination of objective decompositions in multiobjective programming. *Journal of Management Science*, 54(7):1350–1363, 2008.

8 G. Fadel, I. Haque, V. Blouin, and M. M. Wiecek. Multi-criteria multi-scenario approaches in the design of vehicles. In *Proceedings of 6th World Congresses of Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, 2005.

9 M. Gardenghi. *Multiobjective Optimization for Complex Systems*. PhD thesis, Clemson University, Clemson, SC, 2009.

10 M. Gardenghi, F. Miquel, T. G. Nunez, and M. M. Wiecek. Algebra of efficient sets for complex systems. *Journal of Optimization Theory and Applications*, 149:385–410, 2011.

11 M. Gardenghi and M. M. Wiecek. Efficiency for multiobjective multidisciplinary optimization problems with quasi-separable subsystems. *Optimization and Engineering*, 13(2):293–318, 2012.

12 T. Gomez, M. Gonzalez, M. Luque, F.. Miguel, and F. Ruiz. Multiple objectives decomposition-coordination methods for hierarchical organizations. *Eur. J. Oper. Res.*, 133(2):323–341, 2001.

13 A. Habbal, J. Petersson, and M. Thellner. Multidisciplinary topology optimization solved as a Nash game. *International Journal for Numerical Methods in Engineering*, 61(7):949–963, 2004.

14 Y. Y. Haimes, K. Tarvainen, T. Shima, and J. Thadathil. *Hierarchical Multiobjective Analysis of Large-Scale Systems*. Hemisphere Publishing Corp., New York, 1990.

15 C.-H. Huang, J. Galuski, and C. L. Bloebaum. Multi-objective Pareto concurrent subspace optimization for multidisciplinary design. *AIAA Journal*, 45(8):1894–1906, 2007.

16 Ch.-H. Huang. *Development of Multi-objective Concurrent Subspace Optimization and Visualization Methods for Multidisciplinary Design*. PhD thesis, State University of New York at Buffalo, Buffalo, NY, 2003.

17 H. M. Kim, D. G. Rideout, P. Y. Papalambros, and J. L. Stein. Analytical target cascading in automotive vehicle design. *Journal of Mechanical Design*, 125:481–489, 2003.

18 R. Lazimy. Solving multiple criteria problems by interactive decomposition. *Mathematical Programming*, 35(3):334–361, 1986.

19 S. M. Lee and B. H. Rho. Multicriteria decomposition model for two-level, decentralized organizations. *International Journal on Policy and Information*, 9(1):119–133, 1985.

20 D. Li and Y. Y. Haimes. Hierarchical generating method for large-scale multiobjective systems. *Journal of Optimization Theory and Applications*, 54(2):303–333, 1987.

21 D. Li and Y. Y. Haimes. Multilevel methodology for a class of non-separable optimization problems. *International Journal of Systems Sciences*, 21(11):2351–2360, 1990.

22 E. R. Lieberman. *Multi-Objective Programming in the USSR*. Academic Press, Inc., Boston, 1991.

23 M. Maier. Architecting principles for systems-of-systems. *Systems Engineering*, 1:267–284, 1998.

24 Ch. Malivert and N. Boissard. Structure of efficient sets for strictly quasi-convex objectives. *J. Convex Anal.*, 1(2):143–150, 1995.

25 N. Popovici. Pareto reducible multicriteria optimization problems. *Optimization*, 54(3):253–263, 2005.

26 M. A. Potter and K. A. De Jong. Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary computation*, 8(1):1–29, 2000.

27 S. Rabeau, P. Dépincé, and F. Bennis. Collaborative optimization of complex systems: a multidisciplinary approach. *International Journal on Interactive Design and Manufacturing*, 1:209–218, 2007.

28 T. Tanino and H. Satomi. Optimization methods for two-level multiobjective problems. In A. Lewandowski and V. Volkovich, editors, *Multiobjective Problems of Mathematical Programming*, Proceedings of the International Conference on Multiobjective Problems of Mathematical Programming held in Yalta, USSR, 1988, pages 128–137, Berlin, 1991. Springer.

29 R. V. Tappeta and J. E. Renaud. Multiobjective collaborative optimization. *Journal of Mechanical Design*, 119(3):403–411, 1997.

30 J. Ward. Structure of efficient sets for convex objectives. *Math. Oper. Res.*, 14(2):249–257, 1989.

31 M. M. Wiecek. Multi-scenario multi-objective optimization for engineering design. In K. Deb, P. Chakroborty, N. G. R. Iyengar, and Gupta S. K., editors, *Advances in Computational Optimization and its Applications*, pages 170–174, India, 2007. Universities Press.

32 M. M. Wiecek, V. Y. Blouin, G. M. Fadel, A. Engau, B. J. Hunt, and V. Singh. Multi-scenario multi-objective optimization with applications in engineering design. In V. Barichard, M. Ehrgott, X. Gandibleux, and V. T'Kindt, editors, *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, volume 618 of *Lecture Notes in Economics and Mathematical Systems*, pages 283–298, Berlin, 2009. Springer.

33 M. Xiao, X. Shao, L. Gao, and Z. Luo. A new methodology for multi-objective multidisciplinary design optimization problems based on game theory. *Expert Systems with Applications*, 42(3):1602–1612, 2015.

34 Z. Yang, K. Tang, and X. Yao. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178(15):2985–2999, 2008.

35 K.-S. Zhang, Z.-H. Han, W.-J. Li, and W.-P. Song. Bilevel adaptive weighted sum method for multidisciplinary multi-objective optimization. *AIAA Journal*, 46(10):2611–2622, 2008.

4.6 Surrogate-Assisted Multicriteria Optimization (WG6)

Richard Allmendinger, Carlos A. Coello Coello, Michael T. M. Emmerich, Jussi Hakanen, Yaochu Jin, and Enrico Rigoni

4.6.1 Introduction

In real-world optimization it is very common to use either physical experimentation or simulators to evaluate solutions (see e.g. [39, 22, 43]). Such evaluation procedures can be costly and time-consuming, and there only a limited budget of evaluations is available. Surrogate-assisted optimization [23] (sometimes also referred to as metamodel-assisted optimization) is a common technique for solving such problems. There are many (specific) research questions that arise when studying this methodology, some of them specific to surrogate-model assisted multiobjective optimization. In this report we summarize sources of

complexity and challenges to be met in this field and then discuss recently proposed solutions or prospective solution ideas on how to analyze complexity and how to deal with it.

The report is divided into two main sections: First, Section 4.6.2 discusses challenges and sources of complexity, distinguishing between problems specific to multicriteria surrogate-assisted problems and challenges that are inherited from more general optimization problem settings. Then, Section 4.6.3 proposes some initial ideas on how to meet open challenges related to problem complexity in this research field.

4.6.2 Challenges and Sources of Complexity

In this section, we review common and emerging challenges and sources for complexity. We divide them into two categories: challenges specific to optimization in general and challenges specifically relevant to surrogate-assisted optimization. In both of these categories, there are challenges that are relevant for multiobjective optimization and we will highlight them where applicable.

General challenges in optimization

The following challenges and sources of complexity can be encountered generally in optimization and are not specific to surrogate-assisted optimization. However, these issues can become even more relevant when using surrogates.

- **Functional landscape.** The structure of the fitness landscape has a huge impact on optimization. Examples of challenging landscape features include nonlinearity of objective functions, discontinuity in the objective space, and multimodal (deceptive) functions [25].
- **Decision variables and constraints.** Typical challenges in solving optimization problems include a large number of decision variables as well as a large number of constraints [9]. Recently, problems with dynamic constraints [5] and changing decision variables [3] arose, particularly in the experimental optimization community. Furthermore, problems with mixed-integer variables pose a general challenge in evolutionary optimization [30, 37, 19].
- **Objectives.** In addition, in multiobjective optimization, a high number of objective functions [16, 21] and heterogeneous functions (see e.g. [4, 2]) can provide additional challenges for both the algorithms and post-processing or decision making. Dynamically changing objective functions can increase the complexity further [8, 36].
- **Noise and uncertainty.** Noise and uncertainty are byproducts that are common in simulation-based and experimental optimization, and appropriate methods are needed for solving such problems [24]. Uncertainty can exist both in the decision space and the objective space. An example of noise are measurement errors typically present in experimental optimization [43], whilst imprecise knowledge about the model used in simulation-based optimization would represent a classical example of uncertainty [16].
- **Optimal use of many-core computers.** Recent problems in simulation-based optimization may feature time consuming objective function evaluations (simulations). Those problems can be solved by using general optimization algorithms, but often, algorithms tailored for such problems are needed if, for example, there exist a time limitation. One approach could be to utilize parallelization of the algorithm or the function evaluations [1].

Challenges specific to surrogate-assisted optimization

The following challenges and sources of complexity are directly related to approaches where surrogates are utilized and, thus, are not encountered generally in optimization.

- **Training time.** An important aspect in surrogate-assisted optimization is how much time it takes to train the metamodels used. If training takes too long, then it can significantly reduce the time saved by metamodelling. For example, if the data used in training is large, then matrix inversion needed in some metamodels could be time consuming [27].
- **Metamodel selection.** Many different types of metamodels have been developed over the years and it is not a straightforward task to choose the best one for the problem in question. One approach of overcoming this difficulty is to utilize multiple metamodels or ensembles of metamodels where the best metamodel can be dynamically selected during the optimization run, similar to [17, 29, 52]. Sometimes in multiobjective optimization different metamodels have to be used for different objective functions due to e.g. complexity of the objectives [48].
- **Surrogates for the Pareto front.** It is also possible to use surrogates for the Pareto front instead of the individual objective functions [14, 18, 34]. In that case, the input for training the surrogate is a (small) set of precomputed Pareto optimal solutions. The resulting surrogate can then be used for example for fast decision making with interactive multiobjective optimization methods. Typically in multiobjective optimization the number of objective functions is smaller than the number of decision variables and, therefore, building a surrogate for the Pareto front can be beneficial.
- **Discrete search spaces.** While applications with both discrete and continuous decision variables feature a general challenge to optimization, their presence can become a serious issue for surrogate-assisted optimization. Recent work aiming at overcoming these issues can be found in [31, 6, 47, 35].
- **Multi-fidelity models.** An approach to solve optimization problems with time consuming objective function evaluations is to use a collection of (meta)models that have different fidelity. In these approaches, one has to identify which (meta)model to use in which phase of the solution process. Controlling the model fidelity can be made dynamic by having an automated way of managing this at runtime [32].
- **Additional measurements/outputs.** Simulation-based and experimental optimization can produce a large amount of data although only a tiny fraction of it is utilized to compute the objective function values. It is an open question whether the remaining data can be utilized meaningfully to enhance search [22].

4.6.3 Prospective Solutions

Within the discussion of the workshop some interesting directions for prospective solutions were identified. They will be elaborated in the following.

Model learning for different objective and constraint functions

As outlined in the previous section, different objective and constraint functions can have different characteristics, such as computational effort, types of nonlinearity, e.g. multimodality and discontinuities, and noise. In this context we would like to point to the fact that such heterogeneity in multi-objective optimization is an emerging research topic in itself, discussed in another workgroup of the Dagstuhl Seminar, and here we will limit our discussion only to aspects relevant to surrogate models.

In [40] an automatic procedure for improving the accuracy of metamodels in an adaptive and iterative way is implemented. During the optimization process different modelling techniques are competing for modelling each single function. The performance assessment of metamodels is done independently for the different objective and constraint functions. Also,

the evaluation takes place repeatedly during the run. In every iteration it is decided anew which model type is the best one to use for modeling a function. The last run's performance is decisive in this approach: Basically, the winning model on the data points evaluated in the last round will perform surrogate based optimization in the next round. Only, if one model becomes dominant in multiple runs it is taking over the task without further considering the other models (to save computation time).

This idea can be further elaborated by considering different online update schemes of the model-function assignment. An idea that seems to be straightforward in the machine learning context would be to use reinforcement learning [46] here, in order to learn by reward and punishment gradually the frequency of models to be used. It is known that reinforcement is robust, but adapts the frequencies relatively slow. This is, why we render this strategy to be promising only if the budget of function evaluation is moderate (say $\gg 100$) and not very small. A variation of the reinforcement paradigm that seems to lend itself well to online model selection is the multi-armed bandit paradigm [13], which has recently been used in operator selection for multi-criteria optimization. The reward function could take into account the achieved improvement (for instance in (hypervolume) set-performance indicators) or in average errors (model improvement).

Fast linear algebra techniques for large point sets

One of the main challenges specific to metamodeling is that the cost for training metamodels, in particular Gaussian processes (or Kriging) and to a smaller extend Radial Basis Function networks, becomes prohibitively high when the number of training instances (evaluated design points) becomes large.

Recall, that the computational cost of commonly used metamodels is related to the time required to invert the matrix of correlations based on the pairwise distances between design point. Therefore, the size of the matrix grows quadratically with the number of design points.

A solution to this problem that is often proposed is to use *fast approximate matrix inversion*. Although there are efficient algorithms for approximate matrix inversion available in the literature, they are to our knowledge not widely used in the surrogate-assisted optimization community. An interesting research topic would therefore be to compare these techniques in the context of surrogate-assisted optimization. As a first step in this direction we looked in the literature for some relevant techniques and overview papers.

The problem of approximate matrix inversion has been studied since the 70ties in applied mathematics [15], and has received recently increased attention in the machine learning research community. A good survey paper for approximate techniques for matrix inversion in the context of Gaussian processes is [38]. A state of the art method, that was implemented recently in mathematical packages is called Fully Independent Training Conditional (FITC), originally called Sparse Gaussian Processes using of Pseudo-Inputs (SGPP) [44]. These methods make use of the positive definiteness of the correlation matrix. Moreover, they select a relevant subset of the training points and perform the matrix inversion only on the submatrix for these point, while the other points still contribute to the computation of the final result. However, the selection of a subset of points is still based on simple heuristic and it will be interesting to investigate this deeper.

Another technique that is already used in metamodel-assisted evolutionary computation is called *local metamodelling*, where, as stated in [26], 'models are trained separately for each new population member on its closest data among the previously evaluated solutions'. Here the term population members refers to new candidate design points. This method has the advantage of smaller training time, but also to provide metamodels that are more based on

the regional characteristic of the response surface rather than on its global structure. This is of particular importance if there is non-stationarity and hyperparameters that lead to good performance in one region but do not perform well in other regions.

A problem that occurs in this context is that discontinuities arise, when the set of nearest neighbors changes, causing problems for gradient-based optimization methods that require smooth surfaces. Moreover, artifacts such as local optima might be created – although this has hardly been studied up to now. In addition to this, if only the nearest neighbors are considered, clustering of sets might lead to ill-conditioned matrices or introduce a bias (e.g. considering points in one direction only). An interesting technique could be to use an adaptive archiving technique, similar to those proposed in [28] in the context of global robust optimization. The idea is to generate a design of experiments, for instance a Latin Hypercube Design [7], around the new design point and collect a nearest neighbors for each design point from the database. If there is no near neighbor to one of the design points, then a new evaluation is scheduled at this point. This strategy is a variant of an *active learning* approach [10], but more targeted towards the needs of optimization. In the context of multi-criteria optimization the amount of information and the radius of the design of experiments should be based on the characteristics of the function, which in first approximation can be derived from the hyperparameters of the model (for instance the estimated auto-correlation(s) and variances in Kriging/Gaussian process models). Already in the classical book on spatial statistics by Cressie [12] some advise for the radius in which relevant training points can be found was given, albeit for rather low dimensional data sets (2, 3 dimensions).

Exploiting dependences between objective and constraint functions

Nowadays, the common approach to use metamodels in multicriteria optimization is to train independent models for each objective and (implicit) constraint function. This makes computations simpler, for instance to compute multiobjective expected improvement [11, 42, 49], but on the other hand these models cannot exploit the possible correlation between different response variables. Hence, there are two difficulties that arise when using dependence information and we will briefly describe which techniques look promising in order to meet them:

Firstly, the computation of metamodels needs to be adapted. In the statistical community, it was dealt with using a technique called multi-output nonparametric regression [33]. More specifically, in the context of Kriging metamodels, it has been recently discussed under the term *multi-response metamodels* [41]. The idea in both approaches is to exploit the covariance between output variables (which could be objective function values or constraint function values). Also the computation of metamodel indicators will become more difficult.

Secondly, in order to compute measures, such as expected improvement, based on multivariate response formula, exact computation schemes [20] need to be modified. The block decomposition schemes right now need to be adapted by computing truncated multivariate Gaussian distributions. Recently, a package on truncated multivariate Gaussian distributions became available [50], which could be a good starting point in this direction.

Creating a benchmark

A recommendation for well referenced test problems was recently released by Surjanovic and Bingham [45]. It is available under the link http://www.sfu.ca/~ssurjano and contains a representative set of popular benchmark problems, including the Branin function, which

became a standard test problem in surrogate-assisted optimization. However, a similar benchmark specific for simulator-based *multi-objective* optimization is to our knowledge still missing so far.

Metamodels for mixed-integer and combinatorial optimization

A first approach to use metamodels in mixed-integer and discrete parameter optimization is described in Li et al. [31]. It uses a heterogeneous metric that was developed for radial-basis function neural networks [51]. In the context of combinatorial optimization and permutations a comparison of distance measures was recently conducted by Zaefferer et al. [53]. Although using distances is an approach that works on more parametric problems, it could be interesting to look at machine learning approaches that can model discrete decision variables in a more problem specific way. Often the meaning and impact of a discrete decision variable can be estimated a-priori (e.g. switching on and off a process alternative in a flowsheet). In such cases modeling a problem specific graph metric could be a promising direction, e.g. by defining a transition graph and computing path distances in it. Also, as opposed to neural networks, the theory of Gaussian processes is more heavily based on the assumption of learning continuous functions. In this case we suggest to instead consider Markov random field models, when it comes to combinatorial search spaces. These also model local correlations, but are more natural to the problem and by introducing edge weights (transition probabilities) a neighborhood in terms of design point similarity can be modeled in a more intuitive manner. An open question, to our knowledge, is however to generalize the theory of Gaussian processes to mixed-integer spaces and fundamental research needs to be done in this direction.

References

1 E. Alba, G. Luque, and S. Nesmachnow. Parallel metaheuristics: recent advances and new trends. *International Transactions in Operational Research*, 20(1):1–48, 2013.

2 R. Allmendinger, J. Handl, and J. D. Knowles. Multiobjective optimization: When objectives exhibit non-uniform latencies. *European Journal of Operational Research*, 243(2):497–513.

3 R. Allmendinger and J. D. Knowles. Evolutionary optimization on problems subject to changes of variables. In *Parallel Problem Solving from Nature, PPSN XI*, pages 151–160. Springer, 2010.

4 R. Allmendinger and J. D. Knowles. 'hang on a minute': Investigations on the effects of delayed objective functions in multiobjective optimization. In *Evolutionary Multi-Criterion Optimization*, pages 6–20. Springer, 2013.

5 R. Allmendinger and J. D. Knowles. On handling ephemeral resource constraints in evolutionary search. *Evolutionary computation*, 21(3):497–531, 2013.

6 L. Bajer and M. Holeňa. Surrogate model for continuous and discrete genetic optimization based on rbf networks. In *Intelligent Data Engineering and Automated Learning–IDEAL 2010*, pages 251–258. Springer, 2010.

7 G. E. P. Box, J. S. Hunter, and W. G. Hunter. *Statistics for experimenters: design, innovation, and discovery*. John Wiley, 2005.

8 J. Branke. *Evolutionary optimization in dynamic environments*. Kluwer academic publishers, 2001.

9 C. A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, 191(11):1245–1287, 2002.

10 D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

11 I. Couckuyt, D. Deschrijver, and T. Dhaene. Fast calculation of multiobjective probability of improvement and expected improvement criteria for pareto optimization. *Journal of Global Optimization*, pages 1–20, 2013.

12 N. Cressie. *Statistics for Spatial Data: Wiley Series in Probability and Statistics*. Wiley, 1993.

13 M. M. Drugan and A. Nowe. Designing multi-objective multi-armed bandits algorithms: a study. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013.

14 P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32:211–227, 2010.

15 R. B. Flavell. Approximate matrix inversion. *Operational Research Quarterly*, pages 517–520, 1977.

16 P. J. Fleming, R. C. Purshouse, and R. J. Lygoe. Many-objective optimization: An engineering design perspective. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, volume 3410 of *Lecture Notes in Computer Science*, pages 14–32. Springer Berlin Heidelberg, 2005.

17 D. Gorissen, T. Dhaene, and F. De Turck. Evolutionary model type selection for global surrogate modeling. *The Journal of Machine Learning Research*, 10:2039–2078, 2009.

18 M. Hartikainen, K. Miettinen, and M. M. Wiecek. PAINT: Pareto front interpolation for nonlinear multiobjective optimization. *Computational Optimization and Applications*, 52(3):845–867, 2012.

19 M. Holena, T. Cukic, U. Rodemerck, and D. Linke. Optimization of catalysts using specific, description-based genetic algorithms. *Journal of chemical information and modeling*, 48(2):274–282, 2008.

20 I. Hupkens, A. Deutz, K. Yang, and M. T. M. Emmerich. Faster exact algorithms for computing expected hypervolume improvement. In *Evolutionary Multi-Criterion Optimization, Proc. of Int. Conf. on*. Springer (accepted for), 2015.

21 H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation*, pages 2419–2426, 2008.

22 J. Jakumeit and M. T. M. Emmerich. Optimization of gas turbine blade casting using evolution strategies and kriking. In B. Filipic and J. Silc, editors, *Proceedings of BIOMA 2004, the International Conference on Bioinspired Optimization Methods and their Applications*, pages 95–104. Kluwer Academic Publishers, 2004.

23 Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

24 Y. Jin and J. Branke. Evolutionary optimization in uncertain environments-a survey. *Evolutionary Computation, IEEE Transactions on*, 9(3):303–317, 2005.

25 L. Kallel, B. Naudts, and C. R. Reeves. Properties of fitness functions and search landscapes. In *Theoretical aspects of evolutionary computing*, pages 175–206. Springer, 2001.

26 I. C. Kampolis and K. C. Giannakoglou. A multilevel approach to single-and multiobjective aerodynamic optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33):2963–2975, 2008.

27 J. D. Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *Evolutionary Computation, IEEE Transactions on*, 10(1):50–66, 2006.

28 J. Kruisselbrink, M. T. M. Emmerich, and T. Bäck. An archive maintenance scheme for finding robust solutions. In *Parallel Problem Solving from Nature, PPSN XI*, pages 214–223. Springer, 2010.

29 M. N. Le, Y.-S. Ong, S. Menzel, Y. Jin, and B. Sendhoff. Evolution by adapting surrogates. *Evol. Comput.*, 21(2):313–340, May 2013.

30 R. Li, M. T. M. Emmerich, J. Eggermont, T. Bäck, M. Schütz, J. Dijkstra, and J. H. C. Reiber. Mixed integer evolution strategies for parameter optimization. *Evolutionary computation*, 21(1):29–64, 2013.

31 R. Li, M. T. M. Emmerich, J. Eggermont, E.G.P. Bovenkamp, T. Bäck, J. Dijkstra, and J. Reiber. Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2764–2771. IEEE, 2008.

32 D. Lim, Y.-S. Ong, Y. Jin, and B. Sendhoff. Evolutionary optimization with dynamic fidelity computational models. In De-Shuang Huang, II Wunsch, Donald C., Daniel S. Levine, and Kang-Hyun Jo, editors, *Lecture Notes in Computer Science*, volume 5227, pages 235–242. Springer Berlin Heidelberg, 2008.

33 J. M. Matías. Multi-output nonparametric regression. In *Progress in artificial intelligence*, pages 288–292. Springer, 2005.

34 M. Monz, K. H. Kufer, T. R. Bortfeld, and C. Thieke. Pareto navigation – algorithmic foundation of interactive multi-criteria IMRT planning. *Physics in Medicine and Biology*, 53(4):985–998, 2008.

35 A. Moraglio and A. Kattan. Geometric generalisation of surrogate model based optimisation to combinatorial spaces. In *Evolutionary Computation in Combinatorial Optimization*, pages 142–154. Springer, 2011.

36 T. T. Nguyen, S. Yang, and J. Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24, 2012.

37 M. Olhofer, B. Sendhoff, T. Arima, and T. Sonoda. Optimisation of a stator blade used in a transonic compressor cascade with evolution strategies. In *Evolutionary Design and Manufacture*, pages 45–54. Springer, 2000.

38 J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

39 I. Rechenberg. Case studies in evolutionary experimentation and computation. *Computer Methods in Applied Mechanics and Engineering*, 2-4(186):125–140, 2000.

40 E. Rigoni and A. Turco. Metamodels for fast multi-objective optimization: trading off global exploration and local exploitation. In *Simulated Evolution and Learning*, pages 523–532. Springer, 2010.

41 D. A. Romero. *A multi-stage, multi-response Bayesian methodology for surrogate modeling in engineering design*. ProQuest, 2008.

42 K. Shimoyama, K. Sato, S. Jeong, and S. Obayashi. Comparison of the criteria for updating kriging response surface models in multi-objective optimization. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE, 2012.

43 B. G. Small, B. W. McColl, R. Allmendinger, J. Pahle, G. López-Castejón, N. J. Rothwell, J. D. Knowles, P. Mendes, D. Brough, and D. B. Kell. Efficient discovery of anti-inflammatory small-molecule combinations using evolutionary computing. *Nature Chemical Biology*, 7(12):902–908, 2011.

44 E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2006.

45 S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved January 15, 2015, from http://www.sfu.ca/~ssurjano.

46 R. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 1998.

47 L. P. Swiler, P. D. Hough, P. Qian, X. Xu, C. Storlie, and H. Lee. Surrogate models for mixed discrete-continuous variables. In *Constraint Programming and Decision Making*, pages 181–202. Springer, 2014.

48 I. Voutchkov and A. Keane. Multi-objective optimization using surrogates. In Yoel Tenne and Chi-Keong Goh, editors, *Computational Intelligence in Optimization*, volume 7 of *Adaptation, Learning, and Optimization*, pages 155–175. Springer Berlin Heidelberg, 2010.

49 T. Wagner, M. Emmerich, A. Deutz, and W. Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. In *Parallel Problem Solving from Nature, PPSN XI*, pages 718–727. Springer, 2010.

50 S. Wilhelm and M. Godinho de Matos. Estimating spatial probit models in r. *R Journal*, 5(1):130–43, 2013.

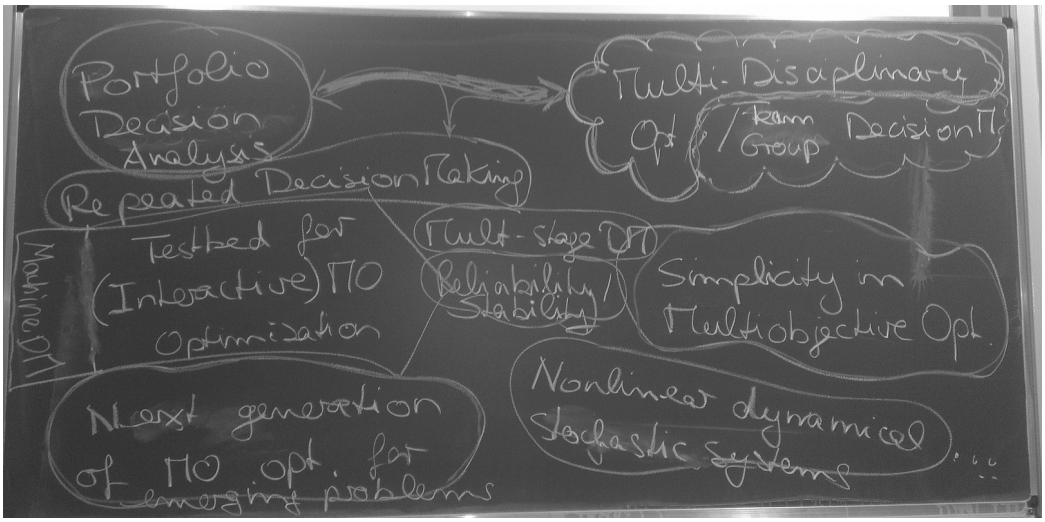
51 R. D. Wilson and T. R. Martinez. Heterogeneous radial basis function networks. In *Proceedings of the International Conference on Neural Networks*, volume 2, pages 1263–1276, 1996.

52 B. S. Yang, Y.-S. Yeun, and W.-S. Ruy. Managing approximation models in multiobjective optimization. *Structural and Multidisciplinary Optimization*, 24(2):141–156, 2002.

53 M. Zaefferer, J. Stork, and T. Bartz-Beielstein. Distance measures for permutations in combinatorial efficient global optimization. In *Parallel Problem Solving from Nature–PPSN XIII*, pages 373–383. Springer, 2014.

5 Topics of interest for participants for next Dagstuhl seminar

Photograph of topics of interest for participants for next Dagstuhl seminar.



6 Changes in the seminar organization body

6.1 Salvatore Greco steps down as co-organizer

On behalf of all the participants of the seminar, JK, KK and GR would like to extend our warm thanks to Salvatore Greco for his contributions to this Dagstuhl seminar series on Multiobjective Optimization as he steps down from the role of co-organizer, which he has held for three terms of office.

Salvo's passion and enthusiasm for research in multiobjective optimization made the seminars even more vivid and joyful than they are anyway. We are thankful for his advice and activities in the preparation and conduction of the seminar. Thank you, Salvo!

6.2 Welcome to Margaret M. Wiecek

We are pleased that our esteemed colleague Margaret M. Wiecek has agreed to serve as co-organizer for future editions of this Dagstuhl Deminar series on Multiobjective Optimization.

7 Seminar schedule

Monday, January 12, 2015

09:00–10:30: Welcome Session
- Welcome and Introduction
- Short presentation of all participants (3 minutes each!)

Coffee Break

11:00–12:00: Introduction to Complexity in Applications
- Robin Purshouse: Perspectives on the application of multi-objective optimization within complex engineering design environments
- Kaisa Miettinen: Sources of computational challenges in multiobjective optimization

Lunch

13:30–14:30: Introduction to Complexity in Preference
- Jürgen Branke, Salvatore Corrente, Salvatore Greco, Roman Slowinski, Piotr Zielnewicz: Preference learning in EMO: Complexity of preference models
- Manuel López-Ibánez: Machine Decision Makers: From Modeling Preferences to Modeling Decision Makers

Coffee Break

15:00- 16:00: Introduction to Complexity in Optimisation
- Matthias Ehrgott: Computational Complexity in Multi-objective (Combinatorial) Optimisation
- Michael Emmerich: An Open Problems Project for Set-Oriented and Indicator-Based Multicriteria Optimization

Break

16:15–18:00: Group Discussion about Hot Topics and Working Groups

Tuesday, January 13, 2015

09:00 – 10:00: Complexity in MO optimization Chair: Daniel Vanderpooten
- Carlos Fonseca: Pareto front approximation statistics
- Andrzej Jaskiewicz: Complex combinatorial problems with heterogeneous objectives

Coffee Break

10:30–12:00: Working Groups

Lunch

13:30–14:30: Complexity in Applications Chair: Sanaz Mostaghim

- Silvia Poles: Understanding and managing complexity in real-case applications
- Patrick M. Reed: Many-objective robust decision making under deep uncertainty: A multi-city regional water supply example

Coffee Break

15:00–17:00: Working Groups

17:00–18:00: Reports from Working Groups

- 6 minutes / 3 slides per working group
- General discussion and working group adaptations

Wednesday, January 14, 2015

09:00–10:00: Complexity in Applications Chair: Carlos Coello Coello

- Ralph Steuer: Tutorial on large-scale multicriteria portfolio selection leading up to difficulties obstructing further progress
- Yaochu Jin: Bridging the gap between theory and application in multi-objective optimization

Coffee Break

10:30–12:00: Working Groups

Lunch

14:00: Group Foto (Outside)

14:05–16:00: Hiking Trip

16:30–18:00: Reports from Working Groups

- 15 minutes / 5 slides per working group

Thursday, January 15, 2015

09:00–12:00: Working Groups

Lunch

13:30–14:30: Complexity in Optimization Chair: Serpil Sayin

- Margaret M. Wiecek: Distributed MCDM under partial information
- Gabriele Eichfelder: Variable ordering structures – what can be assumed?

Coffee Break

15:00–16:00: General Discussion: 10 Years of MCDM-EMO Dagstuhl Seminars. What do we Expect for the Future?

Break

16:30–18:00: Working Groups

20:00: Wine & Cheese Party (Music Room)

Friday, January 16, 2015

09:00–11:00: Presentation of Working Group Results

Coffee Break

11:30–12:00: Summary, Feedback, and Next Steps

Lunch & Goodbye

Participants

- Richard Allmendinger
University College London, GB
- Jürgen Branke
University of Warwick, GB
- Dimo Brockhoff
INRIA – University of Lille 1, FR
- Carlos A. Coello Coello
CINVESTAV, MX
- Salvatore Corrente
Università di Catania, IT
- Matthias Ehrgott
Lancaster University, GB
- Gabriele Eichfelder
TU Ilmenau, DE
- Michael Emmerich
Leiden University, NL
- José Rui Figueira
IST – Lisbon, PT
- Carlos M. Fonseca
University of Coimbra, PT
- Xavier Gandibleux
University of Nantes, FR
- Martin Josef Geiger
Helmut-Schmidt-Universität – Hamburg, DE
- Salvatore Greco
University of Portsmouth, GB
- Jussi Hakanen
University of Jyväskylä, FI
- Carlos Henggeler Antunes
University of Coimbra, PT
- Hisao Ishibuchi
Osaka Prefecture University, JP
- Johannes Jahn
Univ. Erlangen-Nürnberg, DE
- Andrzej Jaszkiewicz
Poznan Univ. of Technology, PL
- Yaochu Jin
University of Surrey, GB
- Miłosz Kadziński
Poznan Univ. of Technology, PL
- Kathrin Klamroth
Universität Wuppertal, DE
- Joshua D. Knowles
University of Manchester, GB
- Renaud Lacour
Universität Wuppertal, DE
- Manuel López-Ibáñez
Free University of Brussels, BE
- Luis Marté
PUC – Rio de Janeiro, BR
- Kaisa Miettinen
University of Jyväskylä, FI
- Sanaz Mostaghim
Universität Magdeburg, DE
- Vincent Mousseau
Ecole Centrale Paris, FR
- Mauro Munerato
ESTECO SpA – Trieste, IT
- Boris Naujoks
FH Köln, DE
- Luís Paquete
University of Coimbra, PT
- Silvia Poles
Noesis Solutions – Leuven, BE
- Robin Purshouse
University of Sheffield, GB
- Patrick M. Reed
Cornell University, US
- Enrico Rigoni
ESTECO SpA – Trieste, IT
- Günter Rudolph
TU Dortmund, DE
- Stefan Ruzika
Universität Koblenz-Landau, DE
- Serpil Sayın
Koc University – Istanbul, TR
- Pradyumn Kumar Shukla
KIT – Karlsruher Institut für Technologie, DE
- Roman Słowiński
Poznan Univ. of Technology, PL
- Ralph E. Steuer
University of Georgia, US
- Michael Stiglmayr
Universität Wuppertal, DE
- Heike Trautmann
Universität Münster, DE
- Tea Tusar
Jozef Stefan Institute – Ljubljana, SI
- Daniel Vanderpooten
University Paris-Dauphine, FR
- Simon Wessing
TU Dortmund, DE
- Margaret M. Wiecek
Clemson University, US
- Xin Yao
University of Birmingham, GB

Report from Dagstuhl Seminar 15041

Model-driven Algorithms and Architectures for Self-Aware Computing Systems

Edited by

Samuel Kounev[1], Xiaoyun Zhu[2], Jeffrey O. Kephart[3], and Marta Kwiatkowska[4]

1 **University of Würzburg, DE,** samuel.kounev@uni-wuerzburg.de
2 **VMware, Inc., US,** xiaoyzhu@yahoo.com
3 **IBM TJ Watson Research Center – Hawthorne, US,** kephart@us.ibm.com
4 **University of Oxford, GB,** Marta.Kwiatkowska@cs.ox.ac.uk

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 15041 "Model-driven Algorithms and Architectures for Self-Aware Computing Systems". The design of self-aware computing systems calls for an integrated interdisciplinary approach building on results from multiple areas of computer science and engineering, including software and systems engineering, systems modeling, simulation and analysis, autonomic and organic computing, machine learning and artificial intelligence, data center resource management, and so on. The Dagstuhl Seminar 15041 served as a platform to raise the awareness about the relevant research efforts in the respective research communities as well as existing synergies that can be exploited to advance the state-of-the-art, formulate a new research agenda that takes a broader view on the problem following an integrated and interdisciplinary approach, and establish collaborations between academia and industry.

Seminar January 18–23, 2015 – http://www.dagstuhl.de/15041
1998 ACM Subject Classification D.2 Software Engineering, D.2.2 Design Tools and Techniques, D.2.4 Software/Program Verification, D.2.5 Testing and Debugging, I.2 Artificial Intelligence, I.6 Simulation and Modelling
Keywords and phrases autonomic systems, self-adaptive, self-managing, model-driven, architecture-based, systems management, machine learning, feedback-based design
Digital Object Identifier 10.4230/DagRep.5.1.164
Edited in cooperation with Aleksandar Milenkoski (University of Würzburg, DE)

1 Executive Summary

Samuel Kounev
Jeffrey O. Kephart
Xiaoyun Zhu
Marta Kwiatkowska

Seminar Description

Self-aware computing systems are best understood as a subclass of autonomic computing systems. The term, autonomic computing, was first introduced by IBM in 2001. Expressing a concern that the ever-growing size and complexity of IT systems would soon become too

difficult for human administrators to manage, IBM proposed a biologically-inspired solution. An analogy was drawn between the autonomic nervous system, which continually adjusts the heart and respiratory rates, pupil dilation, and other lower-level biological functions in response to conscious decisions made by individuals, and autonomic computing systems, which are envisioned as managing themselves in accordance with high-level objectives from humans.

In an effort to enlist the academic community in a worldwide effort to meet this grand challenge, Kephart and Chess laid out a vision of autonomic computing in an IEEE Computing article in 2003 [1]. The article postulated a multi-agent architecture for autonomic computing systems consisting of interacting software agents (called autonomic elements) that consume computational resources and deliver services to humans and to other autonomic elements, and used that architecture as a structure against which a diverse set of research challenges were defined. One of the major challenges from a scientific perspective was the definition of appropriate abstractions and models for understanding, controlling, and designing emergent behavior in autonomic systems. Many different components of IT systems could be autonomic elements – database management systems, load balancers, provisioning systems, anomaly detection system, etc. In addition to managing their own behavior in accordance with policies established by humans or other autonomic elements, they also manage their relationships with other autonomic elements.

The self-managing properties of autonomic computing systems, including self-optimization, self-configuration, self-healing and self-protection, are expected to arise not just from the intrinsic self-managing capabilities of the individual elements, but even more so from the interactions among those elements, in a manner akin to the social intelligence of ant colonies. Understanding the mapping from local behavior to global behavior, as well as the inverse relationship, was identified as a key condition for controlling and designing autonomic systems. One proposed approach was the coupling of advanced search and optimization techniques with parameterized models of the local-to-global relationship and the likely set of environmental influences to which the system will be subjected.

In the ensuing decade, there has been much research activity in the field of autonomic computing. At least 8000 papers have been written on the topic, and explicit solicitations for papers on autonomic computing can be found in the call for papers of at least 200 conferences and workshops annually, including the International Conference on Autonomic Computing (ICAC), now in its tenth year. The European government has funded autonomic computing research projects for several million euros via the FP6 and FP7 programs, and the US government has funded research in this field as well.

In a retrospective keynote at ICAC 2011, Kephart assessed the state of the field, finding through bibliometric analysis that progress in the field has been good but uneven [2]. While there has been a strong emphasis on self-optimization in its many forms, there have been considerably fewer works on other key autonomic properties such as self-configuration, self-protection and self-healing. An apparent reason for this imbalance is that benchmarks that quantify these properties and allow them to be compared across different systems and methods are still largely lacking. Another finding was that much work remains to be done at the system level. In particular, while there has been considerable success in using machine learning and feedback control techniques to create adaptive autonomic elements, few authors have successfully built autonomic computing systems containing a variety of interacting adaptive elements. Several authors have observed that interactions among multiple machine learners or feedback loops can produce interesting unanticipated and sometimes destructive emergent behaviors; such phenomena are well known in the multi-agent systems realm as

well, but insufficiently understood from a theoretical and practical perspective.

It is worth noting that there is a substantial sub-community within the autonomic computing field that applies feedback control to computing systems. FeBID (Feedback Control Implementation and Design in Computing Systems and Networks), a key workshop in this space, began in 2006 as a forum for describing advances in the application of control theory to computing systems and networks. In 2012, FeBID acquired a new name (Feedback Computing) to reflect a much broader and colloquial interpretation of "feedback", in which the goals are no longer merely set points, and system models are not merely used to help transform or transduce signals, but may themselves be adapted through learning. The evolution of this sub-community of autonomic computing reflects a growing acceptance of the idea that, for an autonomic computing element or system to manage itself competently, it needs to exploit (and often learn) models of how actions it might take would affect its own state and the state of the part of the world with which it interacts.

Self-Aware Computing Systems

To understand how self-aware computing systems fit within the broader context of autonomic and feedback computing, we started with the following definition [4, 5] in the beginning of the seminar:

▶ **Definition 1.** A computing system is considered to be "self-aware" if it possesses, and/or is able to acquire at runtime, the following three properties, ideally to an increasing degree the longer the system is in operation:

- Self-reflective: Aware of its software architecture, execution environment, and hardware infrastructure on which it is running as well as of its operational goals (e.g., quality-of-service requirements, cost- and energy-efficiency targets),
- Self-predictive: Able to predict the effect of dynamic changes (e.g., changing service workloads) as well as predict the effect of possible adaptation actions (e.g., changing system configuration, adding/removing resources),
- Self-adaptive: Proactively adapting as the environment evolves in order to ensure that its operational goals are continuously met.

The three properties in the above definition are obviously not binary, and different systems may satisfy them to a different degree, however, in order to speak of "self-awareness", all three properties must apply to the considered system.

To realize the vision of "self-aware" computing systems, as defined above, we advocated a holistic model-based approach where systems are designed from the ground up with built-in self-reflective and self-predictive capabilities, encapsulated in the form of online system architecture models. The latter are assumed to capture the relevant influences (with respect to the system's operational goals) of the system's software architecture, its configuration, its usage profile, and its execution environment (e.g., physical hardware, virtualization, and middleware). The online models are also assumed to explicitly capture the system's operational goals and policies (e.g., quality-of-service requirements, service level agreements, efficiency targets) as well as the system's adaptation space, adaptation strategies and processes.

Figure 1 presents our vision of a self-aware system adaptation loop based on the MAPE-K control loop [3] in combination with the online system architecture models used to guide the system adaptation at runtime. In the following, we briefly describe the four phases of the adaptation loop.

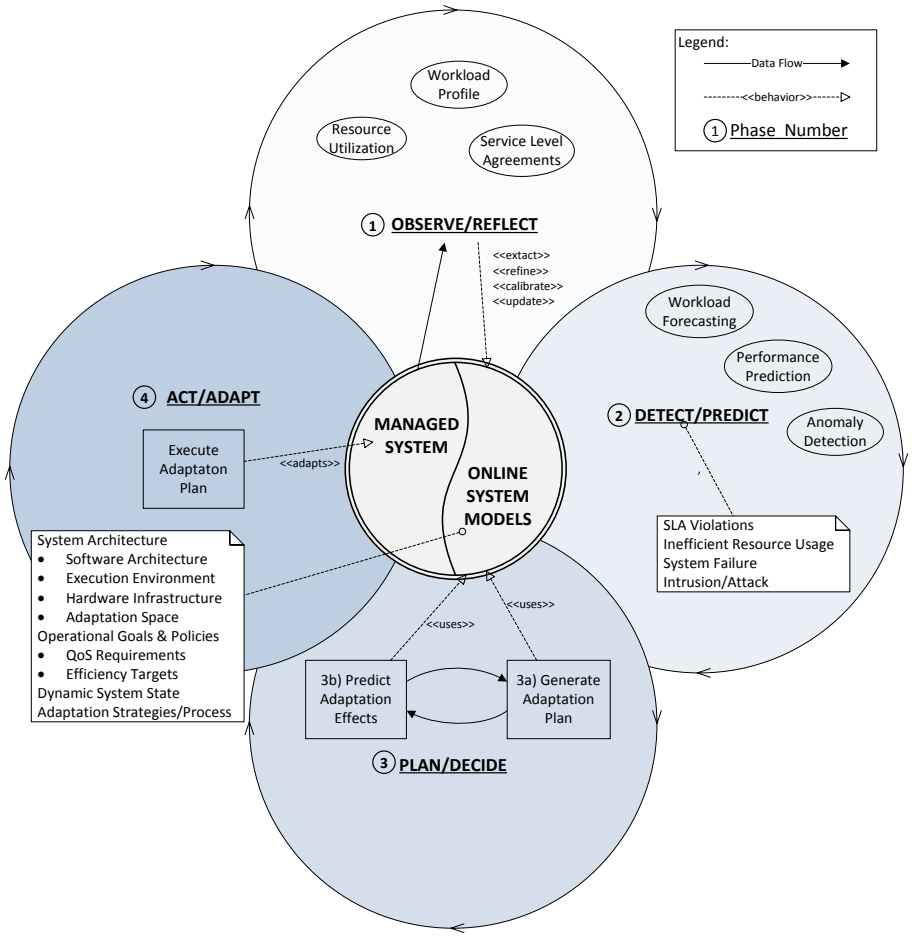


Figure 1 Self-aware system adaptation loop.

Phase 1 (Observe/Reflect): In this phase, the managed system is observed and monitoring data is collected and used to extract, refine, calibrate, and continuously update the online system models, reflecting the relevant influences that need to be captured in order to realize the self-predictive property with respect to the system's operational goals. In the context of this phase, expertise from software engineering, systems modeling and analysis, as well as machine learning, is required for the automatic extraction, refinement and calibration of the online models based on observations of the system at runtime.

Phase 2 (Detect/Predict): In this phase, the monitoring data and online models are used to analyze the current state of the system in order to detect or predict problems such as SLA violations, inefficient resource usage, system failures, network attacks, and so on. Workload forecasting combined with performance prediction and anomaly detection techniques can be used to predict the impact of changes in the environment (e.g., varying system workloads) and anticipate problems before they have actually occurred. In the context of this phase, expertise from systems modeling, simulation, and analysis, as well as autonomic computing and artificial intelligence, is required to detect and predict problems at different time scales during operation.

Phase 3 (Plan/Decide): In this phase, the online system models are used to find an adequate solution to a detected or predicted problem by adapting the system at runtime. Two steps are executed iteratively in this phase: i) generation of an adaptation plan, and ii) prediction of the adaptation effects. In the first step, a candidate adaptation plan is generated based on the online models that capture the system adaptation strategies, taking into account the urgency of the problem that needs to be resolved. In the second step, the effects of the considered possible adaptation plan are predicted, again by means of the online system architecture models. The two steps are repeated until an adequate adaptation plan is found that would successfully resolve the detected or predicted problem. In the context of this phase, expertise from systems modeling, simulation, and analysis, as well as autonomic computing, artificial intelligence, and data center resource management, is required to implement predictable adaptation processes.

Phase 4 (Act/Adapt): In this phase, the selected adaptation plan is applied on the real system at runtime. The actuators provided by the system are used to execute the individual adaptation actions captured in the adaptation plan. In the context of this phase, expertise from data center resource management (virtualization, cluster, grid and cloud computing), distributed systems, and autonomic computing, is required to execute adaptation processes in an efficient and timely manner.

Broader Notion of Self-aware Computing

As a result of the working group "Defining Self-aware Computing Systems", a broader notion of self-aware computing was formulated:

▶ **Definition 2.** Self-aware computing systems are computing systems that:
1. *learn models* capturing *knowledge* about themselves and their environment (such as their structure, design, state, possible actions, and run-time behavior) on an ongoing basis and
2. *reason* using the models (for example predict, analyze, consider, plan) enabling them to *act* based on their knowledge and reasoning (for example explore, explain, report, suggest, self-adapt, or impact their environment)

in accordance with *higher-level goals*, which may also be subject to change.

For a detailed discussion of the interpretation of this definition, we refer the reader to Section 4.1.

Industrial Relevance

The envisioned novel algorithms and architectures for self-aware computing systems are of high relevance to the real-world problems faced by software developers and practitioners in the IT industry. Even though many of the specific problems have been researched upon within the aforementioned disciplines and communities, we believed the timing is right for adopting a broader integrated and interdisciplinary approach and exploiting synergies in the existing modeling and management approaches. The demand and the urgency for providing practical model-driven solutions to the described problems have never been higher, for the following reasons:

Large-scale, on-demand infrastructure: Although the cloud computing concept has been around for a long time, it wasn't until the last few years did we see a wide availability and adoption of cloud computing platforms. Such platforms provide infrastructure-on-demand to business critical applications and high performance computing workloads. Such highly

dynamic, demand-driven environments make many existing automation schemes in computing systems inadequate, because they are mostly rule-based or heuristics-driven and cannot self-adapt to changes in both the infrastructure and the workloads.

Applications and workloads: The ever-increasing variety and complexity of modern applications and their workloads are placing more stress on computing systems and making many traditional management approaches obsolete. This is exacerbated by the extensive use of mobile devices and applications by an increasing population that produces new usage patterns and resource requirements.

Sensors and data: The numbers and types of sensors deployed in computing systems have never been greater, which lead to an explosion of runtime monitoring data that accurately capture the operating conditions of systems and software. Such data significantly enhance the chances for computing systems to Observe/Reflect (Phase 1) and to extract/refine/calibrate online system models that were difficult to learn otherwise, making a model-driven approach more feasible and reliable.

Need for automation: The IT industry is crying out ever so loud for automation technologies to help deal with the above challenges. Automation also helps reduce manual labor cost in management and administration and addresses the increasing gap between the number of skilled IT professionals and the industrial demand. There have been a growing number of startup companies that aim at developing automation solutions for capacity planning, provisioning and deployment, service level assurance, anomaly detection, failure/performance diagnosis, high availability, disaster recovery, and security enforcement. More research on modern-driven algorithms and architectures for self-aware computing can really feed into this new wave of innovations.

Organization of the Seminar

As inspired by the above described vision and approach towards its realization, we believed that the design of self-aware computing systems calls for an integrated interdisciplinary approach building on results from multiple areas of computer science and engineering including: i) software and systems engineering; ii) systems modeling, simulation and analysis; iii) autonomic and organic computing, machine learning and artificial intelligence; iv) data center resource management including virtualization, cluster, grid and cloud computing. This was the motivation of the research seminar. The list of invitees was carefully composed to provide a balance among these fields including both theoretical and applied research with participation from both academia and industry. We note that, in reality, each of the four mentioned communities is in fact comprised of multiple separate sub-communities although they have some overlap in their membership. While they can be seen as separate research communities, we consider them related in terms of their goals, with the difference being mostly in the specific focus of each sub-community and the employed scientific methods. The final participants of the seminar included representatives from each sub-community such that we cover the different relevant focus areas and scientific methodologies.

Achievements of the Seminar

This seminar has achieved its original goal of bringing together scientists, researchers, and practitioners from four different communities, including Software Engineering, Modeling and Analysis, Autonomic Computing, and Resource Management, in a balanced manner. The seminar program provided a basis for exchange of ideas and experiences from these different communities, offered a forum for deliberation and collaboration, and helped identify the technical challenges and open questions around self-aware computing systems. In summary, its achievements are mainly in the following two areas.

Identification of Synergies and Research Questions

By bringing together researchers from the above research fields and their respective communities, we avoid duplication of effort and exploit synergies between related research efforts.

During the seminar, we identified the following research questions and challenges that are of common interest to multiple communities:

- Design of abstractions for modeling quality-of-service (QoS) relevant aspects of systems and services deployed in dynamic virtualized environments. The abstractions should make it possible to capture information at different levels of detail and granularity allowing to explicitly model the individual layers of the system architecture and execution environment, context dependencies, and dynamic system parameters.
- Automatic model extraction, maintenance, refinement, and calibration during operation. Models should be tightly coupled with the system components they represent while at the same time they should abstract information in a platform-neutral manner.
- Efficient resolution of context dependencies including dependencies between the service deployment context and input parameters passed upon invocation, on the one hand, and resource demands, invoked third-party services, and control flow of underlying software components, on the other hand.
- Automatic generation of predictive models on-the-fly for online QoS prediction. The models should be tailored to answering specific online QoS queries. The model type, level of abstraction and granularity, as well as the model solution technique, should be determined based on: i) the type of the query (e.g., metrics that must be predicted, size of the relevant parts of the system), ii) the required accuracy of the results, iii) the time constraints, iv) the amount of information available about the system components and services involved.
- Efficient heuristics exploiting the online QoS prediction techniques for dynamic system adaptation and utility-based optimization.
- Novel techniques for self-aware QoS management guaranteeing service-level agreements (SLAs) while maximizing resource efficiency or minimizing energy cost.
- Standard metrics and benchmarking methodologies for quantifying the QoS- and efficiency-related aspects (e.g., platform elasticity) of systems running on virtualized infrastructures.

The above research questions and challenges were considered in the context of our holistic model-based approach and the self-aware system adaptation loop presented in the previous section. Answering these questions can help determine what system aspects should be modeled, how they should be modeled, how model instances should be constructed and maintained at runtime, and how they should be leveraged for online QoS prediction and proactive self-adaptation.

The online system models play a central role in implementing the four phases of the described system adaptation loop. The term "'model'" in this context is understood in a broad sense since models can be used to capture a range of different system aspects and modeling techniques of different type and nature can be employed (e.g., an analytical queuing model for online performance prediction, a machine learning model for managing resource allocations, a statistical regression model capturing the relationship between two different system parameters, a descriptive model defining an adaptation policy applied under certain conditions). At the seminar, we advocate a model-based approach that does not prescribe specific types of models to be employed and instead we use the term "online system models" to refer to all information and knowledge about the system available for use at runtime as part of the system adaptation loop. This includes both descriptive and predictive models.

Descriptive models describe a certain aspect of the system such as the system's operational goals and policies (quality-of-service requirements and resource efficiency targets), the system's software architecture and hardware infrastructure, or the system's adaptation space and adaptation processes. Such models may, for example, be described using the Meta-Object-Facility (MOF) standard for model-driven engineering, heavily used in the software engineering community.

Predictive models are typically applied in three different contexts: i) to predict dynamic changes in the environment, e.g., varying and evolving system workloads, ii) to predict the impact of such changes on system metrics of interest, iii) to predict the impact of possible adaptation actions at runtime, e.g., application deployment and configuration changes. A range of different predictive modeling techniques have been developed in the systems modeling, simulation and analysis community, which can be used in the "detect/predict" phase of our adaptation loop, e.g., analytical or simulative stochastic performance models, workload forecasting models based on time-series analysis, reliability and availability models based on Markov chains, black-box models based on statistical regression techniques. Finally, models from the autonomic computing and machine learning communities can be used as a basis for implementing the "plan/decide" phase of our adaptation loop. Examples of such models are machine learning models based on reinforcement learning or analytical models based on control theory.

Two important goals of the seminar were to discuss the applicability of the various types of models mentioned above in the context of self-aware computing systems, and to evaluate the tradeoffs in the use of different modeling techniques and how these techniques can be effectively combined and tailored to the specific scenario. As discussed above, in each phase of the self-aware adaptation loop, multiple modeling techniques can be employed. Depending on the characteristics of the specific scenario, different techniques provide different tradeoffs between the modeling accuracy and overhead. Approaches to leverage these tradeoffs at runtime in order to provide increased flexibility will be discussed and analyzed.

Finally, the practical feasibility and associated costs of developing system architecture models was also extensively discussed. We also identified a major target of future research in the area of self-aware computing, which is to automate the construction of online system models and to defer as much as possible of the model building process to system runtime (e.g., the selection of a suitable model to use in a given online scenario, the derivation of adequate model structure by dynamically composing existing template models of the involved system components and layers, the parameterization of the model, and finally, the iterative validation and calibration of the model). Such an approach has the potential not only to help reduce the costs of building system architecture models, but also to bring models closer to the real systems and applications by composing and calibrating them at runtime based on

monitoring of the real observed system behavior in the target production environment when executing real-life operational workloads.

Impact on the Research Community

By bringing together the aforementioned four communities, the research seminar allowed for cross-fertilization between research in the respective area. It has raised the awareness of the relevant research efforts in the respective research communities as well as existing synergies that can be exploited to advance the state-of-the-art of the field of self-aware computing systems. The seminar has left to this Dagstuhl Report that provides an up-to-date point of reference to the related work, currently active researchers, as well as open research challenges in this new field. Given that a significant proportion of the proposed participants are from industry, the seminar also fostered the transfer of knowledge and experiences in the respective areas between industry and academia.

In addition to producing this joint report summarizing, we also found enough support and interest among the seminar participants to continue the collaboration through the following venues: i) writing a joint book to publish at Springer with chapter contributions from the seminar participants, ii) establish a new annual workshop on self-aware computing to provide a forum for exchanging ideas and experiences in the areas targeted by the seminar.

Overall, the seminar opened up new and exciting research opportunities in each of the related research areas contributing to the emergence of a new research area at their intersection.

References

1 Jeffrey O. Kephart, David M. Chess, "*The Vision of Autonomic Computing*," in IEEE Computer, 36(1):41–50, 2003. DOI: 10.1109/MC.2003.1160055

2 Jeffrey O. Kephart, "*Autonomic Computing: The First Decade*", in Proc. of the 8th ACM Int'l Conf. on Autonomic Computing (ICAC'11), pp. 1-2, ACM, 2011. DOI: 10.1145/1998582.1998584

3 IBM Corporation, "*An Architectural Blueprint for Autonomic Computing*", IBM White Paper, 4th Edition, 2006.

4 Samuel Kounev, "*Engineering of Self-Aware IT Systems and Services: State-of-the-Art and Research Challenges*", in Proc. of the 8th European Performance Engineering Workshop (EPEW'11), LNCS, Vol. 6977, pp. 10–13, Springer, 2011. DOI: 10.1007/978-3-642-24749-1_2

5 Samuel Kounev, Fabian Brosig, and Nikolaus Huber, "*Self-Aware QoS Management in Virtualized Infrastructures*", in Proc. of the 8th ACM Int'l Conf. on Autonomic Computing (ICAC'11), pp. 175–176, ACM, 2011. DOI: 10.1145/1998582.1998615

2 Table of Contents

3 Overview of Talks

3.1 Software Engineering for Self-Aware Computing

Samuel Kounev (University of Würzburg, DE)

In this talk, we present an overview of software engineering methods and models for self-aware computing. We first review projects in the area of self-aware computing, as well as related initiatives, discussing and contrasting existing notions of "self-aware computing". We describe the most common types of models in the SE community, and how they are typically analyzed or otherwise used. Finally, we present an example of a modeling language (meta-model) specifically designed for self-aware performance and resource management in modern IT systems.

3.2 Modelling and Analysis (and towards Synthesis)

Marta Kwiatkowska (University of Oxford, GB)

This presentation gave an overview of current research in model-based quantitative analysis, focusing on automated verification and the more recent shift towards synthesis. Probabilistic verification aims to establish the correctness of probabilistic models against safety, reliability and performance properties such as "the probability of an airbag failing to deploy within 0.02s" is less that 1 percent. A widely used tool for this purpose is the probabilistic model checker PRISM (www.prismmodelchecker.org). It is important to also consider the interaction between the system and its environment, which includes human users and other systems. The lecture focused on Markov decision process models, which model environmental uncertainty, and the corresponding verification algorithms for the temporal logic PCTL. It then introduced the synthesis problem, which aims to construct a strategy for the system so that a given quantitative goal, expressed in LTL, is satisfied. Since requirements may conflict, multi-objective synthesis was also discussed. Finally, the need for self-adapation was discussed and the limitations of off-line verification in this regard, due to the need for the systems to take decisions based on current state and external events and to adapt/reconfigure so as to maintain the satisfaction of the objectives. Quantitative runtime verification based on the MAPE loop was put forward for this purpose. The lecture ended with an overview of future challenges. More information can be found in the following papers.

References

1 V. Forejt, and M. Kwiatkowska, G. Norman, D. Parker, "*Automated Verification Techniques for Probabilistic Systems*," in Formal Methods for Eternal Networked Software Systems (SFM'11), LNCS, Vol. 6659, pp. 53–113, Springer, 2011.
2 R. Calinescu, L. Grunske, M. Kwiatkowska, R. Mirandola, G. Tamburrelli, "*Dynamic QoS Management and Optimisation in Service-Based Systems*," in IEEE Transactions on Software Engineering, 37(3):387–409, 2011.
3 R. Calinescu, C. Ghezzi, M. Kwiatkowska, R. Mirandola, "*Self-adaptive software needs quantitative verification at runtime*," in Communications of the ACM, 55(9):69–77, 2012.

3.3 Design-Time Analysis of Properties of Self-Adaptive Systems

Steffen Becker (TU Chemnitz, DE)

Joint work of Becker, Matthias; Lehrig, Sebastian; Becker, Steffen
Main reference M. Becker, S. Lehrig, S. Becker, "Systematically Deriving Quality Metrics for Cloud Computing Systems," in Proc. of the 6th ACM/SPEC Int'l Conf. on Performance Engineering (ICPE'15), pp. 169–174, ACM, 2015.
URL http://dx.doi.org/10.1145/2668930.2688043

Analysing the properties of self-adaptive systems already at design time reduces the risks to implement systems with undesired behaviour that will be detected only in later stages or even in operation. Using an example based on Cloud Computing applications, the talk introduces metrics and analysis approaches we are currently researching. In particular, this includes new metrics for properties of self-adaptive systems. A simulation based approach can be used to analyse these properties using design time models.

3.4 Helping Reflective Systems Build (Better) Self-models

Kirstie Bellman (The Aerospace Corp. – Los Angeles, US)

The purpose of this short talk was to stimulate discussion on the critical need to build self-modeling methods for self-aware systems. To have to provide all models to a system undercuts its ability to adapt and to operate within new environments (or to surprise us with new perceptions or understandings). A "self-aware" model is soon out of sync with its environment (especially new or dynamic ones) and itself if it cannot up-date and even, dramatically change its self-models (as well as environmental models), self-knowledge and behaviors. We described how we currently experiment with self-modeling in our CARS test bed (run by colleague Dr. Phyllis Nelson at Cal State Pomona) by giving the different robotic cars parameterized models which they "actively experiment" to fill in the values of the parameters for their own system. The results of their individualized self-models (e.g., how fast a specific car can go uphill or in grass or how well its different sensors work under different conditions, etc.) impact which strategies it will select to play different games in the test bed. We want to go beyond such templates to where in addition to some initial built-in information, the system has powerful methods for building models. One possibility is to use advanced data mining methods to discover new features and relationships both within the system and within its operational environments. In order for the system to build and evaluate self-models, it needs a way to test how its unique capabilities map into different operational environments and to learn the best way to integrate and to use its capabilities as well as learning its limitations.

Animals spend a great deal of energy and time exploring their environments and 'playing'. This discussion emphasized that complex adaptive systems also need safe places to explore the interactions among their components, to try out new behaviors and to fail, and in so doing, to learn their best strategies for different operational conditions. It also allows the systems to look for gaps and errors in all of its models and essentially to practice 'fire drills', e.g., what to do when errors/problems/unexpected conditions. We then briefly discussed our current approaches to building the necessary language between the self-modeling system and its

human users and our first choices for mathematical methods that can support the development of self- models based on the data resulting from active experimentation (correlating one's own capabilities to different operational environments). We provided some references for our experiences in implementing computational reflection, one form of self-awareness, and self-adaptive architectures for different types of applications with our Wrappings approach (1989).

3.5 Analyzing Architecture-Based Self-Adaptation via Probabilistic Model Checking

Javier Camara (Carnegie Mellon University – Pittsburgh, US)

Joint work of Camara, Javier; Garlan, David; Moreno, Gabriel A.; Schmerl, Bradley;
Main reference J. Cámara, G. A. Moreno, D. Garlan, "Stochastic Game Analysis and Latency Awareness for Proactive Self-Adaptation," in Proc. of the 9th Int'l Symp. on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'14), pp. 155–164, ACM, 2014; pre-print available from project webpage.
URL http://dx.doi.org/10.1145/2593929.2593933
URL http://acme.able.cs.cmu.edu/pubs/show.php?id=407

Design decisions made during early development stages of self-adaptive systems tend to have a significant impact upon system properties (e.g., safety, QoS) during operation. However, understanding a priori the outcome of these decisions is difficult due to the high degree of uncertainty introduced by run-time changes (e.g., resource variability) in the execution context of such systems. Despite the fact that design decisions can be informed by artifacts (e.g., simulations, prototypes), these demand a significant effort to develop and tend to be available only during later stages of the development process. In this talk, I will describe an approach to enable the provision of assurances during early stages in the development cycle of self-adaptive systems. The approach is based on model checking of stochastic multiplayer games (SMGs), allowing developers to approximate the behavioral envelope of a self-adaptive system by analyzing best- and worst-case scenarios of alternative designs for self-adaptation mechanisms. Compared to other sources of evidence, such as simulations or prototypes, our approach provides developers with a preliminary understanding of adaptation behavior with little effort, and without the need to have any specific adaptation algorithms or infrastructure put in place. Moreover, this approach can be used complementarily with other sources of evidence and be used as a starting point to move from development-time to run-time assurances.

3.6 Benchmark requirments for self-aware system

Lydia Y. Chen (IBM Research GmbH – Zürich, CH)

There is a plethora of benchmarks developed by the academics and industries; however there is no such benchmark for evaluating the functionalities and performance of self-aware systems. Developing benchmarks suitable for self-ware system is deemed crucial and challenging. To such an end, one of key issues is how to produce representative workloads for self-aware

systems. For the ease of comprehension, lets consider the cloud datacenter as one of the live examples to illustrate such an issue. One would like to develop a self-aware resource management system for datacenter, so that the short-term resource provisioning and long-term capacity planning can be done in an autonomous fashion.

A valid benchmark specifically designed to evaluate the self-aware aspects shall first generate representative workload patterns. One could argue the definition of representative from the following perspectives. First, the workload patterns shall be identified from the real world applications and systems. Traces provided from the industries provide a rich base for such a need. Second, the workload patterns shall exhibit characteristics that can evoke the self-aware functions in the resource management systems. For example, some application workload has very flat pattern, e.g., constant arrival rate across control management horizon, and there exists no strong need for self-tuning resources. Thus, though such a workload pattern is commonly seen in some corporate datacenter, one may not incorporate such a pattern in benchmarks for the self-aware systems. Nevertheless, the second perspective depends very much on application and system which can have different requirements from each other. To summarize, we are in need to develop benchmarks for self-aware systems and their workload generation shall be representative of real world systems as well as inductive to self-awareness.

3.7 An Extensible Model-driven Architecture for Controlled Self-organisation

Ada Diaconescu (Telecom Paris Tech, FR)

Joint work of Debbabi, Bassem; Diaconescu, Ada; Lalanda, Philippe
Main reference B. Debbabi, A. Diaconescu, P. Lalanda, "Controlling Self-Organising Software Applications with Archetypes," in Proc. of the IEEE 6th Int'l Conf. on Self-Adaptive and Self-Organizing Systems (SASO'12), pp. 69–78, IEEE CS, 2012.
URL http://dx.doi.org/10.1109/SASO.2012.21

Autonomic systems promise to help manage the increasing complexity of modern computing and communication systems. When introducing such autonomic functions, system designers must first determine the necessary balance between system control and runtime flexibility – between provable predictability (guarantees) and open adaptability (survival) of systems running in unpredictable environments.

Models provide formal means to define a system's objectives and runtime state – these offer necessary knowledge for controlling the system's self-* processes and constraining their outcomes. Dynamic extensibility and self-organisation provide means of opening and decentralising system control, in order to expand system adaptability, scalability and robustness.

We have been developing an approach that combines model-driven self-adaptation with decentralised self-organisation in order to construct autonomic systems that are highly adaptable, scalable and robust. A reusable and extensible meta-model and framework are available to help specify and develop autonomic systems in this manner. A prototype has been developed and experimented with in two application domains – smart home resource monitoring and e-health.

3.8 Models in the middle and automated control synthesis: how to improve a software. . . engineer

Antonio Filieri (University of Stuttgart, DE)

Joint work of Filieri, Antonio; Ghezzi, Carlo; Hoffmann, Henry; Leva, Alberto; Maggio, Martina;
Main reference A. Filieri, H. Hoffmann, M. Maggio, "Automated Design of Self-Adaptive Software with Control-Theoretical Formal Guarantees," in Proc. of the 36th Int'l Conf. on Software Engineering (ICSE'14), pp. 299–310, ACM, 2014.
URL http://dx.doi.org/10.1145/2568225.2568272
Main reference A. Filieri, C. Ghezzi, A. Leva, M. Maggio, "Self-Adaptive Software Meets Control Theory: A Preliminary Approach Supporting Reliability Requirements," in Proc. of the 26th IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE'11), pp. 283–292, IEEE, 2011.
URL http://dx.doi.org/10.1109/ASE.2011.6100064

Self-adaptation mechanisms empower software with the ability of withstanding unpredictable changes in its execution environment and handling uncertain knowledge about it. However, these mechanisms rarely provide formal guarantees about their effectiveness and dependability, limiting their applicability in practice.

Control theory has been concerned for decades with controlling of industrial plants aimed at the achievement of prescribed user goals. The mathematical grounding of control theory allows creating controllers effective and dependable by design, under reasonable assumptions about the environmental phenomena that may affect a system behavior.

Despite the conceptual similarity between controlling a plant and adapting software, the application of control theory to self-adaptive systems is very limited, with ad-hoc solutions hard to generalize. One of the main difficulties for a control theory of software is modeling a software behavior in a mathematical form suitable for control design.

This talk will recall two recent approaches for extracting mathematical models. First, the "model-in-the-middle" paradigm, exploiting established analytical quality models as pivot for filling the semantic gap between software design models and dynamic systems of equations. Second, a fully automated model inference and control synthesis technique considering software as a black box, which allow non experts to create software adaptation mechanisms with control theoretical guarantees.

3.9 Runtime Models for Dynamic Teams

Kurt Geihs (University of Kassel, DE)

Joint work of Geihs, Kurt; Niemczyk, Stefan;

Collaboration of autonomous actors (autonomous robots, intelligent agents etc.) requires a common understanding of the joint goals, context, and conditions, i.e. a common shared world model which is maintained at runtime and supports the self-awareness of the team. For example, in order to take concerted decisions on the allocation of tasks a team of soccer robots needs to know the position of the ball, the position of the opponents, the current strategy, etc. Here the required runtime model is known at design time and its structure is static. However, if we consider a more dynamic scenario, such as emergency response where the concrete environment and actors are not known in advance and several teams of heterogeneous autonomous robots as well as human rescue teams need to collaborate, then the question of building and maintaining a shared world model is a real challenge. The

basic research question that we are addressing is: How to create a shared world model for a dynamic group of heterogeneous actors in order to cooperate in unknown situations?

To establish and maintain a shared world model we create a runtime model reflecting the current context, the required information, and the currently available information sources. This model is composed dynamically at runtime by an ASP (Answer Set Programming) solver based on the semantic descriptions of the available system components. We use an OWL ontology to express the structure and semantics of information and system components. Autonomous actors share their semantic knowledge and integrate new elements into their own views. This enables the creation of a common understanding at runtime in order to establish a shared world model and to support the collaboration of autonomous actors in dynamic scenarios.

This research is part of a project cluster called NICER that focuses on 'Networked Infrastructureless Cooperation for Emergency Response'". NICER is a joint research endeavour of TU Darmstadt, University of Kassel, and University of Marburg, supported financially by the state of Hesse in Germany.

3.10 Software Engineering for Self-Adaptive Systems & Self-Aware Computing

Holger Giese (Hasso-Plattner-Institut – Potsdam, DE)

Self-Aware Computing is a very promising direction to enable systems to manage themselves in a more powerful manner than simple rule-based feedback loops can. In this presentation we will therefore first outline the overall landscape of software engineering for self-adaptive systems and our own work related to this. Then we will discuss which role self-aware computing can play in this landscape, but also identify which implications the landscape has for the required building blocks for self-ware computing solutions.

3.11 New Results on Property Specification Patterns

Lars Grunske (University of Stuttgart, DE)

Effective system verification requires two important elements: a model that describes the operations and states of a (adaptive) system, and a set of properties that each implementation of this system must satisfy. Both specification need to be correct. However, for property specification its hard to guarantee that a given specification matches a software engineer's intuition about the system in question. A solution is to use property specification patterns, which describes a generalized recurring system property type and provides a solution in form of a corresponding formal specification. This talk will present results from a unified framework [5, 1] for traditional [4], timed [3] and probabilistic [2] logic specifications.

References

1 M. Autili, L. Grunske, M. Lumpe, I. Meedeniya, P. Pelliccione, and A. Tang, "*Property specification patterns wiki pages*," http://ps-patterns.wikidot.com, 2013.

2 L. Grunske, "*Specification Patterns for Probabilistic Quality Properties*," in *Proceedings of the 30th International Conference on Software Engineering (ICSE'08)*. New York, NY, USA: ACM, 2008, pp. 31–40.

3 S. Konrad and B. H. C. Cheng, "*Real-time Specification Patterns*," in *Proceedings of the 27th International Conference on Software Engineering (ICSE'05)*. New York, NY, USA: ACM, 2005, pp. 372–381.

4 M. B. Dwyer, G. S. Avrunin, and J. C. Corbett, "*Patterns in Property Specifications for Finite-State Verification*," in *Proceedings of the 21st International Conference on Software Engineering (ICSE'99)*. ACM, 1999, pp. 411–420.

5 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*," in IEEE Transactions on Software Engineering, to appear .

3.12 Automated Synthesis of Service Choreographies

Paola Inverardi (University of L'Aquila, IT)

Joint work of Autili, Marco; Inverardi, Paola; Tivoli, Massimo;
Main reference M. Autili, P. Inverardi, M. Tivoli, "Automated Synthesis of Service Choreographies," IEEE Software, 32(1):50–57, 2015.
URL http://dx.doi.org/10.1109/MS.2014.131

Future Internet research promotes the production of a distributed-computing environment that will be increasingly surrounded by a virtually infinite number of software services that can be composed to meet user needs. Services will be increasingly active entities that, communicating peer-to-peer, can proactively make decisions and autonomously perform tasks. Service choreography is a form of decentralized service composition that describes peer-to-peer message exchanges among participant services from a global perspective. In a distributed setting, obtaining the coordination logic required to realize a choreography is nontrivial and error prone. So, automatic support for realizing choreographies is needed. For this purpose, researchers developed a choreography synthesis tool. The Web extra at http://www.di.univaq.it/marco.autili/synthesis/shortdemo/demo.htm is a short demonstration of CHOReOSynt, a choreography synthesis tool.

3.13 Self-* Datacenter Management for Business Critical Workloads

Alexandru Iosup (TU Delft, NL)

Joint work of Iosup, Alexandru; Epema, Dick; van Beek, Vincent; Fei, Lipu; Capota, Mihai; Shen, Siqi; Deng, Kefeng; Hegeman, Tim; Ghit, Bogdan; Yigitbasi, Nezih

Multi-cluster datacenters, and, further, multi-datacenter infrastructure, are supporting increasing amounts and types of computer applications. Among the workloads of these datacenters, business-critical workloads, that is, workloads that support business decision and

intelligence, and that provide business and operational back-ends, are increasingly important (over 20% of the general IT load, according to an IDC study from 2010). Our goal is to build self-* resource managers for datacenters supporting high performance, efficient, and available business-critical and other services. We present in this talk new results in characterizing business-critical workloads running in multi-cluster multi-datacenters, and advances in scheduling such workloads using portfolio scheduling, scheduling by guessing (but not predicting) workload characteristics, and scheduling by enabling elasticity even for data-intensive workloads (for example, scheduling for elastic MapReduce frameworks).

References

1 Vincent van Beek, Siqi Shen, and Alexandruu Iosup: Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters. CCGRID 2015.
2 Bogdan Ghit, Nezih Yigitbasi, Alexandru Iosup, and Dick H. J. Epema: Balanced resource allocations across multiple dynamic MapReduce clusters. SIGMETRICS 2014.
3 Lipu Fei, Bogdan Ghit, Alexandru Iosup, and Dick H. J. Epema: KOALA-C: A task allocator for integrated multicluster and multicloud environments. CLUSTER 2014.
4 Kefeng Deng, Junqiang Song, Kaijun Ren, and Alexandru Iosup: Exploring portfolio scheduling for long-term execution of scientific workloads in IaaS clouds. SC 2013.

3.14 Fairness in Data Stream Processing Under Overload

Evangelia Kalyvianaki (City University – London, GB)

Joint work of Kalyvianaki, Evangelia; Fiscato, Marco; Pietzuch, Peter;

Federated stream processing systems, which utilise nodes from multiple independent domains, can be found increasingly in multi-provider cloud deployments, large-scale grid systems and collaborative sensing applications. To pool resources from several sites and take advantage of local processing, submitted queries are split into query fragments, which are executed collaboratively by different sites. When supporting many concurrent users, however, queries may exhaust available processing resources, thus requiring constant load shedding. Given that individual sites have autonomy over how they allocate query fragments on their nodes, it is an open challenge how to ensure global fairness on processing quality experienced by queries in a federated scenario.

We describe THEMIS, a federated stream processing system for resource starved, multi-site deployments. It executes queries in a globally fair fashion and provides users with constant feedback on the experienced processing quality for their queries. THEMIS associates stream data with its source information content (SIC), a metric that quantifies the "value" of that data, as perceived by the user, based on the amount of source data used to generate it. We provide a distributed load shedding algorithm that implements fairness on the SIC values of result data. Our evaluation shows that, compared to a random shedding approach, THEMIS achieves a 33 percent more fair processing quality across queries as measured with the Jain's index fairness metric. Our approach also incurs a low execution time overhead.

3.15 The Descartes Modeling Language for Self-Aware Performance and Resource Management

Samuel Kounev (University of Würzburg, DE)

The Descartes Modeling Language (DML) is a novel architecture-level language for modeling performance and resource management related aspects of modern dynamic software systems and IT infrastructures. Technically, DML is comprised of several sub- languages, each of them specified using OMG's Meta-Object Facility (MOF) and referred to as meta-model in OMG's terminology. The various sublanguages can be used both in offline and online settings for application scenarios like system sizing, capacity planning and trade-off analysis, as well as for self-aware resource management during operation. In this talk, we present an overview of DML and related tools for performance and resource management.

3.16 Interplay of Design Time Optimization and Run Time Optimization

Anne Koziolek (Karlsruhe Institute of Technology, DE)

The aim of designing self-aware systems is to enable the system to meet operational goals at runtime. A common goal is to maintain quality of service properties while minimizing costs of operation. We face an optimization problem. Software optimization is also a topic at design time, e.g. optimizing quality properties of software architectures.

In this talk, I would like to start a discussion on the interaction of design time optimization and runtime optimization. When designing self-aware systems, we should explore what are indeed completely unknown aspects so that the system needs to react in an autonomous way and what are parameters of the environment that vary, but that vary within known bounds so that we can optimize for different parameter combinations already at design time.

3.17 Self-aware Computing in Industry 4.0

Heiko Koziolek (ABB AG Research Center Germany – Ladenburg, DE)

Industry 4.0 is a project in the high-tech strategy of the German government, which promotes the computerization of the manufacturing industry. The goal is the intelligent factory (Smart Factory), which is characterized by adaptability, resource efficiency and ergonomics as well as the integration of customers and business partners in business and value processes. Technological basis are cyber-physical systems and the Internet of Things. The talk outlines several application scenarios envisioned for Industry 4.0, where self-aware computing is expected to help improving production processes. Based on the Industry 4.0 vision, a number of challenges arise for self-aware computing, which are discussed in the talk.

3.18 Types of Computational Self-awareness

Peter Lewis (Aston University – Birmingham, GB)

Joint work of Lewis, Peter R.; Faniyi, Funmilade; Bahsoon, Rami; Yao, Xin; Torresen, Jim; Chandra, Arjun
Main reference F. Faniyi, P. R. Lewis, R. Bahsoon, X. Yao, "Architecting Self-aware Software Systems," in Proc. of the 11th Working IEEE/IFIP Conf. on Software Architecture (WICSA'14), pp. 91–94, IEEE, 2014.
URL http://dx.doi.org/10.1109/WICSA.2014.18

Novel computing systems are increasingly being composed of large numbers of heterogeneous components, each with potentially different goals or local perspectives, and connected in networks which change over time. Management of such systems quickly becomes infeasible for humans. As such, future computing systems should be able to achieve advanced levels of autonomous adaptive behaviour. In order for a system to effectively adapt itself in such a context, its ability to be self-aware becomes important.

There are several clusters of research in computer science and engineering which have used the term self-awareness explicitly. However, we are only now developing a common framework for describing the self-awareness capabilities of these systems. Questions remain about how to measure the benefits and costs that increased self-awareness brings.

In this talk, I shall begin by surveying definitions and current understanding of self-awareness in psychology. I will then describe how these concepts are being translated from psychology to the computing domain, and show how their explicit consideration may be beneficial in the engineering of adaptive computing systems. I will discuss how computational self-awareness may include knowledge of internal state, history, social or physical environment, goals, and perhaps even a system's own way of representing or reasoning about these things.

3.19 Control Theory for Model-Based Software Engineering

Martina Maggio (Lund University, SE)

Joint work of Maggio, Martina; Klein, Cristian; Papadopoulos, Alessandro Vittorio; Årzén, Karl-Erik
Main reference C. Klein, M. Maggio, K.-E. Årzén, F. Hernández-Rodriguez, "Brownout: Building More Robust Cloud Applications," in Proc. of the 36th Int'l Conf. on Software Engineering (ICSE'14), pp. 700–711, ACM, 2014.
URL http://dx.doi.org/10.1145/2568225.2568227

Many software applications may benefit from the introduction of control theory at different stages of the development process. The requirements identification often translates directly into the definition of control goals. In fact, when these requirements are quantifiable and measurable, control theory offers a variety of techniques to complement the software design process with a feedback loop design process, that empowers the original software with self-adaptive capabilities and allows it to fulfill the mentioned quantifiable requirements.

The feedback loop design process consists in the definition of the "knobs", what can be changed during the software life that affect the software behavior and the measurements of the goals. Control theory allows then to define models that describes the relationship between the values of the knobs and the measured values of the software behavior. These models are used to design decision loops and guarantee properties of the closed-loop systems.

In this talk I will briefly describe examples where model-based control allowed us to guarantee the satisfaction of specific properties, like synchronization between different nodes in a wireless sensor network and upper bounds on response times in a cloud application.

3.20 Janus: Optimal Flash Provisioning for Cloud Storage Workloads

Arif Merchant (Google Inc. – Mountain View, US)

Joint work of Albrecht, Christoph; Merchant, Arif; Stokely, Murray; Waliji, Muhammad; Labelle, Francois; Coehlo, Nathan; Shi, Xudong; Schrock, Eric
Main reference C. Albrecht, A. Merchant, M. Stokely, M. Waliji, F. Labelle, N. Coehlo, X. Shi, C. E. Schrock, "Janus: Optimal Flash Provisioning for Cloud Storage Workloads," in Proc. of the USENIX Annual Technical Conf. (ATC'13), pp. 91–102, USENIX, 2013.
URL https://www.usenix.org/conference/atc13/technical-sessions/presentation/albrecht

Janus is a system for partitioning the flash storage tier between workloads in a cloud-scale distributed file system with two tiers, flash storage and disk. The file system stores newly created files in the flash tier and moves them to the disk tier using either a First-In-First-Out (FIFO) policy or a Least-Recently- Used (LRU) policy, subject to per-workload allocations. Janus constructs compact metrics of the cacheability of the different workloads, using sampled distributed traces because of the large scale of the system. From these metrics, we formulate and solve an optimization problem to determine the flash allocation to workloads that maximizes the total reads sent to the flash tier, subject to operator-set priorities and bounds on flash write rates. Using measurements from production workloads in multiple data centers using these recommendations, as well as traces of other production workloads, we show that the resulting allocation improves the hit rate by 47–76 percent compared to a unified tier shared by all workloads. Based on these results and an analysis of several thousand production workloads, we conclude that flash storage is a cost-effective complement to disks in data centers.

3.21 Projecting Disk Usage Based on Historical Trends in a Cloud Environment

Arif Merchant (Google Inc. – Mountain View, US)

Joint work of Stokely, Murray; Mehrabian, Amaan; Albrecht, Christoph; Labelle, Francois; Merchant, Arif
Main reference M. Stokely, A. Mehrabian, C. Albrecht, F. Labelle, A. Merchant, "Projecting disk usage based on historical trends in a cloud environment," in Proc. of the 3rd Workshop on Scientific Cloud Computing (ScienceCloud'12), pp. 63–70, ACM, 2012.
URL http://dx.doi.org/10.1145/2287036.2287050

Provisioning scarce resources among competing users and jobs remains one of the primary challenges of operating large-scale, distributed computing environments. Distributed storage systems, in particular, typically rely on hard operator-set quotas to control disk allocation and enforce isolation for space and I/O bandwidth among disparate users. However, users and operators are very poor at predicting future requirements and, as a result, tend to over-provision grossly. For three years, we collected detailed usage information for data stored in distributed file systems in a large private cloud spanning dozens of clusters on multiple continents. Specifically, we measured the disk space usage, I/O rate, and age of stored data for thousands of different engineering users and teams. We find that although the individual time series often have non-stable usage trends, regional aggregations, user classification, and ensemble forecasting methods can be combined to provide a more accurate prediction of future use for the majority of users.

We applied this methodology for the storage users in one geographic region and back-tested these techniques over the past three years to compare our forecasts against actual usage. We find that by classifying a small subset of users with unforecastable trend changes due to known product launches, we can generate three-month out forecasts with mean absolute errors of less than 12 percent. This compares favorably to the amount of allocated but unused quota that is generally wasted with manual operator-set quotas.

3.22 Evaluating Security Mechanisms in Dynamic Virtualized Environments

Aleksandar Milenkoski (University of Würzburg, DE)

Joint work of Milenkoski, Aleksandar; Vieira, Marco; Payne, Bryan D.; Antunes, Nuno; Kounev, Samuel; Avritzer, Alberto

The benefits of evaluation of security mechanisms (i.e., intrusion detection systems and access control systems) are manyfold. For instance, one may compare multiple IDSes in terms of their attack detection accuracy in order to deploy a mechanism which operates optimally in a given environment, thus reducing the risks of a security breach. Further, one may tune an already deployed mechanism by varying its configuration parameters and investigating their influence through evaluation tests. Many recent research works propose novel architectures of security mechanisms specifically designed to operate in dynamic virtualized environments. In this talk, we present recent advances in evaluating such mechanisms, including a method and a tool for generating workloads that contain virtualisation-specific attacks, and metrics that take elasticity, a feature of modern virtualised environments, into account. Finally, we present open challenges and requirements when it comes to evaluating security mechanisms deployed in self-adaptive virtualized systems with self-protecting features.

3.23 Adapting the Adaptation Logic

Felix Maximilian Roth (University of Mannheim, DE)

Joint work of Roth, Felix Maximilian; Krupitzer, Christian; Becker, Christian

Self-adaptive systems are a response to the increasing complexity and size of information systems. They are able to adapt their behavior to changes in the context or system resources. Thus, they need to be self-aware in order to monitor their state. Furthermore, a self-adaptive system consists of managed resources and the adaptation logic. The managed resources realize functionality – they might be hardware or software components – whereas the adaptation logic controls the adaptation. For this purpose, the adaptation logic uses a feedback loop, such as IBM's MAPE-loop. So far, many researchers have studied the field of adapting the managed resources. However, research in the adaptation of the adaptation logic is still at the beginning. In this talk I will briefly discuss why not only managed resources need to be adapted but also the adaptation logic and, furthermore, how this can be achieved.

3.24 Analysis of Mobile Offloading Strategies

Katinka Wolter (FU Berlin, DE)

Joint work of Wolter, Katinka; Wu, Huaming; Wang, Qiushi;
Main reference Q. Wang, K. Wolter, "Reducing Task Completion Time in Mobile Offloading Systems through Online Adaptive Local Restart," in Proc. of the 6th ACM/SPEC Int'l Conf. on Performance Engineering (ICPE'15), pp. 3–13, ACM, 2015.
URL http://dx.doi.org/10.1145/2668930.2688041

Mobile offloading migrates heavy computation from mobile devices to powerful cloud servers. It is a promising technique that can save energy of the mobile device while keeping job completion time low when cloud servers are available and accessible.

The benefit obtained by offloading greatly depends on whether it is applied at the right time in the right way. We use queueing models to minimize different metrics, such as the Energy-Response time Product (ERP) or a weighted sum of energy consumption and performance expressed in the Energy-Response time Weighted Sum (ERWS) metric. We consider different offloading policies (static and dynamic), where arriving jobs are processed either locally or remotely in the cloud. Offloading can be performed via different wireless technologies. We assume that the transmission techniques differ in energy requirement and speed.

We find that the dynamic offloading policy derived from the tradeoff offloading policy (TOP) outperforms other policies like the random selection of transmission channel by a significant margin. This is because the dynamic offloading policy considers the increase in each queue and the change in metric that newly arriving jobs bring in should they be assigned to that queue. The ERWS metric can be reduced further by considering either energy consumption or response time and it is minimal when optimising only energy consumption.

4 Working Groups

4.1 Working Group: "Defining Self-aware Computing Systems"

Samuel Kounev, Ada Diaconescu, Kirstie Bellman, Peter Lewis, Holger Giese, Javier Camara, Nelly Bencomo, Lukas Esterle, Henry Hoffmann, Hartmut Schmeck, Xin Yao, Sebastian Götz, and Andrea Zisman

This working group discussed broader notions of self-aware computing systems accommodating the research communities focussing on the different aspects of self-aware computing (i.e., software and systems engineering, systems modeling, simulation and analysis, autonomic and organic computing, machine learning and artificial intelligence, data center resource management, and so on). The following definition of self-aware computing systems was formulated:

Self-aware computing systems are computing systems that:

1. *learn models* capturing *knowledge* about themselves and their environment (such as their structure, design, state, possible actions, and run-time behavior) on an ongoing basis and

2. *reason* using the models (for example predict, analyze, consider, plan) enabling them to *act* based on their knowledge and reasoning (for example explore, explain, report, suggest, self-adapt, or impact their environment)

in accordance with *higher-level goals*, which may also be subject to change.

It is assumed that a self-aware system is built by an entity with some higher-level goals in mind. This entity may be a human (e.g., a developer) or a set of humans (e.g., a developer team), but it doesn't necessarily have to be. The entity that built the system may also be another computing system, at a higher-level, that generates code to build a new system for a given purpose.

The major distinctive characteristics of a self-aware computing system are: i) it must have the capability to learn models on an ongoing basis, capturing knowledge relevant to the purpose for which it is built, ii) it must be able to reason about this knowledge and act accordingly. Both the learning and reasoning part are driven by the system's goals, normally imposed by the entity that built the system. The goals are referred to as *higher-level* goals to emphasize that they are at a higher-level than the system itself and they are not under its control. Note that the system itself may generate its own goals (at lower levels) as part of its model learning and reasoning processes.

The term "model" is used here in a general sense and refers to any abstraction of the system and its environment that captures some knowledge and may be used for reasoning with respect to the system goals. In his general model theory, Stachowiak [1] identifies the following three features as essential for models: i) *mapping:* a model is always a model of some *original* (which can be a model itself), ii) *reduction:* a model always *abstracts* from the original by reflecting only a subset of its attributes, and iii) *pragmatic:* a model only replaces the original for a certain *purpose*. Usually, we further distinguish *descriptive* models that capture the originals as they are from *prescriptive* models that describe envisioned futures (planned originals). Descriptive models, in our context, describe a given system aspect that may be relevant with respect to the system's higher-level goals. We further distinguish *predictive models* that support more complex reasoning such as, for example, predicting the system behavior under given conditions or predicting the impact of a considered possible adaptation action.

Some examples of different types of models capturing various aspects that may be relevant in a given scenario include:

- a descriptive model capturing the system's resource landscape and software architecture and their performance-relevant parameters
- a descriptive model describing the system's possible adaptation actions (degrees-of-freedom at run-time)
- a prescriptive model describing how to act in a given situation (e.g., after a component failure)
- a descriptive model describing the system's goals and policies (e.g., service level agreements)
- a predictive statistical regression model capturing the influence of user workloads on the system resource consumption
- a predictive stochastic model allowing to predict the system performance for a given user workload and resource allocation
- a control theory model used to guide the system behavior

We stress that the term "learn" does not imply that all information based on which models are derived is obtained at system run-time. Models may be derived both from

static information provided by the entity that built the system, as well as from dynamic information that is gathered and maintained at run-time. Typically, a combination of both would be expected, for example, a system may be built with integrated skeleton models whose parameters are estimated using monitoring data collected at run-time. The model learning is expected to happen on an *ongoing basis* during operation meaning that models should be continuously refined and calibrated in order to better fulfill the purpose for which they are used.

The term "reason" in the definition refers to any type of model-based analysis that goes beyond applying explicitly programmed, i.e., hard-coded, rules or simple heuristics. Depending on the considered type of system and its respective goals, different types of reasoning may be relevant. For example, in the context of an IT system that has been designed to guarantee certain performance requirements, the following types of reasoning may be relevant:

- predict an IT system's performance (e.g., response time) for a given workload and resource allocation (e.g., number of servers)
- forecast the system load (e.g., number of users or requests sent per unit of time) in a future time horizon
- predict the expected impact of a given system adaptation action (e.g., adding system resources) on the end-to-end system performance

An example of reasoning in the context of a cyber-physical system for traffic management may be to analyze the traffic situation in order to provide a recommendation which route to take for a given target destination.

By stressing the role of model learning and model-based reasoning, driven by higher-level goals, we distinguish the term self-aware computing from related terms such as autonomic computing or self-adaptive systems. Although, in most cases, it would be expected that the system uses the learned models to reason *and self-adapt* to changes in the environment, self-adaptation (often referred to as *self-expression* in this context) is not strictly required. In this way, we accommodate cases where all adaptation actions must be supervised and authorized by an entity outside of the system, such as the entity that built the system or a human system user. For example, in mission-critical cognitive computing applications, systems may provide recommendations on how to act, however, the final decision on what specific action to take is often made by a human operator.

References

1 H. Stachowiak, "*Allgemeine Modelltheorie*," Springer, Wien, 1973.

4.2 Working Group: "Quantification of Self-aware Systems: Metrics & Benchmarks"

Sara Bouchenak, Xiaoyun Zhu, Lydia Y. Chen, Evangelia, Kalyvianaki, Evgenia Smirni, K. R. Jayaram, Alexandru Iosup, Kirstie L. Bellman, Anders Robertsson, Heiko Koziolek, Steffen Becker, Christian Becker, Arif Merchant, and Aleksandar Milenkoski

This working group discussed both metrics and benchmarks for self-aware systems. Most of the participants had a background in cloud computing or storage systems and viewed the topic from that perspective. A few participants also emphasized the broader perspective of self-aware systems outside of the cloud computing domain. The group first discussed criteria for the quantification of self-aware systems. Metrics need to span multiple non-functional properties, such as performance, timeliness, scalability, dependability, trustworthiness, security, safety, costs, adaptability and agility. Benchmarks need to be representative for a certain application domain, exhibit dynamic behavior (i.e., changing at runtime) and provide different failure scenarios that can trigger adaptations. The participants mentioned that several papers on good criteria for constructing benchmarks for cloud computing systems exist.

A few metrics for self-aware system were brought in a brainstorming session: it could be measured how much information is processed by a self-aware system to make informed adaptation actions. Measuring the adaptation overhead in terms of duration, cost, and quality could help comparing similar self-aware systems in an application domain. The "level of self-awareness" is another proposed metric, which is however is hard to quantify. Stability could be measured for systems with lots of adaptation actions. A more extended metric would be "number of surprising finding" made by a self-adaptive systems, i.e., suggestions or decisions for adaptation actions that were not obvious for human observers.

Only a few benchmarks for self-aware or self-adaptive systems were known to the group. But several participants pointed out that these were usually benchmarks originally not designed for self-adaptive systems, e.g., cloud computing benchmarks that were extended with dynamically changing properties at runtime. For example the community website self-adaptive.org list two benchmarks: ZNN.com (a webserver system providing a simplified news site), and ATRP (automated traffic routing problem). Outside the cloud computing domain only a few benchmarks were known. In conclusion, the area of quantification of self-aware systems appeared to the participants as a fruitful research area still in initial stages, which bears many topics for future research.

4.3 Working Group: "Generic Architectures for Collective Self-aware Computing Systems"

Kirstie Bellman, Nelly Bencomo, Ada Diaconescu, Lukas Esterle, Holger Giese, Sebastian Götz, Chris Landauer, Peter Lewis, and Andrea Zisman

The purpose of this working group was to discuss the architectural aspects relevant to collectives of self-aware computing systems, meaning, of several self-aware computing systems interacting in some way, such that they may also themselves comprise a self-aware system.

We discussed where the goals for such a collective could come from, in terms of both sources external and internal to the collective. Further, we considered how one could reason about or control the dynamics of the collective's goals, which might include, for example how to induce their individual goals cohering into a mutually beneficial set. We discussed the case where the individual systems coordinate or cooperate in order to achieve a common goal, when their goals might lead them to competition, and also situations where the systems could pursue their goals in parallel without interference.

One important point emerged that we must not assume that systems inside a collective will all be alike or have the same levels of self-awareness. In fact, one objective of an individual system, for example upon joining a collective, might be the discovery of other members' levels of self-awareness and capabilities. One challenge is to coordinate such a collective given these different levels of awareness (which can include no awareness).

In this context, three key challenges were identified by the group:

1. Whether we could define a meta-architecture for collectives of self-aware systems, which might express the space of possible concrete architectures for a (self-aware) collective of self-aware systems.
2. How high level goals coming from outside the collective could be decomposed to individual goals for component systems.
3. How the types and levels of self-awareness at the component system level and the global system level relate to each other. For example, if a certain level of self-awareness is desired at the global level, what form of self-awareness at the component system level is required?

In attempting to tackle challenge 1, and with a focus on how this might support solutions to challenge 2, we first identified some of the essential interfaces (or input/output ports) that self-aware systems should expose in order to be able to interact with one another. These include:

- Input goal(s): to receive goals from external entities, such as human administrators, users, or other systems;
- Output evaluation(s): to report (to the goal-requiring entities) the extent to which their goals have been achieved;
- Output goal(s): to require goals to be achieved by other systems;
- Input evaluation(s): to obtain reports from the self-aware systems which it has required goals from;
- Input/output communication: to manage relations with other systems.

Concerning the self-aware systems' connectivity within collectives, it was noted that the output goal interface of one self-aware system can be connected to the input goal interface of another system, or of several other systems. Notably, it can also be connected to the system's

own input goal interface, triggering exploratory play behaviour, which is considered essential for the system's proactive learning process. It was also noted that a self-aware system can in turn be composed of other self-aware systems, and so on, in a recursive manner.

Based on these observations, we next focused on the range of seemingly viable integration alternatives of self-aware systems, from different perspectives. This resulted in a first attempt to provide an organisational taxonomy comprising the main axes of possible variations. The purpose was to go towards defining a meta-architecture from which concrete architecture instances of collective self-aware system could be instantiated, by selecting precise values from each one of the variability axes. Three such variability axes were identified:

- System types: representing various levels of self-awareness, ranging from reactive systems to meta-self-aware systems;
- Inter-system relations: including cooperation, competition, ignorance and parasitism;
- System inter-connection patterns: including hierarchy, centralised and peer-to-peer.

Considering challenge 3, and in terms of the implications of having possibly different levels of self-awareness in components of a collective system, as well as different levels apparent between global and local levels, we discussed the learning of individuals versus the learning of the collective as a whole. We discussed the concept of "active experimentation" in the CARS architecture as an example of creating suitable test-beds or 'places' that a system can experiment, learn what its boundaries are, learn where it fails in different operational contexts, and because of which subsystems; or in the case of collectives, learn which systems it can integrate or not in different ways. We also discussed how a collective could learn strategies for different goals that it could potentially achieve and then decide at runtime whether it can achieve those goals or better offload (some of) them to others for instance. Furthermore we briefly explored the effects of individual components leaving or joining the collective during runtime on the level of self-awareness of the collective and its capability to achieve its high-level goals.

We further distinguished between the scope of a component system's self-awareness being limited to the individual component system, or also being aware of the wider collective. In the second case, a feedback loop between the global system and local components' self-awareness properties will have significant impact on many design choices in self-aware collectives.

The group was productive in identifying core challenges involved in the architecture and design of collectives of self-aware systems. The main research challenges identified included the meta-architecture specification, the decomposition of goals and knowledge across self-aware systems within a collective, and the possible relations between the levels of self-awareness of a collective and of the systems within the collective. These challenges, and the results of this working group's subsequent discussions have provided a base for a future collaboration among most participants for writing a corresponding chapter in a book dedicated to self-aware computing.

4.4 Working Group: "Benchmarking Self-aware Computing Systems"

Alexandru Iosup, Sara Bouchenak, Xiaoyun Zhu, Lydia Chen, Evangelia Kalyvianaki, K. R. Jayaram, Kirsty Bellman, Anders Robertson, Heiko Koziolek, Steffen Becker, Evgenia Smirni, Arif Merchant, Aleksandar Milenkoski, and Felix Maximilian Roth

The purpose of this working group was to discuss how to benchmark self-aware computing systems. We discussed the types of problems that appear when benchmarking self-aware computing systems, what are the aspects/bottlenecks to benchmark in self-aware computing systems, what are the unique benchmarking aspects of self-aware computing systems (also relative to non-self-aware computing systems). Each aspect was discussed from a conceptual perspective matched to one or several use cases or examples.

Six important challenges emerged:

1. New tools to evaluate and benchmark self-aware computing systems. There is a general lack of workloads for benchmarking. New workloads should include input data, for the new data-intensive applications that are now common in datacenters and other application scenarios. New workloads should include various patterns of load intensity, including burstiness. How to represent realistic scenarios, e.g., for a specific industry, without introducing undue complexity?
2. New metrics and metric bundles to benchmark self-aware computing systems. These systems need to be self-* in maintaining or optimizing multiple aspects, such as performance, availability, energy consumption, financial and human-resource costs, security, etc.
3. Quantifying the type and level of self-awareness and self-adaptation. Characterizing the transient nature of the operation of these systems is challenging. What is elasticity? How to quantify performance variability? Which metrics for self-configuration, self-expression, etc.? What overheads to consider? etc.
4. Quantifying the type and level of self-protection and self-healing. Measurement and metrics of intrusion-detection scenarios. Measurement and metrics for performance isolation. etc.
5. Using benchmarks online, to improve or even provide self-awareness. How can system properties be uncovered online? How can this knowledge be used online?
6. Benchmarking the benchmarks. What is the coverage provided by (current) benchmarks, in terms of workload representativeness, data aging, etc.? How to define and refine pain points/bottlenecks of self-aware systems? What type and to what level of dynamicity is the benchmark addressing? etc.

The discussion was productive in identifying and sharing metrics, use cases, benchmarking goals, and in identifying and formulating five key challenges for the future of benchmarking self-aware computing systems. Most participants are currently collaborating in topics related to this meeting. In particular, we are collaborating in writing a chapter on benchmarking in a book dedicated to self-aware computing.

4.5 Working Group: "Architecture and Reuse of Self-Aware Systems"

Anne Koziolek, Christopher Landauer, Anne Koziolek, Heiko Koziolek, and Evangelia Kalyvianaki

This working group discussed the architecture of self-aware systems and reuse as a selected aspect of the life-cycle of such systems. Additionally we touched on the theoretical limitations of self-awareness.

The group first discussed the autonomic control loop with its MAPE-K phases in the context of self-aware systems. We found that self-aware systems distinguish themselves from general self-adaptive systems or autonomic systems by focussing on the "Monitor" and "Analyze" phases. Additionally, the "Knowledge" in a self-aware system includes knowledge about the system itself and even about its MAPE-K loop. For a specific control loop for self-aware systems, one might consider to specialize the "Knowledge" by "Self-Knowledge", "Reflective Models" or just "Models". We also noted that for different applications, some parts of MAPE-K loop are not explicitly available or modelled (in code or in the running system). Especially the "Execute" step is not required in self-aware system, as our working definition does not strictly require the system to act upon its observations and analyses.

The group also talked about the models and controllers in the Wrappings approach of Kirstie L. Bellman and Christopher Landauer, where the phases of MAPE-K are not explicitly distinguished. In Wrappings, models and controllers are the same. The controller is also interpreted based on a model. As an example, a system changes itself by bringing in new resources. Still selection and execution process. To cite from on of their papers, Wrappings is based on two key, complementary parts: (1) explicit, machine-interpretable descriptions meta-knowledge of all software, hardware, and other computational resources in a Constructed Complex System (organized into Wrapping Knowledge Bases), and (2) active integration processes that use that information to select, adapt, and combine these resources for particular problems (the Problem Managers). The integration process needs to be guided by constraints to do something useful and it is the task of the designer to define these constraints.

As a second topic, we discussed reuse as an aspect of the life-cycle of self-aware systems. Starting again with the MAPE-K picture, we hypothesized that the system to manage by the MAPE-K loop (the autonomic manager in the original IBM terms) can be exchanged and everything else can be reused to some extend. Models might have to be updated so that they match the new system. The part in the Knowledge describing the awareness of the MAPE-K loop is likely to be highly reusable across different systems. Also analyses can be reusable. For example, a performance analysis approach that learns the appropriate performance model (whether or not the system has exponential distributed service times, what the relevant queues in the system are, etc).

We considered whether the system be changed in other ways than defined by the actuators. One opinion was that there is no way to in general know the side effects of such unanticipated changes, the manager cannot change code in general. At the same time, one could exchange the whole system.

Continuing with reuse, we found that controllers might be reusable. Additionally, monitoring hooks are reusable (actually there are many monitoring frameworks available).

One challenge when reusing are the dependencies among the MAPE steps, as a subsequent step needs the right output from the previous step. It might be practical to have specialized

models, such as a model of web server applications. Such a model can learn specific parameters at run time. It is reusable for different types of web servers, but not beyond that.

This lead us to the question of how generalizable and expressive the models can be. We hypothesized that the models should not be too general. At some point, one would reach the generality of a Turing machine, which is not helpful in practice as it is not analysable.

Continuing on that thought, we hypothesized that if there was a model that can be aware of any system, it would be supported by the architecture. A potential limitation of the self-awareness lies in the used formalism of the model (as one does not want to be as general as Turing machine, there are some limitations of what such models can express). Additionally, there may be limitations in what you can monitoring and execute. Finally, we hypothesized that a self-aware system will not be able to (1) infer what its code does based on measurements and (2) that it will not be able to be fully aware of itself, because the model would have to be at least as complex as the system (cf. law of requisite variety) but at the same time we need models that are abstract and analyzable. To conclude this part of the discussion, we identified the need to more closely study the theory of self-aware systems, their theoretical capabilities and limitations.

Participants

- Tarek F. Abdelzaher
Univ. of Illinois – Urbana, US
- Artur Andrzejak
Universität Heidelberg, DE
- Christian Becker
Universität Mannheim, DE
- Steffen Becker
TU Chemnitz, DE
- Kirstie Bellman
The Aerospace Corp. – Los Angeles, US
- Nelly Bencomo
Aston Univ. – Birmingham, GB
- Sara Bouchenak
University of Grenoble, FR
- Javier Camara
Carnegie Mellon University – Pittsburgh, US
- Giuliano Casale
Imperial College London, GB
- Lydia Y. Chen
IBM Res. GmbH – Zürich, CH
- Ada Diaconescu
Telecom Paris Tech, FR
- Lukas Esterle
Universität Klagenfurt, AT
- Antonio Filieri
Universität Stuttgart, DE
- Francesco Gallo
University of L'Aquila, IT
- Kurt Geihs
Universität Kassel, DE
- Holger Giese
Hasso-Plattner-Institut – Potsdam, DE
- Sebastian Götz
TU Dresden, DE
- Lars Grunske
Universität Stuttgart, DE
- Henry Hoffmann
University of Chicago, US
- Paola Inverardi
University of L'Aquila, IT
- Alexandru Iosup
TU Delft, NL
- K. R. Jayaram
IBM TJ Watson Research Center – Yorktown Heights, US
- Evangelia Kalyvianaki
City University – London, GB
- Joost-Pieter Katoen
RWTH Aachen, DE
- Jeffrey O. Kephart
IBM TJ Watson Research Center – Yorktown Heights, US
- Samuel Kounev
Universität Würzburg, DE
- Anne Koziolek
KIT – Karlsruher Institut für Technologie, DE
- Heiko Koziolek
ABB AG Forschungszentrum Deutschland – Ladenburg, DE
- Marta Kwiatkowska
University of Oxford, GB
- Philippe Lalanda
Université de Grenoble, FR
- Chris Landauer
The Aerospace Corporation – Los Angeles, US
- Peter Lewis
Aston Univ. – Birmingham, GB
- Martina Maggio
Lund University, SE
- Ole Mengshoel
Carnegie Mellon University – Silicon Valley, US
- Arif Merchant
Google Inc. – Mountain View, US
- Aleksandar Milenkoski
Universität Würzburg, DE
- Anders Robertsson
Lund University, SE
- Felix Maximilian Roth
Universität Mannheim, DE
- Hartmut Schmeck
KIT – Karlsruher Institut für Technologie, DE
- Evgenia Smirni
College of William and Mary – Williamsburg, US
- Katinka Wolter
FU Berlin, DE
- Xin Yao
University of Birmingham, GB
- Xiaoyun Zhu
VMWare, Inc. – Palo Alto, US
- Andrea Zisman
The Open University – Milton Keynes, GB

Report from Dagstuhl Seminar 15042

Coalgebraic Semantics of Reflexive Economics

Edited by

Samson Abramsky[1], Alexander Kurz[2], Pierre Lescanne[3], and Viktor Winschel[4]

1 **University of Oxford, GB,** samson.abramsky@cs.ox.ac.uk
2 **University of Leicester, GB,** ak155@le.ac.uk
3 **ENS – Lyon, FR,** Pierre.Lescanne@ens-lyon.fr
4 **Universität Mannheim, DE,** viktor.winschel@gmail.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 15042 "Coalgebraic Semantics of Reflexive Economics".

Seminar January 18–21, 2015 – http://www.dagstuhl.de/15042
1998 ACM Subject Classification D.3.1 [Programming Languages] Formal Definitions and Theory, F.4.3 [Mathematical Logic and Formal Languages] Formal Languages, J.4 Social and Behavorial Sciences – Economics
Keywords and phrases Programming language semantics, Coalgebra, Category theory, Economics, Epistemic game theory
Digital Object Identifier 10.4230/DagRep.5.1.197
Edited in cooperation with Jules Hedges (Queen Mary University of London, GB)

1 Executive Summary

Samson Abramsky
Jules Hedges
Alexander Kurz
Pierre Lescanne
Viktor Winschel

A growing number of researchers have been discovering analogies in the foundations of both computer science and economics. The goal of this seminar is to interface computer science with economics and game theory and to take advantage of the programming language semantics methods in theoretical computer science based on lambda calculus, coalgebras, modal logic and category theory.

The theoretical thread of interest to this seminar and common to both computer science and economics is the phenomenon that may be circumscribed by notions such as reflexivity, self-reference, impredicativity, infinite regress, recursion, or fixed points.

In computer science, the phenomena of self-reference, self-application and recursion played a crucial role in the foundational work of Gödel, Church, Turing and Kleene in the 1930s. Nevertheless, powerful mathematical models of the semantics of recursion became available only with the work of Scott on models of the untyped lambda calculus and subsequent research in domain theory. The combination of domain theory with the theory of types in programming languages and their categorical semantics has led to the development of a

powerful tool box. More recently, this tool box has been further strengthened by advances in coalgebra. It provides for a wide variety of dynamic systems the mathematical tools of (bi)simulation and coinduction as well as a variety of techniques from category theory.

In economics, and the social sciences more generally, reflexivity arises from the obvious fact that cognitive agents reason about themselves, others and the society they live in. This leads to self-reference and recursion in, for example, theories of belief formation as beliefs of beliefs (Harsanyi type spaces) or theories of institutions as rules to change rules. More generally, a social system consists of individuals who are learning about a process in which others are learning as well. Learning the state of an interactive system is therefore rather different compared to learning the parameter values that govern a physical process. When the observer is a part of the system, the act of learning changes the thing to be learned. The traditional mathematical tools in economics are hardly suited to solve these problems in a sufficiently general way and they make it difficult for computer scientists, once they need to solve similar or common problems, to understand the problem formulation and the solutions already achieved by economists.

The specific subfields of computer science and economics discussed above suggest to explore methods of program semantics and category theory in general and, in particular, of bisimulation and coinduction in economics. Furthermore, coalgebra gained prominence as providing models for concurrency, a topic that has hardly been touched upon in economics explicitly, even so it underlies the most general kind of issues in economics, namely those regarding centralization versus decentralization in theories of economic systems, administration, firms and markets.

Particular topics in which we see scope for methods from the semantics of programming languages include infinitely repeated games, econometrics and system theory, epistemic game theory and interactive learning in multi-agent systems.

More generally, research in program semantics and logics in computer science is typically motivated by problems arising in programming languages and software engineering. In one direction, economic modeling will become more important in software engineering. In the other direction, computational economics may as well profit from a modern approach to language design not only in terms of reflexivity at the theoretical level but also at the practical level of modeling software.

2 Table of Contents

3 Abstracts of the Workshops

3.1 Category Theory 2

Neil Ghani (University of Strathclyde, GB)

In Category Theory 2, students will have learned about basic category theory and in particular its core definitions of category, functor, natural transformation, limit and colimit. This will have given them a feel for what category theory is all about. However, the key technical tool that category theory has contributed to mathematics and the every day practice of the working category theorist is the notion of an adjunction. In this lecture we will introduce the concept of an adjunction, cover the different presentations of this concept and cover examples, both elementary and more significant.

3.2 Rationality

Pierre Lescanne (ENS – Lyon, FR)

This session was different in format to the others, being a free-form group discussion rather than a presentation by a speaker. It was spontaneously organised after an intriguing comment by Pierre Lescanne in a previous session. The session was based on the question of which logics are suitable for modelling the reasoning processes of economic agents, and in particular whether intuitionistic logic (in which every proof of an existential statement must include an explicit 'witness' as justification) could be better suited than the more usual classical logic (which allows 'pure existence proofs'). Part of the discussion focussed on operational questions of proof search, and whether logics for reasoning agents should be chosen for the efficiency of reasoning rather than expressive or deductive power. This is linked to discussions in the economics literature on models of bounded rationality as well as models in behavioral economics, but with the modern focus on proof search as a model of computation.

3.3 Coalgebraic Games

Achim Blumensath (TU Darmstadt, DE)

This workshop presented a new abstract framework designed to compose economic games. It is sufficiently general in order to represent all types of games encountered in economics. The framework is based on category theoretical techniques and coalgebras. Coalgebras have been successfully used to model the observable behavior of systems with an unobservable state space. The fundamental notion of a process is represented as a coalgebra of a certain type. It is shown how to use them in order to describe stage games, finitely, infinitely and potentially infinitely repeated games with imperfect and incomplete information based on deterministic,

non-deterministic or probabilistic decisions of observing agents in endogenous networks. The framework allows to compose games sequentially, in parallel and hierarchically and it provides a formal account of the behavior of the aggregated system. The games are directly implementable in high-level functional programing languages. The abstract mathematics of this approach links economics to the latest developments in mathematical game theory and theoretical computer science. Coalgebras arise as functorial fixed points that open the door to self-referential and reflexive structures that underly the Lucas critique, institutional economics, belief formation and many other social modeling issues.

3.4 Harsanyi Type Spaces as Coalgebras

Elias Tsakas (Maastricht University, NL)

A belief hierarchy is a description of an agent's beliefs about some fundamental space of uncertainty, her beliefs about everybody else's beliefs, and so on. During the past few decades, belief hierarchies have become an integral tool of modern economic theory, often used to analyze games with incomplete information (Harsanyi, 1967-68), as well as in order to provide epistemic characterizations for several solution concepts, such as rationalizability (Brandenburger and Dekel, 1987; Tan and Werlang, 1988), Nash equilibrium (Aumann and Brandenburger, 1995), and correlated equilibrium (Aumann, 1987), just to mention a few. Belief hierarchies are in general very complex objects, consisting of infinite sequences of probability measures. This makes them in principle very hard to handle and sometimes even to describe, especially when it comes to high order beliefs. Having recognized this difficulty, Harsanyi (1967-68) proposed an indirect Bayesian representation of belief hierarchies, known as the type space model. Formally, Harsanyi's model consists of a set of types for each agent and a continuous mapping from each type to the corresponding conditional beliefs over the product of the fundamental space of uncertainty and the opponent's type space. This structure induces a belief hierarchy for every type, thus reducing the infinite-dimensional regression of beliefs to a single-dimensional type. Mertens and Zamir (1985) and Brandenburger and Dekel (1993) completed the analysis by showing the existence of the universal type space, which represents all belief hierarchies satisfying some standard coherency properties. When the fundamental space of uncertainty is a Polish space, the universal type space is terminal in the category of type spaces. Recently it was shown that the infinite hierarchies of beliefs can be represented as coalgebras.

3.5 Coalgebras and Algebras in Scientific Modelling

Baltasar Trancon y Widemann (TU Ilmenau, DE)

The coalgebraic approach to the semantics and behavior of composed systems provides a complementary view on complex systems. The traditional scientific method is based on an algebraic approach where a system is decomposed into and composed from parts. This constitutes a point of view which is based on a constructive rather than a behavioral or

observational specification as in the coalgebraic methodology. The algebraic point of view is emphasizing the "doing" while the coalgebraic point of view is emphasizing the "seeing" aspect of a scientific explanation. Accordingly, it is only within the algebraic approach where the notion of emerging properties is needed in order to capture properties at the system level. The behavioral, coalgebraic approach does capture the system properties by instead deriving the behaviour of the parts from the behaviour of the system. The interaction of the parts is captured by the syntax or their allowed composition. However, it is important to see that coalgebras dualise and not supersede the traditional algebraic approach to modelling.

3.6 The Cohomology of Non-Locality and Contextuality

Samson Abramsky (University of Oxford, GB)

In a paper of Samson Abramsky and Adam Brandenburger, sheaf theory was used in order to analyze the structure of non-locality and contextuality. Moreover, on the basis of this formulation, it can be shown that the phenomena of non-locality and contextuality can be characterized precisely in terms of obstructions to the existence of global sections. The aim in the presented work is to build on these results, and to use the powerful tools of sheaf cohomology to study the structure of non-locality and contextuality. The Cech cohomology is used on an abelian presheaf derived from the support of a probabilistic model, viewed as a compatible family of distributions, in order to define a cohomological obstruction for the family as a certain cohomology class. This class vanishes if the family has a global section. Thus the non-vanishing of the obstruction provides a sufficient (but not necessary) condition for the model to be contextual. It is shown that for a number of salient examples, including PR boxes, GHZ states, the PeresMermin magic square, and the 18-vector configuration due to Cabello et al. giving a proof of the Kochen-Specker theorem in four dimensions, the obstruction does not vanish, thus yielding cohomological witnesses for contextuality.

3.7 Higher Order Games

Jules Hedges (Queen Mary University of London, GB)

The study of higher order games comes out of the proof-theoretic work on selection functions by Martin Escardó and Paulo Oliva. This talk focussed on a single worked example, namely that replacing utility maximisation with a fixpoint operator models agents who attempt to coordinate with the majority (proving a simple model of the so-called Keynes beauty contest), and similarly an anti-fixpoint operator models agents who attempt to differentiate from the majority. However this is a quickly evolving area with several directions being investigated. For example higher order games allow the use of monads, a concept from category theory used to model side-effects in programming languages, to also model economic 'side-effects' such as probabilistic choices, learning and external effects. Another direction, using ideas from the theory of control operators in programming languages, leads to a fully compositional game theory that should be very useful in complex practical examples (using a category

whose morphisms are suitably generalised games) and a graphical notation based on string diagrams in quantum information theory. Another idea is to use ideas based on Escardó's 'seemingly impossible functional programs' together with computable analysis to compute solutions of games. Finally there is also a hope that higher order games could serve as a common structure between several other theories, including some described in this report, allowing them to be used together.

4 Outlook

There have been various discussions about the possible paths for the research to be done in the future. The mathematical research for the economic applications is to be done mainly in the fields of category theory, type theory and coalgebras. The applications can be characterized as those with a high model complexity, either by being large models or irregular by being composed from various parts that can not be glued easily by traditional methods. A first effort is to give the current state of research in game theory a semantic and a composable representation.

4.1 Work on Reflexive Economics

A semantical approach to equilibria and rationality, 2009, by Dusko Pavlovic: This is the first paper that is taking advantage of the semantical approach to programming languages and uses it within the realm of economics and game theory.

On the Rationality of Escalation, 2010, by Pierre Lescanne, Perrinel Matthieu: Here it is argued that speculative bubbles are rational if viewed from a coalgebraic point of view.

Coalgebraic Analysis of Subgame-perfect Equilibria in Infinite Games without Discounting, 2013, by Samson Abramsky and Viktor Winschel: In this paper we prove coinductively a property of an infinite game, namely subgame perfectness of a strategy profile.

A Coalgebraic Framework for Games in Economics, 2013, Achim Blumensath, Viktor Winschel: This paper presents a complete coalgebraic framework for composing complex games from simple ones. The glueing is done by natural transformations on deterministic, nondeterministic or probabilistic choice functors.

A Higher-order Framework for Decision Problems and Games, 2014, by Jules Hedges, Paulo Oliva, Evguenia Winschel, Viktor Winschel, Philipp Zahn: This framework, again fully compositional, is build on higher-order functions and provides a high-level language to formulate goals or preferences that are context dependent and that traditionally need to be modelled by transforming the outcome space for representing behavior as utility maximization.

4.2 Research

An important extension and unification of current work is to unite the higher-order games and the coalgebraic games. The coalgebraic games so far lack a generalized or higher-order approach to preferences as the higher-order games do. And the higher-order game lack so far a coalgebraic treatment of infinite games that are essential to many economic applications.

Both approaches provide compositionality and it is interesting how both approaches unite within a category of games.

Having defined coalgebraic structures within game theory it is interesting to extend it to modal logic. This could serve as a general approach to define all kinds of predicates on games that are of interest during the modelling process, with equilibrium predicates being one of many other possible ones. The main modalities of interest in economics are epistemic and temporal modalities. However, other could be useful as well like deontic ones.

Another interesting path of research is to include inductive reasoning within a game theoretical context. Either as learning agents within the very game itself or as tools for the modeller to expose the game to data for an empirical evaluation of the model.

Both applications have to rely on some form of statistical analysis. The classical one would take the axiomatic approach, which can also be seen from a categorical point of view using the Giry monad and convex sets. The alternative algorithmic approach could take advantage of basing induction on the same footing as the approach in the higher-order games. Namely representing the units of modelling, like games, players, strategies or equilibria, as an algorithm or higher-order function. In an algorithmic statistical approach the units of induction are as well algorithms with the advantage that induction takes place at the language level of the modeling tools and not as usual as in classical statistics at the level of parameters of some function representing some economic concept.

An important path for research is finally the refinement of equilibrium solvers. So far there has been some progress to check for equilibria for some given strategy profiles in arbitrarily composed games. But there is hardly a universal approach to be expected to be able to find automatically equilibria in all kinds of games.

Within the quest for game solvers it might be useful to use some traditional numerical approaches like function approximation or some modern variants of it like sparse grids. But also there might be scope for an application of the seemingly impossible functional algorithms of Martin Escardo and in general infinite precision algorithms.

4.3 Industrial Applications

There are various industrial applications possible for the research on the semantics of games.

A large application of a compositional game theory is the field of smart energy grids where large and heterogeneous networks of decentralized reasoner are to be modelled and implemented.

The application of behavioral economics that can be unified under the higher-order games approach are marketing applications or online recommendation systems that need to be based on non-optimizing or heuristic reasoning, learning approaches and software implementation issues for online surveys.

The tools of compositionality that lend themselves into suitable approaches to network theory do also allow for a unified approach to the theory of optimal currency areas and money theory.

Another highly profitable and important area of application for compositional games is industrial organization. Here, questions of the utility of merging companies is of interest to the companies themselves but also to courts and institutions surveying the competition or monopoly developments in markets. Within this application a formal modelling approach is already established.

Finally, a compositional and implementable approach to game theory can be used to build

Table 1 Workshops.

Tutorials	**Monday**		**Tuesday**	**Wednesday**
09:00–10:30	Category Theory (Samson Abramsky)		Coalgebras and Semantics (Alexander Kurz)	Theory of Science: Algebraic and Coalgebraic Point of View (Michael Hauhs)
11:00–12:30	Microeconomics and Game Theory (Philipp Zahn)		Macroeconomics (Viktor Winschel)	
Workshops 13:30–14:30	**Monday** Category Theory 2 (Neil Ghani)	Rationality (Pierre Lescanne)	**Tuesday** Coalgebras and Algebras in Scientific Modelling (Baltasar Trancon y Widemann)	
15:00–16:30	Coalgebraic Games (Achim Blumensath)		Contextuality and Locality by Sheafs and Cohomology (Samson Abramsky)	
17:00–18:00	Harsanyi Type Spaces as Coalgebras (Elias Tsakas)		Higher Order Games (Jules Hedges)	

education software for universities, massive online courses or MBA schools where students of strategic decision making are educated. Here a software where actual players or computer players interact in order to train decision makers can be of much interest.

An essential goal for the emerging field of the semantics of economics is the provision of software. The software will be a compiler for some high-level language that is tailored to the domain of economics such that modelling and programming can take place as close as possible to the high-level concepts in narrative and natural language economics.

5 Schedule

We have organized this interdisciplinary workshop to encompass two parts, the morning and the afternoon sessions as shown in Table 1.

In the morning there have been presentations from both researchers of computer science and economics. Each group gave lectures in order to introduce the other group to the basic methodological tools in the two sciences and the basic problems that are addressed within. The economics talks were divided in microeconomics and macroeconomics. The computer scientific presentations introduced the audience to category theory and to coalgebras and their use in the semantics of programming languages. Also, there was a methodological talk on the scientific method from an algebraic and a coalgebraic point of view.

In the afternoons, the second part of the seminar, we had several workshops that introduced already existing work within our emerging field of reflexivity and semantics of economics and game theory.

Participants

- Samson Abramsky
University of Oxford, GB
- Achim Blumensath
TU Darmstadt, DE
- Filippo Bonchi
ENS – Lyon, FR
- Neil Ghani
University of Strathclyde, GB
- Helle Hvid Hansen
Radboud Univ. Nijmegen, NL
- Michael Hauhs
Universität Bayreuth, DE
- Julian Hedges
Queen Mary University of London, GB
- Alexander Kurz
University of Leicester, GB
- Stéphane Le Roux
University of Brussels, BE
- Pierre Lescanne
ENS – Lyon, FR
- Fabio Mogavero
University of Napoli, IT
- Paulo Oliva
Queen Mary University of London, GB
- Prakash Panangaden
McGill University, CA
- Daniela Petrisan
Radboud Univ. Nijmegen, NL
- Marcus Pivato
University of Cergy-Pontoise, FR
- Jan Rutten
CWI – Amsterdam, NL
- Martin Scheffel
Universität Köln, DE
- Heiner Schumacher
Aarhus University, DK
- Alexandra Silva
Radboud Univ. Nijmegen, NL
- Baltasar Trancon y Widemann
TU Ilmenau, DE
- Elias Tsakas
Maastricht University, NL
- Evguenia Winschel
Universität Mannheim, DE
- Viktor Winschel
Universität Mannheim, DE
- Philipp Zahn
Universität Mannheim, DE

Report from Dagstuhl Seminar 15051

Artificial and Computational Intelligence in Games: Integration

Edited by

Simon M. Lucas[1], Michael Mateas[2], Mike Preuss[3], Pieter Spronck[4], and Julian Togelius[5]

1 **University of Essex, GB,** sml@essex.ac.uk
2 **University of California – Santa Cruz, US,** michaelm@cs.ucsc.edu
3 **TU Dortmund – Dortmund, DE,** mike.preuss@cs.uni-dortmund.de
4 **Tilburg University, NL,** p.spronck@uvt.nl
5 **New York University, US,** julian.togelius@gmail.com

Abstract

This report documents Dagstuhl Seminar 15051 "Artificial and Computational Intelligence in Games: Integration". The focus of the seminar was on the computational techniques used to create, enhance, and improve the experiences of humans interacting with and within virtual environments. Different researchers in this field have different goals, including developing and testing new AI methods, creating interesting and believable non-player characters, improving the game production pipeline, studying game design through computational means, and understanding players and patterns of interaction. In recent years it has become increasingly clear that many of the research goals in the field require a multidisciplinary approach, or at least a combination of techniques that, in the past, were considered separate research topics. The goal of the seminar was to explicitly take the first steps along this path of integration, and investigate which topics and techniques would benefit most from collaboration, how collaboration could be shaped, and which new research questions may potentially be answered.

Seminar January 25–30, 2015 – http://www.dagstuhl.de/15051
1998 ACM Subject Classification I.2.1 Artificial Intelligence – Games
Keywords and phrases Multi-agent systems, Dynamical systems, Entertainment modeling, Player satisfaction, Game design, Serious games, Game theory
Digital Object Identifier 10.4230/DagRep.5.1.207

1 Executive Summary

Simon Lucas
Michael Mateas
Mike Preuss
Pieter Spronck
Julian Togelius

The research field of artificial and computational intelligence in games focuses on the wide variety of advanced computational techniques used to create, enhance, and improve the experiences of humans interacting with and within virtual environments. By its nature the field is broad and multidisciplinary. People working in it include academic researchers from a

Artificial and Computational Intelligence in Games: Integration, *Dagstuhl Reports*, Vol. 5, Issue 1, pp. 207–242
Editors: Simon M. Lucas, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

variety of disciplines, corporate researchers in the games industry as well as in other industries, and independent game developers. The methods used include symbolic AI techniques such as reasoning, constraint-solving and partial-order planning as well as biologically-inspired techniques such as evolutionary computation and neural networks, statistical techniques such as support vector machines and clustering, as well as special-purpose techniques such as behavior trees. These are applied to games ranging from board and card games to first-person shooters and real-time strategy games as well as abstract mathematical games, and are used to, for instance, play games, model players, tell stories, generate levels and other game content, and match players. Different researchers have different goals, including developing and testing new AI methods, creating interesting and believable non-player characters, improving the game production pipeline, studying game design through computational means, and understanding players and patterns of interaction. Often several goals overlap in the same project.

Recent years have seen considerable progress in several of the techniques used in the field, as well as rapid development in what kind of research questions are asked and what kind of games are studied with which methods. It has become increasingly clear that many of the research goals require a multidisciplinary approach, or at least a combination of techniques that, in the past, were considered separate research topics. For instance, with respect to the behavior of virtual agents, ten years ago researchers mainly aimed at making such behavior more "effective," which can often be achieved with straightforward computational reasoning. Nowadays, however, researchers aim at making the behavior of virtual agents more "natural" in their interaction with humans, which requires contributions not only from computer science, but also from psychology and social sciences, and which requires a wide variety of techniques, such as player modeling, adaptation, reasoning, and computational linguistics.

To move the research field forward, it is therefore of crucial importance to facilitate the integration of the disciplines and techniques that are involved in this research. The various strands, methodological approaches, and research directions need to inform each other and collaborate, to achieve a whole that is more than the sum of its parts. The goal of the second *Dagstuhl Seminar on Computational and Artificial Intelligence in Games* was to explicitly take the first steps along this path of integration, and investigate which topics and techniques would benefit most from collaboration, how collaboration could be shaped, and which new research questions may potentially be answered.

The seminar was held between January 25 and January 30, 2015. To stimulate interaction between the participants, which is essential in this context, the seminar was structured around workgroups rather than presentations. The organizers started the seminar on Monday morning with a series of brief presentations on potential discussion topics, after which the participants formed their own workgroups around a variety of topics, not necessarily those brought up by the organizers. Workgroups typically consisted of 3 to 10 people from different backgrounds, who worked together for no more than one day. At regular intervals workgroups reported on their findings in a plenary session, after which new workgroups were formed.

At the start of the seminar it was announced that Thursday would be set aside for practical work. Participants could use that day to implement some of the ideas that had come up in the previous days, in the form of a game, a competition, a design document, or a research proposal. While the organizers deliberately gave the participants the option to simply continue with the workgroups if they so wished, all participants actually got involved in the practical work, some of them even working on multiple projects in parallel.

The results of the workgroups and the practical sessions are briefly related in the remainder of these proceedings. The 13 abstracts on workgroups cover automated and AI-based game

design; game analytics; interdisciplinary research methods; design of believable characters; general video game playing; creativity facet orchestration; methods and formal design for procedural content generation; design of "fun" gameplaying bots; communication on game AI research, computers that play like humans; and neural networks for games. The 11 abstracts on practical sessions cover the Planet Wars competition; the automatic generation of games, mazes, and text; Twitter bots; sonification of character reasoning; MCTS and representation learning for procedural content generation; two AI-based games; and the design for a board game.

A special issue of the *IEEE Transactions on Computational and Artificial Intelligence in Games* will be published on the topic of this Dagstuhl Seminar. While this issue is open for submission for any researcher in this field, it is expected that several of the workgroups of the seminar will submit papers on their results.

As organizers, we knew that the first seminar that we organized in 2012 was considered a great success, and we had expected more people to accept our invitations for this second seminar than for the previous one. However, demand for attending the seminar was even greater than we expected. Almost everyone we first invited immediately accepted our invitation. Moreover, everybody who accepted their invitation indeed showed up at the seminar. We were forced by capacity concerns to not invite many people who, by their strength of contribution in the field, should have been present. We are certain that we could easily have doubled the number of participants visiting the seminar, and that each of those participants would have made a strong contribution.

The value of these Dagstuhl Seminars is indisputable. Considering the large number of researchers that should be invited to a seminar that attempts to cover the whole, very broad research field of Computational and Artificial Intelligence in Games, we believe that it is wise for a future seminar to narrow down the topic, so that it can be restricted to a smaller number of participants that are active in the selected subfield. Naturally, considering the fact that "integration" is such an important issue in the research field, care must be taken to ensure that every discipline interested in and involved in the subfield is represented.

2 Table of Contents

3 Workgroups

3.1 Bold Automatic Game Design

Dan Ashlock (University of Guelph, CA)

Joint work of Ashlock, Daniel; Colton, Simon; Eladhari, Mirjam; Van Kreveld, Marc; Lanzi, Pier Luca; Sipper Moshe

"Bold Automatic Game Design" was the title of a session at Dagstuhl Seminar 15051 intended to explore farther reaching and more daring approaches to automatic game design. This abstract summarizes the recommendations and debating points located by the participants. These include downloading some of the design to the players, incorporating environmental information into games, the need for agents to behave irrationally, the need for difficulty estimation and adjustment with a game, the value of explicit versus implicit tutorial levels within a game, and regularizing the presentation of automatically generated content for automatic game design.

Automatic Game Design. This session identified a number of issues, some contentious and some consensus, in the area of automatic game design. Automatic game design is any process that automates some or all of the process of producing a new game. At its weakest is consists in a model for assembling available game elements into a particular instance of a particular type of game. At its strongest it represents a version of solving the strong AI software in which a software system comes up the the mechanics and details of a game and writes the code to implement it.

Leveraging Players as Data. The group felt that the ability to incorporate and leverage the behavior of players was a capability that should be explicitly left into, or at least left in as an option, for automatically designed games. This can take many forms from player modeling intended to match the player with an enjoyable game experience to using player behavior and actions to trigger subsequent branches of the game design process.

Location Based Games. For mobile devices with GPS capability there is potential for incorporating information about the player's current location into games. Since the supply of locations is quite large, this in turn makes games incorporating location information natural targets for partially or completely automatic game design.

Irrational Agents. There are many definitions of "rational" behavior from survival to the accumulation of treasure. Agents that behave according to a single notion of rationality are, potentially, predictable. Agents that change their notion of rationality or, in fact, sometimes act irrationally or from obscure motives can enhance the game experience by reducing the degree to which game agents and automata become predictable.

Automatic Difficulty Adjustment. The problem of adjusting game difficulty to a player's skill level is a current, active research topic outside of automatic game design. The group believes that best-practice dynamic difficulty techniques should be part of any automatic game design system.

Tutorial Levels: explicit or implicit. It is sometimes practical to leave players to work out game mechanics for themselves. At the other extreme there are games that come with novel length instruction booklets or files. Between these extremes lies the in game tutorial. The group felt levels designed to introduce and school a player in a given game mechanic were

superior to more explicit techniques like text screens or pop-ups. We thus recommend that levels introducing game mechanics and, when distinct game mechanics interact, introducing them as a group be something that an automatic game design system can produce.

Snap Games. The idea of small simple games that vanish if not played within a given time limit, in analogy to SnapChat, might be an interesting venue for automatic game design. The impermanence of such games almost demands automatic generation and such games could reasonably be small enough not to be too challenging as a test environment for automatic game design.

Element Libraries or Generators. There is a great deal of current research on automatic content generation (ACG). Any automatic game design system should have access to and incorporate appropriate ACG material. This suggests that VGL or some similar standard should be one of the possible output methods for ACG software.

Leveraging Mathematical Objects. This subject was somewhat contentious. Vector spaces, Cayley graph based on finite groups, and other mathematical objects can be used to generate level maps or terrain maps with complex but controllable properties. Some members of the group felt that understanding advanced mathematics would be too great an effort for game designers and that such techniques were complex enough to defy encapsulation by experts. Since this is a debate subject to resolution by research, some members of the group have undertaken to investigate the issue and publish the results.

3.2 Analytics and Game AI

Christian Bauckhage (Fraunhofer IAIS – St. Augustin, DE)

Joint work of Bauckhage, Christian; Bouzy, Bruno; Buro, Michael, Cowling, Peter; Kendall, Graham; Lanzi, Pier Luca; Lucas, Simon; Samothrakis, Spyridon; Schaul, Tom; Spronck, Pieter; Winands, Mark

The goal of this session was to fathom how or in how far modern data science can be integrated into game design and game development so as to assist in these processes. A particular focus was put on analytics for game AI and the session attempted to structure the landscape of ideas in this area.

Correspondingly, participants set out to discuss what analytics for game AI could be, why it would be helpful, and how it can be accomplished? Key points that were identified as general paradigms or research directions for game analytics included learning from human game play, learning from self play, i.e., reinforcement learning like approaches for AI algorithms, analyzing game results, e.g., game tree structures, and the problem of understanding player behaviors.

Regarding the question as to why corresponding research would be beneficial for game AI programming, participants agreed that it could help to improve player experiences, automatize game bot programming, automatize game design, develop actionable metrics for game analysis, and balance games.

In order to accomplish these goals, methods to extract direct and indirect features would be needed which in turn would allow for the application analytic methods. In this context, the term direct features refers to measurements or observations such as durations of games, actions taken by players, or information available during play. Indirect or derived features, on the other hand, refer to measures of, say, skill depth, frustration, addictiveness, or even

game aesthetics. Once a game has been played by one or many players and features have become available, simple statistics as well as more elaborate methods could be used to look for patterns in the data. Examples of more elaborate techniques that were reviewed in detail included ELO-like measures, cluster analysis, game tree analysis, A/B testing, and neural networks.

Participants then discussed industrial and academic objectives for game analytics. With respect to industrial interests in game data mining, they identified the goals of maximizing revenue, maximizing retention, maximizing the number of premium players in freemium games, maximizing player experience (e.g., through balancing and content adaptation) as well as the goal of automatically maximizing the quality of software. Academic objectives, on the other hand, appeared to be more closely related to basic AI research and included progressing towards human-like AI through behavior analysis, automatizing map layouts and content generation, as well as the question of how to "gamify" scientific problems so that games could be used more easily in genetics research (protein folding) or in simulations for financial and social research.

In the second part of the session, participant decided to put these ideas into practice. To this end, the following games were chosen to guide the discussion in smaller working groups:

- "planet wars", a simple RTS game of perfect information
- "resistance", a multi-player card game of imperfect information
- "retry", a simple platform game
- "skiddy", a simple puzzle game.

For each of these games, the group identified direct and indirect features as well as methods that would allow for their analysis. In particular, the game "planet wars" was then considered for further practical studies. Participants set out to program simple rule-based as well as more advanced neural network and Monte Carlo tree search (MCTS) based game bots for this game whose parameterizations were informed by or learned from game logs. In addition, practical experiments in behavior modeling and analysis were carried out where behaviors were represented in terms of sequences of prototypical actions taken by either human players or game bots. In preliminary results it was found, that even simple learning-based approaches, i.e., an ensemble of linear perceptrons, lead to more engaging game bots and that even simple behavioral models are capable of distinguishing styles of game play and can thus provide hints for the dynamic adaptation of bot behavior. Training of a deep neural network for "planet wars" and the implementation of an MCTS bot were begun during the seminar but could not yet be evaluated; however, results are expected to be available and published soon.

In conclusion, the practical work done in this session demonstrated in an exemplary manner that the integration of game analytics into game development can indeed contribute to the general goals and objectives identified above. This confirms that the idea of game analytics is an auspicious direction for future research and, in the concluding session of the seminar, the participants of the Analytics and Game AI working group could raise awareness for this emerging topic among the other participants of the seminar.

3.3 Interdisciplinary Research Methods

Ian Horswill (Northwestern University – Evanston, US)

Joint work of Horswill, Ian; Mateas, Michael; McCoy, Josh; Paiva, Ana; Rudolph, Günther; Smith, Gillian; Young, Michael; Zhu, Jichen

Game AI research is inherently interdisciplinary, involving not only computer science, but also design, psychology, sociology, graphic design, theater, filmmaking, creative writing, and a host of other areas. A system can be successful from a technical computer science standpoint, while still being an abject failure as a piece of art or entertainment. There is thus an open question as to the appropriate ways to evaluate Game AI research contributions.

While the traditional evaluation methodologies of computer science, including HCI methodologies such as controlled user studies, are undeniably useful, we will likely need to look to other disciplines, particularly in the arts and humanities, for methods for evaluating the aesthetic, thematic, or ideological dimensions of a work.

Evaluation of this kind of interdisciplinary work is complicated by the differing notions of problem adopted by the various disciplines. Engineers, including computer scientists, typically work on well-posed problems – problems with a clear formulation and natural performance metrics. Designers on the other hand, often work on so-called "wicked problems," ones without definitive formulations or metrics, and little ability to generalize solutions across problems. Artists may not even conceptualize their work as solving a problem, but merely as exploring an interesting region of design space to see what they encounter within it.

Another complication lies in the different intellectual and social purposes of evaluation. Although evaluation sections of papers are often written as if answering the question of how well a system worked, they are often read by reviewers and other gatekeepers as answering the question of how well the researchers worked, and thus, whether the work should be published. And yet, they might be most usefully written and read as trying to articulate what insights can be derived from the system for the guidance of future system builders.

We can find resources for evaluation from the component disciplines. The use of performance metrics and correctness proofs from engineering; quantitative methods such as survey instruments as well as qualitative methods such as ethnography from the social sciences; artist talks, critique sessions, and juried shows from the fine arts; postmortems and commentary tracks from the game industry; close reading and critical theory from the humanities; symbiotic interaction between critical reflection and algorithmic development from critical technical practice; and the use of the case study, common in areas such as medicine, law, and business.

However, we don't currently have good evaluation methods for characterizing many of the issues that come about in game AI, and acknowledge the need to create new evaluation methods. How do we evaluate architectures as opposed to systems? How do we evaluate generation systems in PCG? Or the authorial effort involved in using a particular system?

Understanding these issues is important to the future development of the field. We look forward to a rich dialog on evaluation, both within the field of game AI and in conversation with sibling disciplines from which it can learn.

3.4 GVGP Competition Revisited

John M. Levine (University of Strathclyde, GB)

Joint work of Ashlock, Dan; Levine, John; Ontañón, Santiago; Thawonmas, Ruck; Thompson, Tommy

During this Dagstuhl Seminar session, we analyzed the current status of the General Video Game Playing (GVGP) competition, discussed its potential biases toward some families of techniques, and produced a list of suggestions for future directions of the competition.

Throughout the history of artificial intelligence, competitions and challenge tasks have often driven a significant amount of the research being done in the field. Publicly available and open competition domains allow for: the consolidation of a relevant research problem, fair comparison of rival techniques, increase the visibility of research, attract new researchers to the field and are often incubators of new techniques. The GVGP competition is an effort to incubate general AI techniques for video games, and move away from game-specific competitions, which might result in solutions that are too tailored to specific domains.

Specifically, the GVGP competition provides a general framework where 2D, tile-based, single-agent, full-information video games can be specified via the Video Game Description Language (VGDL). A forward model is provided, with which entries can "simulate" sequences of actions in the world. The goal of the competition is to create algorithms that can play any video game defined using the VGDL. Simplified versions of a collection of classic video games are provided as "training" games, for researchers to test their algorithms in, and a hidden collection of "testing" games is used during the competition to compare the algorithms submitted.

The most common family of algorithms in this competition is game-tree search, with UCT being the algorithm underlying most submissions. Given this fact, the inclusion of a forward model in the competition seems to favor this types of algorithm. Other families of algorithms struggle in this setting: for example, reinforcement learning struggles with the large state space of these games and the fact that it needs to be trained for each new game; supervised learning cannot be applied in the absence of training data; heuristic search suffers sue to the difficulty of devising general heuristics for a large class of games; and evolutionary algorithms suffer from the lack of a training period.

Given the initial goals of the GVGP competition, we identified a set of lines for consideration in future editions of the competition:

- Forward model vs no-forward model tracks: the inclusion of a forward model favors game- tree techniques. Moreover, assuming the existence of such model is too strong an assumption in general, and thus, would prevent the competition from spurring development of general algorithms that can play new games for which such model is not available.
- Inclusion of training time: allowing an initial training phase would enable a larger diversity of algorithms to be applicable.
- Partial observability track: most real video games are partially observable, and thus algorithms that can work under such assumption could be encouraged with such a track.
- Adversarial track: all games in the GVGP are currently single player. However, multi player games introduce unique challenges that could result in interesting algorithmic developments.
- Team-coordination track: another interesting possibility would be to include a track containing team games where more than one agent need to collaborate together to play a game against another team of agents.

Finally, it is worth mentioning that the organizers acknowledged at the seminar that that some of these issues would be addressed. In fact, it has been recently announced that at the CIG 2015 competition, the organizers will be introducing two new tracks: Learning Track and Procedural Content Generation Track. Although their information has not yet been revealed at the time of writing this report, the former track apparently will allow a training phase.

3.5 Creativity Facet Orchestration: the Whys and the Hows

Antonios Liapis (IT University of Copenhagen, DK)

Joint work of Bidarra, Rafael; Liapis, Antonios; Nelson, Mark; Preuss, Mike; Yannakakis, Georgios

Creativity facet orchestration aims to combine generation across the multiple creative domains that comprise game design. Literature identifies six core facets existent in games: level design, game design, audio, visuals, narrative and gameplay (involving NPCs or not) [1]. The first two facets are *necessary* for a game to be instantiated whereas the remaining four are *optional*. While there have been a few attempts to integrate more than one facet during the generative process (e.g. Game-o-Matic [2], Angelina [3]) these have been limited to mere hierarchical (linear) procedures. It is only very recently that research on computational game creativity has focused on ways in which more that two facets are interweaved during their generation (such as the work of Hoover et al. [4] orchestrating visuals, audio and gameplay elements).

We argue that to generate novel and valuable games that lie on unexplored regions of the game design space, an *orchestration* approach is needed to automate game generation in a truly integrated manner. We view this approach as an *iterative refining process* particularly suited for the **generation of playable prototypes** for designers to consider and get inspired from. Orchestration requires that the human designer specifies the desired semantics for a query within a (large but manageable) space of possible games. For example, a designer might request a horror game (directly affecting the mechanics of the game), with open-space-style levels (affecting level design), with a warm ambiance (affecting visuals), relaxing music (affecting audio), linear narrative and aggressive NPCs. The generative system blends available concepts (using e.g. ConceptNet [5]) with the aim to deviate from the query in the game design space (e.g. this process could involve searching for novel games from semantically annotated databases of existing games). At this point each facet operates independently, constrained by the semantic information the designer has provided. The facet-specific generator operates using the semantics of all generators, thus providing a high-level context to guide its generative processes. The result is a set of prototypical games with unconventional combinations of facets' outputs, all matching the same underlying semantics. Those games are then presented for designer consideration, along with information about their distance (dissimilarity across several dimensions) to typical games, in order to choose which are to be refined. Refinement tailors parameters of the game space and polishes facets such as visuals and audio.

References

1 Antonios Liapis, Georgios N. Yannakakis, Julian Togelius. *Computational Game Creativity*. In Proceedings of the Fifth International Conference on Computational Creativity, 2014.

2 Michael Treanor, Bryan Blackford, Michael Mateas, Ian Bogost. *Game-o-matic: Generating videogames that represent ideas*. In Procedural Content Generation Workshop at the Foundations of Digital Games Conference, 2012.
3 Michael Cook, Simon Colton, Azalea Raad, Jeremy Gow. *Mechanic miner: Reflection-driven game mechanic discovery and level design*. In Proceedings of Applications of Evolutionary Computation, volume 7835, LNCS, 284–293, 2013.
4 Amy K. Hoover, William Cachia, Antonios Liapis, Georgios N. Yannakakis. *AudioInSpace: A Proof-of-Concept Exploring the Creative Fusion of Generative Audio, Visuals and Gameplay*. In Proceedings of Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMusArt), 2015.
5 Hugo Liu, Push Singh. *ConceptNet – A Practical Commonsense Reasoning Tool-Kit*. BT Technology Journal 22, 4, 211–226, 2004.

3.6 Believable Characters

Brian Magerko (Georgia Tech – Atlanta, Georgia, US)

Joint work of Champandard, Alex; Eladhari, Mirjam; Horswill, Ian; Magerko, Brian; McCoy, Josh; Spronck, Pieter; Treanor, Mike; Young, Michael; Zhu, Jichen

The goal of believable characters in computer games (and other media) is to achieve dramatic believability. In other words, "audiences do not pay for reality." This task group considered two views of the problem of creating dramatically believable characters, as has been considered in film, theater, and academic games research in decades prior: the requirements for "believable behavior" and the outstanding issues in computational architectures for creating said behavior.

As a driving example, we considered a specific dramatic scene of "asking someone out at a bar" and what perceivable individual and social cues were related to the dramatic content of the scene. These externally visible cues included: gaze, feeling, posture, proxemics, dialogue, and action selection & execution. The internal drives that could therefore enable these cues in the scene are functions like: anxiety, a theory of mind, a concept of social contract & cultural norms, character histories, and idiomatics.

The concept of procedural idiomatics seemed to be an avenue of particular interest. Within a system that explores the "asking out on a date" scene (or, conversely, the "Little Red Riding Hood" story that has been a prevalent example story in interactive narrative systems), one could consider automatically reasoning about narrative discourse, genre conventions and tropes, character archetypes, and character status as a means of exploring the scene in dramatically believable fashion with highly different dramatic results.

In terms of architectures, there are a number of important outstanding issues. One important issue is the mid-level consistency problem: arbitrating between high-level plans and low-level action selection so as to avoid trashing behavior. The authorial consistency problem is a related issue. While many character architectures can produce consistent and intelligent behavior within a given character, and high level narrative sequencers such as beat systems and drama managers can coordinate long-term narrative consistency, coordinating between the two systems in a way that produces sensible behavior is difficult. Finally, there is a problem with achieving thematic consistency: preventing low-level systems from choosing locally rational dramatically inappropriate action, such as a Sims character washing the dishes immediately after the death of a loved one.

3.7 ASP versus EAs: What Are We Really Searching For in PCG?

Adam M. Smith (University of Washington – Seattle, US)

Joint work of Nelson, Mark; Mateas, Michael; Smith, Adam; Stanley, Kenneth

Answer-set programming (ASP) and evolutionary algorithms (EAs) have emerged as powerful building blocks for search-intensive procedural content generation (PCG) systems. The two traditions (each with a rich history outside of PCG) imply very different problem formulations and, subsequently, very different algorithmic approaches. With the intent to provide guidance to the PCG research community, a working group with representative experts from both traditions was assembled to assess the alternatives. The surprising outcome was that, despite the revelation that neither tradition was particularly well aligned with the needs of PCG systems, both traditions offered oft-undiscussed variations that mapped well to PCG needs.

In the discussion, ASP stood as the representative for a broad class of approaches based on applying exhaustive search to constraint satisfaction problems (CSPs). In such formulations, the goal is to find any solution that violates zero constraints or terminate with a proof that no solutions exist. Typically, these approaches search choice-by-choice, making incremental elaborations to a partially-defined solution, pruning subspaces of possibilities on the basis declaratively-specified constraints. Despite the strong guarantees that can be made about these algorithms and their outputs when they are run to completion (in finite time), little clues are offered to practitioners about what happens when they are forcibly terminated early (as is sometimes needed in practical systems). Additionally, PCG practitioners often struggle with formulating their design concerns as declarative constraints.

EAs, on the other hand, represented approaches that apply stochastic local search to optimization problems. In these formulations, the goal is to find the best solution possible in an anytime fashion. Typically, those approaches follow a generate-and-test paradigm where complete candidate solutions are repeatedly constructed and then assessed (often to produce a single fitness score). The most promising candidate solutions usually become the prototypes for the next round of candidates. Because these approaches treat the assessment of candidates as a black-box process, PCG practitioners are offered significant flexibility in how they formalize optimization criteria (however these criteria must sometimes be perturbed in order to achieve acceptable search performance). Owing to the anytime formulation, it is difficult to characterize if and when a given type of solution will be produced. Nevertheless, the stream of candidate solutions provides observers with a clear sense of progress.

Understanding the typical properties of these two traditions does not immediately clarify the situation in PCG for two reasons. First, there exists systems which subvert these expectations; there are constraint solvers that use stochastic local search and exhaustive global optimizers. Second, most PCG systems do not simply want the first feasible solution or the best-scoring solution, they seek collections of content artifacts that faithfully represent the solution space. Neither problem formulation directly expresses our interest in producing a diverse archive of high-quality solutions (from which solutions can be later selected on-the-fly without re-running the resource-intensive search process).

In *Automatically Categorizing Procedurally Generated Content for Collecting Games*, Risi et al. [1] described how they used a self-organizing map (SOM) to produce a visually intuitive, finite categorization of the space of generated flowers in the *Petalz* game. Their most effective SOM grouped content according to emergent (phenotypic) features rather than defining (genotypic) features, and supported a novel game mechanic where players

used an open-ended content generator (based on EAs) to incrementally explore the finite categorization.

In *Automatic Game Progression Design through Analysis of Solution Features*, Butler et al. [2] generated "a large and diverse database of puzzles" for *Refraction* (using ASP) by grouping those puzzles according to properties of player- constructed solutions those puzzles implied (an emergent property). The creation of this diverse database directly powered their online progression generation system which systematically introduced the player to every combination of concepts that were present in the database (allowing progression design to be reshaped by puzzle generation concerns).

Is the idea of building diverse archives of high-quality content where solutions are distinguished by their emergent properties somehow foreign to ASP and EAs? Apparently not. Within the realm of EAs, "archive-based" algorithms (such as AMGA [3]) explicitly build and maintain such archives, returning the whole archive as their final result rather than just the best individual in the working population. For ASP, the "projected enumeration" feature in some solvers [4] allows users to explicitly request the generation of an example from every one of the possible classes of solutions, where classes are defined by some subset (projection) of the solution features. In the *Refraction* puzzle generator, projected enumeration was used to ensure each puzzle had a distinct laser graph (implying the puzzles differed by more than a trivial spatial adjustment).

References

1 Risi, S. and Lehman, J. and D'Ambrosio, D. B. and Stanley, K. O. *Automatically Categorizing Procedurally Generated Content for Collecting Games*. In: Proc. of the Workshop on Procedural Content Generation in games (PCG) at the 9th Intl. Conf. on the Foundations of Digital Games (FDG-2014). 2014.

2 Butler, E. and Andersen, E. and Smith, A. M. and Gulwani, S. and Popović, Z. *Automatic Game Progression Design through Analysis of Solution Features*. In: Proc. of the SIGCHI Conf. on Human Factors in Computing (CHI'2015). 2015.

3 Tiwari, S. and Koch, P. and Fadel, G. and Deb, K. *AMGA: An Archive-based Micro Genetic Algorithm for Multi-objective Optimization*. Cybernetics, IEEE Transactions on, volume 45 issue 1, pp. 40–52. Jan. 2015.

4 Gebser, M. and Kaufmann, B. and Schaub, T. *Solution Enumeration for Projected Boolean Search Problems*. In: Proc. of the 6th Int'l Conf. on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR'09), pp. 71–86. 2009.

3.8 Fun Resistance Bots

Pieter Spronck (Tilburg University, NL)

Joint work of Champandard, Alex; Cowling, Peter; Spronck, Pieter

During the 2012 Dagstuhl Seminar on Computational and Artificial Intelligence in Games, a workgroup focused on intelligence in modern board and card games. The work of this group culminated in a one-day programming jam session, in which several participants created intelligences for the game "The Resistance."

"The Resistance" (http://www.indieboardsandcards.com/resistance.php) is a game in which each player is either a Spy or a Resistance fighter. The roles are secret but the Spies know each other. One player is the leader (a role which rotates between the players

continuously), who has the responsibility to propose a team of a specific size to go on a mission. Everybody gets to vote on the team, and if the majority agrees with the leader's selection, the team gets to execute the mission. During the mission, the members of the team each play a card that indicates whether they support or sabotage the mission. These cards are played in secret, shuffled, and revealed. If one of the cards says that the mission is sabotaged, the mission fails. Otherwise it succeeds. As soon as three missions have succeeded, the Resistance fighters win. If, however, before that time the Spies manage to sabotage three missions, the Spies win. The Spies also win if five proposed teams are rejected in sequence.

The bots developed for "The Resistance" in 2012, and in the two years after that, have all been focused on playing the game as strongly as possible. This means that they try to make a solid estimate of which player is on which team, and of how opponent players play the game. The majority of the bots use expert-systems backed by statistical analyses. The setup of the competitions was aimed at allowing players to build opponent models if they wanted, with matches consisting of tens of thousands of games, and all communication between players happening via method calls. However, the original implementation of the framework encouraged bots to use a learning-centric based approach rather than a search-based approach. A game state that can easily be searched exhaustively was added before the meeting in Dagstuhl.

For the 2015 seminar, we aimed to investigate whether Resistance bots could be developed that are interesting or entertaining for human players to play against, rather than "just" effective. Effective bots were deemed to be a relatively straightforward problem to solve given time, but the Dagstuhl environment seemed better suited to building bots with more creative behaviors. To create such bots, three requirements must be met:

1. The bots must be able to play a reasonable game from the onset, and not need dozens of rounds of training, because when playing against humans there is no time for extensive training. Existing bots were refactored into easily extensible base classes for this purpose, e.g. Bounder and Invalidator.
2. The bots must be able to express their opinions on the game and on the other players, to allow for in-game discussion (which is an essential game element when humans play). This often required storing the information in a different format than when used purely for optimal play.
3. The bots must be able to communicate in human-understandable form.

The Resistance engine used for competitions has been expanded with the the ability for the bots to express messages on three levels:

1. Game moves: e.g., "I vote against the proposed team."
2. Structured opinions: e.g., "I estimate the probability that player 3 is a spy at 37.5%."
3. Free-form text: e.g., "That'll teach ya, ya scurvy dogs!"

Several bots have been implemented which make use of these features. The engine has also been connected to a chat system, so that mixed human/bot groups can play, using the chat-system features, and using messages for their communication. Also, a text-to-speech engine via a Web API and the built-in speech-to-text were added as features as well. With such a setup, we hope to investigate the following questions:

- Will the extra communication abilities change bot behavior at all? This is not immediately clear. If the bots do not get stronger by using the extra features, then increased playing strength provides no reason to use them, and the behavior of bots which aim only to maximise their win rate will remain unchanged. Ideally the artificial intelligence must be able to exploit the extra information to increase its win-rate. Experimentation will have to show whether or not that is possible,

- Will the extra communication abilities make the bots more fun to play against as a human? Naturally, as long as the bot is new, players might like occasional banter from the side of a bot, but if such banter is repetitive or not related to the game situation, it will not take long before it gets boring or even annoying. Experiments can be run to test out different bot implementations against humans in a webchat-based environment, and query the humans on their enjoyment of the bots.
- Can bot personalities help in creating more interesting bots? Once a good bot has been created with a strong communication system, it might be possible to create variants of it using a personalization system, which slightly colors behaviors and messages, for example to tailor the "fun" aspects of a bots behaviour to the moves and messages of the other players.
- Can bots be created that have human-reasonable opinions of whether the other players behaviour is "fun"?

The intention was to run an initial competition at the 2015 seminar, but different projects took precedence since this topic was already addressed at the previous Dagstuhl Seminar on AI/CI in games. However, some prototypes based on language-based interaction proved very successful, and further follow-up work has been scheduled for later in 2015.

3.9 AI Game Research

Kenneth O. Stanley (University of Central Florida – Orlando, US)

The website http://www.aigameresearch.org was conceived at the original Dagstuhl session on Artificial and Computational Intelligence in Games in 2012. Several researchers subsequently worked together to make the website a reality. The idea behind the site is to provide a focal point where video games that are based on artificial intelligence or computational intelligence research can be showcased to the general public. By providing such a venue, aigameresearch.org makes it possible for researchers to reach a broader audience with their games than would otherwise be possible. Furthermore, the more games the site showcases, the more interest it attracts. In that way, it amplifies the creative energy of the community by allowing all of us to benefit from each other's work. Each game submitted to the site is reviewed by an Editorial Board (http://www.aigameresearch.org/editorial-board/) in a manner similar to a journal. The current editor-in-chief is Kenneth Stanley of the University of Central Florida.

3.10 Algorithms That Learn To Play Like People

Julian Togelius (New York University, US)

Joint work of Paiva, Ana; Samothrakis, Spyridon; Schaul, Tom; Shaker, Noor; Sipper, Moshe; Togelius, Julian; Yannakakis, Georgios; Zambetta, Fabio

This workgroup was concerned with algorithms for learning to play games like humans. Learning to play a game like a human goes over and above learning to play a game in general, and has potential applications in game recommendations and personalization and content

evaluation for procedural content generation. It is also an interesting scientific question in its own right.

We assumed that different people have different game playing skills, just like the have different cognitive profiles in general. Some people have fast reactions, others are good at assessing large quantities of information, planning far ahead in the future, or maybe spatial reasoning. There is likely to be a player skill/performance profile that to some extent carries over between games.

With learning to pay like a human, one could mean several different things. We identified at least the following ways: mimicking the style of a particular player, mimicking the performance profile of a player, or mimicking the way a player learns. To illustrate the latter concept, an algorithm that learns to play Super Mario Bros while mimicking human learning style should learn skills in the same order: if the human learns to jump over gaps first and then learns to pick up power-ups, the algorithm should learn to learn to do this in the in the same order.

To further analyze the question, the various ways in which an algorithm could learn to play like a human could be described according to three axes: (1) Data which we are learning from: game profiles (e.g. from Steam accounts), human play-traces or interactively; (2) The resolution at which the model operates, from a single human to a player type to an average human; (3) the goal of the learning, i.e. predicting performance, predicting learning, mimicking learning, mimicking performance or mimicking style. The group agreed that predicting performance for classes of people based on interaction and human play traces is almost certainly doable with extensions of existing algorithm. Mimicking style (across games) is almost certainly doable. Mimicking learning is thought to be doable, but hard.

Finally, a proposal for a research project for predicting performance and predicting learning was sketched out. This included recording multiple humans' performance on multiple games, and using singular value decomposition to learn performance embeddings of humans. This would essentially identify a number of principal components of skill. It was proposed that the General Video Game Playing framework (based on the Video Game Description Language which was largely designed in the previous Dagstuhl Seminar) could be used for this.

3.11 AI-Based Game Design

Michael Treanor (American University – Washington, US)

Joint work of Cook, Michael; Eladhari, Mirjam; Levine, John; Magerko, Brian; Smith, Adam; Smith, Gillian; Thompson, Tommy; Togelius, Julian; Treanor, Michael; Zook, Alex

In our working group, we created a model for designing games around Artificial Intelligence (AI). AI-based games put AI in the foreground of the player experience rather than in supporting roles as is often the case in many commercial games. To develop the model we investigated the use of AI in a number of existing games and generalized our findings into several design patterns for how AI can be used in games. From there, we created a generative ideation technique where a design pattern is combined with an AI technique or capacity to result in an AI-based game. Next, the technique was put into practice in the creation of two AI-based game prototypes. From this work, a paper was created, submitted and is in review for the upcoming Foundations of Digital Games Conference (2015).

3.12 Neural Networks for Video Game AI

Mark Winands (Maastricht University, NL)

Joint work of Bauckhage, Christian; Bouzy, Bruno; Buro, Michael; Champandard, Alex; Cowling, Peter; Kendall, Graham; Lucas, Simon; Samothrakis, Spyridon; Schaul, Tom; Winands, Mark

Recently, deep convolutional neural networks (DNNs) have made huge advances in non-search based Go engines, playing a strong game without any lookahead, easily defeating GNU Go (a traditional search-based Go program). Though they are not defeating leading Monte Carlo Tree Search (MCTS)-based programs, they are performing respectably against them.

The goal of this work group was to further investigate how these Deep NNs can be applied for Game AI. In particular the work group was interested of its application to Real-Time Strategy (RTS) games. As a test case the RTS game Planet Wars was chosen, which was the Google AI Challenge of 2010. An advantage of this game is that its rules are straightforward, but the game possesses quite an amount of strategic depth. Also many bots are available to serve as a benchmark.

As an additional challenge is that the DNNs should be able to handle variable input sizes (e.g., variable planets) and to handle the game dynamics (e.g., generalize from different number of planets). Two solutions ideas for the architecture were discussed.

The first idea was a designed feature set. It contained concepts such as the number of ships, total growth, growth of the source / destination planet, etc. The core idea is a pairwise treatment of planets. An action consists of sending ships from a source planet to a destination planet. All possible actions are evaluated by the NN, and the one with highest score is chosen.

The second idea was a DNN that takes as input a session of frames. It would extend the architecture to image analysis. The game is encoded in terms of a stack of matrices so that convolutions would be applied.

For integration, optimal ways of combining deep learning with MCTS were discussed. As tree policy the DNNs could be applied only in promising nodes (i.e., which have been visited frequently). Another option is to use it for progressive widening (i.e., only to investigate interesting actions). For the rollout policy, the DNNs serve as action selection. There is though a tradeoff between the power of the standard DDN versus the speed of the pairwise treatment.

3.13 Procedural Content Generation and Formal Design

Alex Zook (Georgia Institute of Technology, US)

Joint work of Bidarra, Rafael; Cook, Michael; Liapis, Antonios; Smith, Gillian; Thompson, Tommy; Van Kreveld, Mark; Zook, Alex

With the consolidation of Procedural Content Generation as an academic area of study, there now exists both a need for greater theoretical foundation for design and simultaneously an opportunity to build these theories. We posit that all procedural content generators are themselves encoding a formal theory of game design, in terms of both the process being followed and the products that are being generated. Understanding formal design theories (including their construction, analysis, and evaluation) requires integration of research and

practice across several disciplines: the arts, humanities, design studies, and computer science. Game design and AI stand to benefit greatly from the varied perspectives on content generation by systematizing, generalizing, and deepening existing knowledge, while also broadening the range of topics addressed through procedural content generation.

Generators build upon design theories both explicitly and implicitly. Explicit models result from deliberate choices of a system designer to encode a form of the design knowledge for a domain (e.g., mazes) or a design process (e.g., the Mechanics, Dynamics, and Aesthetic framework). Implicit models are encoded within unacknowledged commitments expressed through algorithmic details (e.g., data structures, representation, generative methods) that may require critical analysis to be uncovered. Uncovering and acknowledging both the explicitly and implicitly encoded theories about the design process is key to learning from generators. Comparing them to each other and generalizing lessons learned from individual systems will lead to a broader, more inclusive, formal theory of game design. By examining the gap between generated products and exemplars made by human designers, it is possible to better understand the nature of the artifact being procedurally designed, thus building a formal theory of the artifacts being designed as well as the process taken to design them.

There is therefore value in understanding the design theory behind generators in terms of their goals, metaphors, methods, and ethics. Such a conscious commitment to an epistemological view on formal design theories in generators can lead to a better understanding of the generative process and the generated products. Formal approaches to design theories can support the analysis, interpretation and dissemination of PCG as an academic field and integrate practitioners including artists, designers, players, programmers, and researchers. For generative systems where the goal is primarily to produce a particular kind of playable experience, design theories allow artists to communicate their aesthetics, ethics, or message and reflect on it (in conjunction with the responses of a wider audience). For generative systems where the goal is primarily the system itself, design theories allow the creators of the system to fine-tune the algorithms as well as challenge current conventions.

In this working group we discussed motivations, high-level research topics, and initial research projects at the intersection of procedural content generation and formal design theory.

References

1 Shaker, N. and Togelius, J. and Nelson, M. J. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2015

2 Togelius, J. and Yannakakis, G. N. and Stanley, K. O. and Browne, C. *Search-based Procedural Content Generation: A Taxonomy and Survey*. IEEE Transactions on Computational Intelligence and AI in Games (TCIAIG), volume 3 issue 3, pp. 172-186, 2011.

3 Hunicke, R. and Leblanc, M. and Zubek, R. *MDA: A formal approach to game design and game research*. AAAI Press, 2004

4 Practical Sessions

4.1 The Dagstuhl Planet Wars Hackathon

Michael Buro (University of Alberta – Edmonton, CA)

Joint work of Bouzy, Bruno; Buro, Michael; Champandard, Alex; Cowling, Peter; Lucas, Simon; Samothrakis, Spyridon; Schaul, Tom; Spronck, Pieter

This abstract describes the Hackathon event that followed up on the Dagstuhl working group on neural networks for real-time game AI, in which neural networks for Planet Wars were developed. Planet Wars was the subject of the Google AI challenge in 2010 (http://planetwars.aichallenge.org/). It was chosen as the application domain for this Hackathon project for its simple rule set, existing programming frameworks, real-time decision complexity, and our curiosity about how state-of-the art AI techniques could be successfully applied to the game by AI experts who only have a day to implement a complete player. After introducing the Planet Wars game, we describe each approach in turn, what worked and what did not, experimental results, and why we as a group feel that this was one of the best workgroup experiences we ever had. We conclude the paper with a discussion of future research directions.

Planet Wars

Planet Wars is a two player zero-sum simultaneous move game in which players aim to destroy opponent's fleets in real-time. Figure 1 shows a screenshot. Circles represent planets with fixed positions and rates at which they generate ships. The numbers displayed inside planets indicate how many ships are currently stationed there. The planet's color indicates ownership (grey = neutral). Moves consists of picking a planet under your control and sending a number of ships to another planet. Ships cannot change course while in transit and do not interact with other ships (another version that was implemented during the workshop considered on-transit fleet collisions, but no AI systems were specifically designed for this variation). Once ships arrive at their target planets their count is either added to the planet's ship count (if friendly), or deducted from the count if the planet is neutral or hostile, in which case whoever has more ships left owns the planet. Neutral planets never send ships or create any ships. A player wins if all opponent's ships are destroyed. In the game version we used, planets and fleets are visible at all times.

The Planet Wars Google AI challenge was won by a bot written in LISP implementing alpha-beta search (http://quotenil.com/Planet-Wars-Post-Mortem.html). We first considered to using Google's server-client framework so that we would be able to compare our bots with the state-of-the-art, but then ended up using a simple Python implementation because we couldn't find Google's server code, and the Python project we found lent itself to quick development in the short time we had for the project. Alex Champandard set up a git repository (https://github.com/alexjc/planetwars) which contains all code that was generated during the Dagstuhl Seminar. The software is based on a Python project (https://github.com/maxbogue/planetwars) which provides game simulation code and a web server for game visualization.

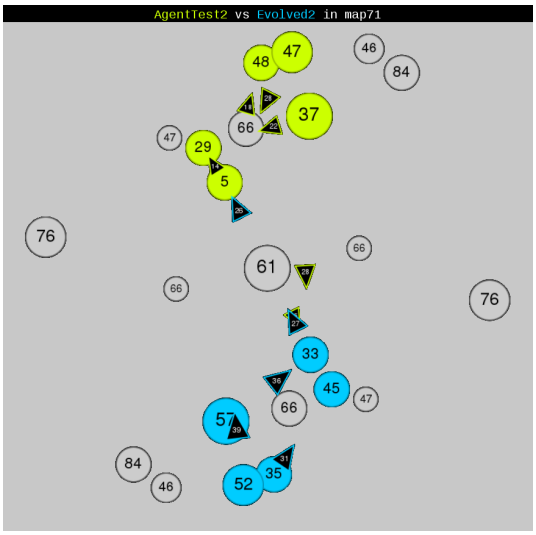


Figure 1 Planet Wars Screenshot.

Our Planet Wars Bots

In this section each group describes their approach for building a Planet Wars bot in 1.5 days using different AI techniques.

Evolutionary Algorithm (Lucas and Buro). We wanted to see how far we could get with a simple but interesting approach: the main idea was to use a simple evolutionary algorithm (EA) to learn a set of weights to play the game. We took the following steps:

1. Define how the AI would interface to the Planet Wars game
2. Define a feature set
3. Implement an initial set of weights for a linear classifier
4. Write a simple evolutionary algorithm in Python, for ease of interfacing to the game engine
5. Run the algorithm, both using random initial weights and hand-tuned initial weights
6. Evaluate the results

Steps 1 and 2 had been sketched out during an earlier session in the week.

For interfacing to the game (step 1), the approach was to consider a single possible action per time step, although the game allowed a list of actions to be executed each step. We defined an action as a planet pair, with the first element listing the source planet (which must be owned by this player) and the second planet being the destination planet, which could be any planet. In the event that the source was the same as the destination, this would have no effect and hence allowed for null moves. Otherwise, the action would be to transfer 50% of the ships on the source planet to the destination planet: a straight transfer in the case the planet was owned by this player, or an attack in the event that it was owned by the opponent.

Step 2: for each possible action, a feature vector was constructed based on the properties of each planet in the pair (for example, the number of ships, the size of the planet), and also on the relationship between them (e.g., the distance between them). Although the features had been sketched out in a previous session, during the hackathon it became clear that some

of them were ill-defined and needed some more work to implement in a meaningful way. Mapping the outlined features to a Python implementation was an interesting and satisfying part of the process.

Step 3: We designed an initial hand-tuned set of weights. There was an obvious intuition to some of them (e.g., give a high rating to choosing a source planet with lots of ships). Ones without an obvious way to set them were set to zero. We then tested this and found (very satisfyingly) that it beat all but the strongest of sample bots.

Step 4: Here's where we discovered just how rusty our Python skills were. Michael had left the EA team to try to improve feature weights manually together with Bruno Bouzy. So, Simon's initial approach was to take an existing Java EA code base and port it to Python: although EAs are quite simple to implement, to get a bug free version running in Python still took a couple of hours. This was tested on some toy problems, and then run to evolve feature weight vectors for the controller described above. With a population size of 10 running for just 100 generations from initial random weights (and even 50 generations) the evolved bot was able to defeat many of the sample bots. A limitation here was that fitness was evaluated by playing against a random bot: the problem is that after some initial evolution the evolved bot beats the random agent every time, and so there is no incentive for it to improve beyond this (there is no search gradient). Co-evolution was also implemented, which can lead to a more open-ended evolutionary process. Reasonable results were obtained with this, but there was insufficient time to see the full extent of what could be learned in this way.

Seeding the population with the hand-tuned weights led to a bot that was able to beat all of the sample bots, and were defeated only by the super hand-tuned bot (more about the actual tournament results later).

The only frustrating part of running the EA was the time taken for each fitness evaluation: each evaluation takes several seconds, and this severely limits what can be evolved on a single machine in a short period of time. This could be ameliorated by using a Java implementation of the game server.

One satisfying aspect of the simple evolutionary approach is that the idea of evolving a linear heuristic function had been roundly mocked earlier in the week as being too simple and therefore of no interest. Yet this turned out to be the leading approach of all the adaptive / learning methods tried during the hackathon. It reinforces the wisdom of starting with a simple method and then building on it.

There are many more experiments that can and should be made along these lines, and it will be interesting to see which approach wins out in the long run, and just how far simple methods can be taken.

"Professor Descent" (Bouzy and Buro). For the Hackathon day we chose to work on Planet Wars. We were considering to implement an MCTS bot together with Mark or Pieter. But given that we only had one day to produce a working player, we didn't know Python very well, and the anticipated playout speed obstacle for Planet Wars, we gave up on the MCTS idea. The previous day, Simon and Michael had written code for feature extraction and designed a "one-neuron bot" called AgentTest for a Python Planet Wars framework. Between lines 20 and 40 of this bot there were a dozen features evaluated and weighted by values that were easy to modify. So, we decided to play around with these values and having fun observing Planet Wars games with our manually tuned one-neuron bot playing against the existing bots all afternoon.

We created **AgentTest2**, a copy of the reference bot. Some features were sensitive to slight weight changes, and others were not. Manually changing the values of the sensitive features made AgentTest2 beat AgentTest most of the time rather quickly. That was great –

our novel manual method worked, and we were ready to give a name: "Professor Descent", a variation on the "Graduate Descent" method which professors use to solve hard research problems by assigning tedious work to a number of graduate students. However, at that point we discovered that while professor descent enabled AgentTest2 to soundly defeat the reference bot, it had lost its ability to beat an existing bot that AgentTest consistently defeated. Not worrying too much, we solved this problem by executing one more iteration of professor descent. Watching a few more games and further tweaking parameters, we came up with a weight set that was able to defeat all bots that shipped with the Python Planet Wars framework and the AgentTest reference bot.

Hackathon for us was the most enjoyable part of this year's Dagstuhl Seminar. We learned how to get root access without knowing the root password on Bruno's university Linux notebook to install Python, we refined our Python programming and Git skills (which were rather rudimentary when we started to be honest – thanks Alex!), enjoyed watching Planet War games – leaning back, sipping coffee, and cheering for AgentTest2, while watching other teams frantically typing, trying to improve their bots. The professors were rather pleased with the results of their "descent" method. Little did they know that another strong competitor was in the making ...

Machine Learning (Champandard, Samothrakis, Schaul). Our goal was to implement a system that learns approximate state-action values using variations of Q-Learning and Monte Carlo methods in the game of Planet Wars. State-action values are approximated using ReLU-type neural networks. The learning process involves fixing opponents and maps and training for a predetermined set of episodes. The process worked reasonably well, but requires extensive hyper-parameter tuning on the following fronts: (a) neural network weight initialisation, (b) neural network learning rates, (c) exploration rates, (d) network topology, (e) input and output scaling, (f) exploration rate decay, (g) discount rates. The number of hyper-parameters that need tuning makes standard neuro-control methods hard to apply out of the box in games and we think this is a vital shortcoming for the widespread adoption of these methods. Possible ways ahead include, but are not limited to, using already existing knowledge to guide exploration (similar to default policy playouts in Monte Carlo Tree Search) and incorporating learning rate adaptation techniques from supervised learning.

Rule-Based AI (Spronck). The **Hotshot** Planet Wars bot takes a parameterized rule-base approach. At every turn, for each planet, it will decide whether or not to launch ships. It has a "max-in-flight" parameter that determines the maximum percentage of the total ships owned that is allowed in flight. This way, it intends to ensure that there are always enough defenses. Moreover, there is a minimum percentage that must always be left behind on a planet, and if the number of ships on the planet is less than that, it will not send ships from the planet.

If it decides to launch ships, it may send them to neutral and/or enemy planets. Its potential targets are the closest, lowest-growth, highest-growth, and "easiest" of the target planets. In this case, "easiest" means easiest to conquer, i.e., the least defended planet (if there are multiple choices, it will take the closest). To determine how many ships are to be sent to each of the chosen targets, parameters are used.

Once the basic rule-base was in place, default values for all the parameters were chosen. Then the Hotshot bot was pitted against several of the default bots, in particular the stronger ones, such as "all to close or weak". It fought on all 100 maps. A hill-climbing mechanism was used to change the parameter values after each 100 skirmishes, in order to increase the number of victories. As soon as 100% victories was reached, another opponent was chosen.

This way, Hotshot tweaked its parameters to get to a stage that it would win against any of the default bots (nearly) 100% of the time. At that point, the parameters were frozen.

Using the parameters learned in this way, Hotshot is a very fast bot, that indeed defeats all the "beginner" bots nearly 100% of the time. Unfortunately, it does not do so well against more advanced bots. When examining the parameters values that it learned, we see some of the reasons why. The parameters show that Hotshot heavily favors attacking enemy planets with low growth, followed by the closest enemy planet, and the easiest-to-conquer neutral planet. This does not seem to be a strong tactic, as low-growth planets, once conquered, do not contribute much to an empire, while leaving high-growth planets in enemy hands gives the enemy an advantage. There is no denying that the strategy works against the beginner bots, but it is no surprise that it is pretty bad against more advanced bots.

It is not hard to implement more "rules" in the bot, and let it use hill-climbing to learn more parameters. The set of rules it uses now are probably insufficient to defeat more advanced bots in any case, regardless of parameter learning. The basic approach is sound, though, and leads to fast bots.

MCTS (Cowling). The **MCTS2** bot uses UCT to make decisions at each iteration. Doing this at 60 decisions per second poses some difficult challenges, which we addressed to some extent via the time-slicing approach of our successful PTSP player (where we use a time slice consisting a many frames and only make a decision a few times per second). However, in the hackathon there was too little development time for this – the main pressure there was on producing something that worked, made sensible decisions, and embodied an idea that could be enhanced later.

The MCTS2 bot builds on Michael Buro's useful Planet Wars class to capture state and issue commands. It uses the Python implementation of UCT which Ed Powley, Daniel Whitehouse and I wrote a couple of years ago – which can be found at http://mcts.ai/code/python.html. Hence the functions that are needed for a basic implementation are simply Clone(), DoMove(), GetMoves() and GetResult(). Here GetResult() returns the player that has won from a terminal state (and otherwise throws an exception). The Clone() function proved tricky – it had to create a changeable version of the world state, then once a move was chosen, it had to refer back to the original world state. Early versions did not really work – essentially the simulations based on random moves would take ages to terminate. The trick that made it all work was to consider only moves where exactly half of the ships on a planet were sent to a planet where they would be able to successfully conquer (based on the ships currently on the planet), and allow only one move per iteration. This way of pruning moves both allowed simulations to terminate in a reasonable number of moves, and actually provided stronger play than all other players available at the time of testing even with a single iteration. With 1000 iterations the player seemed very strong, but a single move took several seconds. In order to run at competition speed 5 iterations per move were chosen (and for very small numbers of iterations play was better with n+1 iterations than with n. 5 iterations was about right – and that was the competition bot. I wasn't there for the final playoffs, but MCTS2 did seem very strong, and stable, in testing. This in spite of Alex twice adding an "MCTS is not an algorithm" exception ☺

The time pressure of the hackathon, and working together under pressure in the same room as a group of like-minded colleagues, made the hackathon one of the highlights of the week.

Table 1 Tournament 1 results after 25 round-robin rounds. At this point MCTS2 choked and had to be stopped after spending 15 minutes on one game:

	Player	total%	0	1	2	3	t-avg (ms)	t-max (ms)
0	MCTS2(5)	57	–	64	36	72	93.2	347.4
1	AgentTest2	55	36	–	56	72	5.0	18.9
2	Hotshot	52	64	44	–	48	0.1	1.4
3	Evolved	36	28	28	52	–	3.5	13.7

Table 2 Tournament 2 results after 100 round-robin rounds (MCTS2 excluded):

	Player	total%	0	1	2	t-avg (ms)	t-max (ms)
0	AgentTest2	65	–	68	62	6.1	56.4
1	Evolved	48	32	–	64	4.2	15.4
2	Hotshot	37	38	36	–	0.1	1.1

Table 3 Tournament 3 results (100 rounds) with Evolved2:

	Player	total%	0	1	2	t-avg (ms)	t-max (ms)
0	Evolved2	83	–	84	83	4.8	15.4
1	AgentTest2	39	16	–	61	4.9	21.7
2	Hotshot	28	17	39	–	0.1	1.1

Table 4 Tournament 4 results with Evolved2 and MCTS(1) after 50 rounds, when MCTS(1) encountered a list index out of range error:

	Player	total%	0	1	2	3	t-avg (ms)	t-max (ms)
0	Evolved2	76	–	61	84	82	4.4	15.2
1	MCTS2(1)	53	39	–	45	76	19.0	149.2
2	Hotshot	38	16	55	–	43	0.1	1.1
3	AgentTest2	33	18	25	57	–	4.5	21.1

Tournament Results

By Thursday night, four bots were ready for the final competition. Later, two more entries were added to study the effects of parameter changes:

- MCTS2(5) uses move pruning and five playouts per move
- AgentTest2 is based on the "one-neuron" linear move evaluator with hand-tuned weights,
- Evolved uses the same feature set, but evolved weights starting with random values and training against built-in bots.
- Hotshot is a fast rule-based bot.
- MCTS2(1) uses one playout per move
- Evolved2 is identical to Evolved, except for the weights which were evolved by starting with the hand-tuned weight set.

Tables 1–3 show the results of the four bots playing round-robin tournaments using the 100 symmetrical maps that come with the Python Planet Wars framework. The time columns indicate that the submitted MCTS player used considerably more time than the other bots. In fact, the first tournament had to be suspended for lack of progress for 15 minutes in a single game caused by MCTS2(5). For this reason, we removed MCTS2(5) and reran the tournament. Among the remaining bots, AgentTest2 ended up winning the main contest, followed by Evolved and Hotshot. We then entered Evolved2 – the evolved player which

started from manually tuned weights, to see how good "professor descent" really is. Not that good, as it turned out, looking at Table 3. Lastly, we re-entered MCTS2, now using just one playout per move, to see how well it plays in a stricter real-time setting. Unfortunately, halfway into the tournament it crashed. At that point Evolve2 which is more than 4 times faster prevailed again.

Conclusion

In this abstract we reported on an exciting Hackathon Dagstuhl event whose goal it was to find out which state-of-the-art AI techniques work best to create AI systems for simple real-time games within 1.5 days.

Sparked by recent advances in deep learning we were optimistic that we could create a strong player based on reinforcement learning. Unfortunately, this effort failed, due to slow game generation. We also witnessed EA running into issues when seeded with random weights. The weight set obtained by starting with our hand-tuned weights performed much better, in fact, winning the final tournament.

There are a lot of venues that can be explored from here. For instance, it would be interesting to see how well MCTS performs when using playout policies based on Evolved2. Comparing our bots with the ones that performed well in the original Planet Wars competition would shed light into how MCTS and learning bots fare against classical minimax based players. Finally, exploring the trade-off between acting fast and acting well in real-time games is worth studying to help us create strong AI systems capable of defeating human expert players.

4.2 What The Hell Is Going On?

Simon Colton (University of London/Goldsmiths, GB)

In the MetaMakers Institute at Falmouth University, we are developing an automated game generator within a Computational Creativity context. On the technical side, this is the first system we have developed from scratch which is intended to automatically alter its own code, in order to increase the yield, variety, and unexpectedness of games, and address some of the higher level issues of autonomy and intentionality in software. On the gaming side, we aim to build a community of players who enjoy and share the games, with the added benefit of sometimes being the first person in the world to solve a game level. On the philosophical side, we are interested in the correspondence between an open conjecture in pure mathematics and an unsolved game level, especially in balancing the joy of solving a previously un-solved game with the futility of attempting to solve a game that might not be solvable, and the strategies of knowing when to give up that this leads to.

At the start of the Dagstuhl Seminar, the first prototype of the game generation system was demonstrated individually to people, and feedback was gratefully received and processed. The system produces puzzle mini-games, where a strategy for placing pieces on board cells has to be devised by the player and then used to solve the game. As both the pieces and the board move around, there is a physical element to each game which increases difficulty. Currently, games can be randomly generated and hand-finished, and ten such games were demonstrated, with the intent of showing that the game space contains difficult puzzles.

Many seminar attendees played the games and found them in part enjoyable and in part frustrating, with some people attempting more than 100 times to solve particular games over a substantial period spanning several glasses of wine. Lessons were learned about player interaction, making the games more like toys to increase the fun, and how people feel about playing a game that might be unsolvable.

During the sessions on whole game generation and through the interaction with seminar attendees, the approach was improved so that the next stage of development could take place during the game-jam session of the seminar. Here, the aim was to show that, by including images in the games, the space includes games which have contextual meaning. To this end, five games with a Dagstuhl theme were produced in a semi-automated way and showcased at the end of the seminar. The games were as follows: "Let it snow" (where the best tactic is just to the let the snow fall); "Too many Michaels, Too many ideas" (where the aim is to enable a researcher to concentrate on one topic at a time); "Fabulous Dan Ashlock" (where the aim is to get to the coffee without being sidetracked by interesting conversations); "Random Lunchtime Shuffle" (where the player must seat all the seminar attendees on the moving tables); and "Tired and Emotional" (where the aim is to herd the researchers to their bed, even though they want to be elsewhere). In the context of the Dagstuhl Seminar, these games had much meaning, and were well received by the seminar attendees.

4.3 Twitter Bot Tutorial

Michael Cook (University of London/Goldsmiths, GB)

In this session participants were invited to set up their workstations for the development of Twitter bots, small programs that run at regular intervals usually on external web servers. Bots normally produce some kind of content when they run, which they post to Twitter either as a regular tweet on the service (a message of 140 characters, possibly including image URLs or other media) or an interaction with another user (a semi-private message directed at one of Twitter's users). Twitter bots are an emerging medium for art and experimentation, and are also being used to design and augment games and related technology. The session was organised to enable more people to engage with the medium, as it has a lot of potential for researchers.

Many games interact directly with Twitter by allowing their users to post information to it such as high scores, and some games such as Hashtag Dungeon use Twitter data to generate content within the game dynamically. In addition to design, Twitter is also a useful platform for experimentation: researchers are already using Twitter as a way of conducting surveys, gaining user feedback, or writing lightweight applications that allow users to submit information (such as location data) with technology they already have on their phones. Because Twitter is restricted to short messages, it's an ideal medium for communicating with simple apps, and Twitter bots are an excellent way to build software that facilitates this.

All participants successfully set up bots, and some used it to connect other code they had written at Dagstuhl to bots which posted to Twitter. We hope to see the fruits of this work in research projects in the future.

4.4 Sonification of Character Reasoning

Ian Horswill (Northwestern University – Evanston, US)

As with any complex program, character AI systems perform long chains of operations through time. And like other programs, understanding the operations of these systems can be difficult. This is an issue not only for debugging but for making the behavioral state of the system transparent to the player.

Take the case of debugging. The real-time nature of games limits the usefulness of typical debugging tools such as breakpointing and single-stepping, so game programmers often resort to exhaustive logging. However, log files can be difficult to work with. A human can look at only a small portion of the log at a time; even with filtering tools, understanding the log can still be a laborious process. Visualization techniques can help make patterns and anomalies visually apparent without requiring the programmer to fully digest the text being represented. But even they require the programmer to interrupt the game, or at least gaze away from it, so as to attend to the visualization.

An interesting alternative is the use of *s*onification: the rendering of logs as sound in time, rather than as text or image in space. The basic approach is simple. As with other techniques, one augments the code to "log" events as they happen. However, rather than logging the events as text in a file, one logs them as sound events in the audio stream generated by the game. This allows the "log" to be produced continually as the game runs, giving feedback to the programmer without having to interrupt execution.

As part of the seminar's hackathon, I added sonification to the experimental AI-based game *MKULTRA*. The sonification used granular sound synthesis [2] where each logged event generated a 1–10 ms "grain" of sound whose spectral characteristics depend on the event being logged. Consideration of a problem-solving operator produces one type of grain; resolving conflicts between competing operators, another; exception handling, another; and consideration of lying to another character, produces yet another type of grain. All grains were synthesized using a simple FM sound synthesizer [1] whose parameters (carrier frequency, modulation frequency, and modulation level) were adjusted based on the type of event being logged.

Although not a replacement for proper text logs, the sonification system can make apparent patterns in temporal behavior such as looping. It can also allow the programmer to notice anomalies such as differences in sound from one run to another, allowing a kind of passive debugging.

More interestingly, however, sonification also introduces the potential for novel gameplay mechanics where players rather than programmers are required to detect anomalies the sonification. In the fiction of *MKULTRA*, the sonification of NPCs is explained in terms of the player character having limited mind-reading capabilities. The player can gradually learn to recognize lying and other anomalous behavior on the part of an NPC by learning to distinguish the sound textures associated with different cognitive processes in the NPC.

References

1 Roads, Curtis (2001). Microsound. Cambridge: MIT Press. ISBN 0-262-18215-7.
2 Chowning, J. (1973). "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation". Journal of the Audio Engineering Society 21 (7).

4.5 MCTS for PCG

Adam M. Smith (University of Washington – Seattle, US)

Monte-Carlo Tree Search (MCTS) and procedural content generation (PCG) were independently popular topics a the seminar, however little discussion involved combining the two. MCTS is usually seen specifically as a way define a game-playing agent's behavior while the choices made inside a PCG system, often a result of search, are not seen as manifesting the behavior of any specific agent. These default perspectives need not block us from imagining new integrations, however.

I presented a prototype system, partially developed during the seminar's hack day, that applied MCTS to a PCG problem. In my prototype, the tree-search algorithm explored alternative executions of an imperatively-defined nondeterministic procedure, receiving rewards according to how long the procedure could execute before encountering an exception.

The example domain, generating simple solvable algebraic equations, was derived from an earlier generator I created for *DragonBox Adaptive* (http://centerforgamescience.org/portfolio/dragonbox/) In the domain-specific parts of the generator, I wrote imperative Python code which used an abstraction with a similar interface to a random number generator to make nondeterministic choices, e.g. defining equation structure and selecting coefficient values. During search, whenever a nondeterministic choice was encountered, the `fork` syscall was used to branch executions in which a different value was returned. The tree-search algorithm directed exploration of these branches much more effectively than the random or bread-first selection policies I tried.

Although the project was prompted by the potential to combine MCTS and PCG, it also inadvertently explored another PCG-related discussion point: must constraint-based PCG systems necessarily be implemented in declarative languages? My demonstration showed one way for the constraints on a design space to be derived automatically from the execution of an imperative program (similar to the use of path-exploring symbolic execution in formal verification [1]) as well as suggesting how MCTS could be applied as a lightweight constraint solving algorithm (employing a fixed variable ordering). The use of MCTS for solving constraints [2] and the use of operating system call to build search spaces from the execution of native code [3] are not new, however their introduction to PCG researchers is a significant novelty.

References

1 Williams, N. and Marre, B. and Mouy, P. and Roger, M. *PathCrawler: Automatic Generation of Path Tests by Combining Static and Dynamic Analysis*. In Proc. 5th European Dependable Computing Conference (EDCC-5), Budapest, Hungary, April 20-22, 2005, Lecture Notes in Computer Science 3463 Springer 2005, ISBN 3-540-25723-3, Budapest, Hungary, April 2005, pages 281–292.

2 Loth, M. and Sebag, M. and Hamadi, Y. and Schulte, C. and Schoenauer, M. *Bandit-based Search for Constraint Programming*. In Proc. of the AAAI Workshop on COmbining COnstraint solving with MIning and Learning (COCOMILE), 2013.

3 Paige, B. and Wood, F. *A Compilation Target for Probabilistic Programming Languages*. In Proc. of the 31st Int'l Conf. on Machine Learning, pp. 1935–1943, 2014.

4.6 Exploring Embedded Design Theory in Maze Generation

Gillian Smith (Northeastern University – Boston, US)

As an offshoot of work discussed by the PCG and Formal Design Methods group, one concrete project is to examine the formal design theories that are embedded in a single, simple domain: both in terms of the process being followed by the generator and theories of the product being created. A domain for procedural generation that has seen a great deal of prior work is that of maze generation; some of the earliest generative systems created simple mazes and labyrinths. The aim with this project is to examine maze generators from a wide variety of authors: computer scientists and artists, students and professional developers, researchers and hobbyists. What aspects of mazes are prioritized in the representation, what is implicit in the specification, and what is left to fall out in emergent behavior? What kinds of approaches to the design process are modeled? What kind of design properties are considered required or desirable, and how are those properties enforced?

Several Dagstuhl participants created or documented their own maze generators to bootstrap collecting examples for this study. Approaches include generation via tree search, optimization (e.g. genetic algorithms), constraint satisfaction (e.g. answer set programming), and ad-hoc methods. All mazes thus far assume an underlying grid structure. The intent moving forward is to cast a wide net to find other existing maze generators, solicit for additional maze generators from a diverse audience, and study ways to evaluate the fitness of each maze and expressiveness of each generator for a given purpose.

4.7 Representation Learning for Procedural Content Generation

Kenneth O. Stanley (University of Central Florida – Orlando, US)

An important goal for procedural content generation (PCG) is to be driven by data [1, 2, 3, 4]. That is, in such data-driven PCG, the computer could be shown examples of the kind of content it should generate, and it would then on its own generate an entire content space automatically. While this capability does not presently exist, a recent paper [5] hints that deep neural networks may someday provide it. In the paper, *Learning to Generate Chairs with Convolutional Neural Networks*, a deep neural network is shown 50,158 chair images and learns from those to generate new chairs that were never seen before. Interestingly, this process is the reverse of what most deep networks do: instead of training on images and outputting classifications, in effect this network trains on classifications and outputs images. More broadly, generative models learned by neural networks suggest similar potential.

While the full potential of such systems remains to be explored, their realization would open up many promising applications in PCG. For example, sprite sheets and 3D models could be generated en masse for enemies, friends, characters, monsters, animals, objects, weapons, vehicles, dwellings, trees, etc. The technique could also potentially be extended to sound, for example to generate a space of audio explosions. Designers might also benefit by using such systems as a creative tool for inspiration. The capability to generate an image from a set of descriptive parameters also suggests more ambitious potential applications,

such as inputting an image and outputting an animated behavior for that image. Examples include walking gaits for body morphologies, explosions or appropriate physical properties for particular objects, growth processes for plants, or even skin for skeletons. Perhaps even a descriptive vector for a level could be input and yield an entire output level.

Beyond just applications, data-driven PCG raises several intriguing theoretical issues and questions. For example, is it possible to learn to extrapolate beyond the bound of given examples, such as to generate a level more difficult (yet still engaging) than anything seen before? More generally, how effective are such techniques with only limited data? Is it possible to ensure the feasibility of generated content, or is it necessary to prune suggestions away after training is over? One particular challenge might be to obtain different options for the same interpolation, given that a neural network often only outputs one result for any given input.

While the paper on generating chairs provides one possible approach to such data-driven PCG (a deconvolutional network), others are conceivable. For example, generative models like autoencoders or RBMs might provide similar capabilities. Evolutionary diversity methods such as novelty search or MAP-Elites might also provide similar capabilities, in addition to the potential to discover multiple diverse instances of the same interpolation. Transfer learning could also apply, that is, one domain might inform another. For all techniques, obtaining the necessary training data will of course always be an important prerequisite.

References

1 Steve Dahlskog, Julian Togelius, and Mark J Nelson. Linear levels through n-grams. Proceedings of the MindTrek Conference (2014).
2 Sam Snodgrass, and Santiago Ontañón. Experiments in map generation using Markov chains. Proceedings of Foundation of Digital Games (2014).
3 William Raffe, Fabio Zambetta, Xiaodong Li, Kenneth O. Stanley. An Integrated Approach to Personalized Procedural Map Generation using Evolutionary Algorithms. IEEE Transactions on Computational Intelligence and AI in Games (2014).
4 Noor Shaker and Mohamed Abou-Zliekha. Alone We can do so Little, Together We can do so Much: A Combinatorial Approach for Generating Game Content. Proceedings of Artificial Intelligence and Interactive Digital Entertainment (2014).
5 Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to Generate Chairs with Convolutional Neural Networks. arXiv preprint arXiv:1411.5928 (2014).

4.8 What Did You Do?

Julian Togelius (New York University, US)

Joint work of Cook, Michael; Eladhari, Mirjam; Smith, Gillian; Thompson, Tommy; Togelius, Julian; Zook, Alexander

What did you do? is a prototype game designed and mostly implemented during the course of the seminar. The design originates in the discussions of the "AI-based game design" group, where a number of design patterns used to make foreground AI useful in games were identified. In order to validate and exemplify these design patterns, two small game prototypes were hastily designed and very hastily implemented. One if them is "What did you do?". The game is designed to implement three of the identified patters: Player Trains AI, Player Edits AI and Visualize AI State.

The game is turn-based and plays out on a grid. There are five types of entities in the world: the player character (parent shrub), childshrubs, stones, strawberries, and ponds. The parent shrub can move in any of the four cardinal directions, eat or pick up whatever is in the direction it is facing, or drop something it is carrying. The parent has an energy level which can be recharged by eating. Child shrubs have mostly the same actions available, but they have somewhat different effects: in particular, if the child picks up stone it might be crushed under its weight, and it will drown if moving into a pool. The child's energy levels are also constantly decreasing. Both stones and ponds block the way, but stones can be picked up and moved elsewhere. In the course of the game, rivers can be forded by stones, strawberries can be eaten etc.

In a nutshell, the game is about feeding and protecting your kids while also protecting them from the outcomes of the stupid things they while imitating your behavior. A rule learning algorithm runs in the background, and finds common situations the parent faces and what action the parent took. The children's brain (they share a brain) learn rules from this, and these rules are clearly displayed to the player in the GUI. Using resources gotten from eating strawberries, the player can remove certain rules, but this means that the same resources are not available for feeding the kids. There is thus at least two different tradeoffs required during gameplay: between feeding the kids and weeding the thoughts (rules) in their mind, and between defending the kids against the outside world and defending them against themselves. In both cases, this requires engaging with and understanding the workings of the AI system. Thus, we believe that this prototype shows how to combine several of these design patterns into a truly AI-based game.

4.9 The Grey Eminence: A Political Game of Conflict, Influence and the Balance of Power

Fabio Zambetta (RMIT University – Melbourne, AU)

Joint work of Lanzi, Pier Luca; Zambetta, Fabio

We propose an AI-based game designed around the Years of Lead [1], a period of political turmoil in Italy that stretched across the 1960s up to the 1980s. The period was characterised by the heavy influence of secret services of the two blocks in the Cold War in key events of the political life, as well as their interference in now infamous events, such as the bombing at the Bologna Station in 1980 or the Piazza Fontana bombing in 1969.

The player is indeed tasked with trying to implement the Strategy of Tension [1], what used to be the strategic centerpiece of Gladio [2] a so-called stay-behind organisation that was supposed to be reacting to a Soviet invasion in Europe but indeed started to proactively plan false-flag operations. The aim of the game is to spread terror and anxiety in the Public Opinion such that the Prime Minister is pressured into declaring the State of Emergency.

From a technical perspective, we endeavour to represent key actors in the game (the Prime Minister, the Public Opinion, the Leader of the Opposition, etc.) as agents that can be swung via specific operations, whose cost, paid out of a budget provided by Gladio, as well as their risk/reward trade-off can vary considerably. The relationships between such agents will be non-linear so that the player will not be able to easily predict patterns of behaviour that can elicit victory.

Another interesting technical aspect of the game is that feedback provided to the player at the end of a turn, will be in the form of TV news, newspaper articles, etc. detailing the current situation in the parliament as well as other "random" events (e.g., a new conflict in Middle East has started, a very big IT company has gone bust, the global economy is experiencing a recession). Such news will be procedurally generated out of templates and/or news stubs, which will then be customised on the fly via tokens, tags and annotations that will be instantiated based on the current context and gameplay progression.

References

1 Ganser, D. NATO's Secret Armies: Operation Gladio and Terrorism in Western Europe. London (2005).
2 Ganser, D. Terrorism in Western Europe: An Approach to NATO's Secret Stay-Behind Armies. ISN. Whitehead Journal of Diplomacy and International Relations, 6(1), South Orange NJ (2005).

4.10 One Word at a Time: a Magnetic Paper Bot

Jichen Zhu (Drexel University – Philadelphia, US)

Joint work of Zhu, Jichen; Ontañón, Santiago; Magerko, Brian

One Word at a Time (OWAAT) is a program based on an improvisation theater game of the same name. In the original game, two or more people compose a story by taking turns to add one word at a time. Sometimes used as an ice-breaker, an important aspect of the game is the unexpected direction toward which the sentence might be steered, often achieving humorous effects.

The goal of this project was to create bots that could play OWAAT to generate academic writings, while preserving the unexpected and humorous tone in the original game. In addition, we wanted each bot to exhibit certain linguistic characteristics of specific authors of choice, in the visual style of magnetic poetry.

To address this challenge, we designed and developed a bot that can generate sentences, one word at a time, based on a corpus of writings by a given author. In this way, the bot can play the game with a human, or it can play with other bots trained on a different corpus. To make the sentences as grammatically and semantically coherent as possible, we used the following techniques.

Given a corpus of academic papers written by a given author, the OWAAT bot plays the game thanks to two main components:

1. A low-level statistical model: this model captures the frequencies of the different words in the corpus relative to the context in which they appear. For our prototype, we used a first order Markov chain, which captures the probability distribution of a word given the previous word.
2. A sentence-structure model: this model tries to ensure that the generated sentences are syntactically correct (i.e., they have a subject, a verb and a predicate). In our prototype, we used a finite-state machine (FSM) where transitions between states correspond to "part-of- speech" tags ("verb", "noun", "preposition", "determinant", etc.), The FSM is designed in a way that traversals from the start state to any of the terminal states correspond to valid sentences, and was generated by hand.

For the bot to add the next word to the current partial sentence S, it first determines the state in the FSM that S is in. The transitions coming out of this state are the possible part-of-speech tags that the next word can have. The Markov chain is used then to generate the next word, only considering the words that have the appropriate part-of-speech tag. In this way, by combining both low-level and sentence-structure models, our bot can play the game, resulting in sentences that 1) resemble the sentences written by the intended author, 2) have local consistency (thanks to the Markov chain), and 3) have whole-sentence consistency (thanks to the FSM).

To demonstrate our bot, we gathered publications from several researchers at this Dagstuhl Seminar and trained bots to play OWAAT in the style of these authors. Underlining the comic effect, we designed the interface to visually resemble magnetic poetry, which shares the similar aesthetics of unexpected juxtapositions.

4.11 Contrabot

Alex Zook (Georgia Institute of Technology, US)

Joint work of Champandard, Alex; Cook, Michael; Smith, Adam; Smith, Gillian; Thompson, Tommy; Zook, Alex

Contrabot (Figure 2) is a game prototype for an AI-based game, developed at the 2015 Dagstuhl Seminar on Computation and Artificial Intelligence in Games. AI-based games foreground interaction with an AI system in the player's experience. Our goal with Contrabot was to build a game based on agents that learn, crafting gameplay around understanding, playing against and ultimately deceiving a machine learning system.

You play as a smuggler trying to secretly label boxes to communicate with a contact on the other side of a customs checkpoint. The smuggler is trying to learn the code you use to indicate a box is contraband – but an inspector is randomly checking boxes too. Can you design and redesign your secret codes to stop the inspector learning your patterns? Will you still manage to sneak your code through to your contact and smuggle your goods out of the country?

The game mechanics revolve around how the smuggler and inspector agents learn to check codes based on codes they have seen. These agents have two main processes: learning

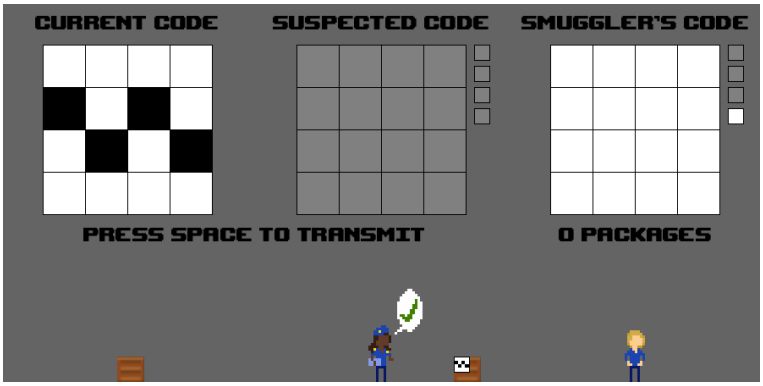


Figure 2 Contrabot.

codes and matching new codes against their learned code. Both agents generalize patterns from example codes – using a form of least general generalization – in their memory to then try to match new codes to these learned patterns. The inspector has a larger memory than the smuggler and gameplay is based on using reverse-engineering how learning works to take advantage of the smuggler forgetting old patterns more quickly than the inspector. The generalization process is simple, when comparing all codes seen the agents memorize exact matches for black or white tiles and generalizing to gray tiles if a position has been occupied by both colors. Despite this simplicity the design accommodates many levels of difficulty and risk-reward considerations for players based on the size of the codes used and memory capacities of each agent.

The game prototype is available to play and fork the design at https://github.com/gamesbyangelina/contrabot.

Participants

- Dan Ashlock
University of Guelph, CA
- Christian Bauckhage
Fraunhofer IAIS – St. Augustin, DE
- Rafael Bidarra
TU Delft, NL
- Bruno Bouzy
Paris Descartes Unversity, FR
- Michael Buro
University of Alberta, CA
- Alex J. Champandard
AiGameDev.com KG – Wien, AT
- Simon Colton
University of London/Goldsmiths, GB
- Michael Cook
University of London/Goldsmiths, GB
- Peter I. Cowling
University of York, GB
- Mirjam P. Eladhari
University of Malta, MT
- Ian Horswill
Northwestern University – Evanston, US
- Graham Kendall
University of Nottingham, GB
- Pier Luca Lanzi
Politecnico di Milano Univ., IT
- John M. Levine
University of Strathclyde, GB
- Antonios Liapis
IT Univ. of Copenhagen, DK
- Simon M. Lucas
University of Essex, GB
- Brian Magerko
Georgia Inst. of Technology, US
- Michael Mateas
University of California – Santa Cruz, US
- Joshua Allen McCoy
University of California – Santa Cruz, US
- Mark J. Nelson
IT Univ. of Copenhagen, DK
- Ana Paiva
INESC-ID – Porto Salvo, PT
- Mike Preuss
Universität Münster, DE
- Günter Rudolph
TU Dortmund, DE
- Spyridon Samothrakis
University of Essex, GB
- Tom Schaul
Google DeepMind – London, GB
- Noor Shaker
IT Univ. of Copenhagen, DK
- Moshe Sipper
Ben Gurion University – Beer Sheva, IL
- Adam M. Smith
University of Washington – Seattle, US
- Gillian Smith
Northeastern University – Boston, US
- Pieter Spronck
Tilburg University, NL
- Kenneth O. Stanley
University of Central Florida – Orlando, US
- Ruck Thawonmas
Ritsumeikan Univ. – Shiga, JP
- Tommy Thompson
The University of Derby, GB
- Julian Togelius
New York University, US
- Michael Treanor
American University – Washington, US
- Marc van Kreveld
Utrecht University, NL
- Santiago Ontanon Villar
Drexel Univ. – Philadelphia, US
- Mark Winands
Maastricht University, NL
- Georgios N. Yannakakis
University of Malta, MT
- R. Michael Young
North Carolina State Univ., US
- Fabio Zambetta
RMIT Univ. – Melbourne, AU
- Jichen Zhu
Drexel Univ. – Philadelphia, US
- Alex Zook
Georgia Inst. of Technology, US

Report from Dagstuhl Seminar 15052

Empirical Evaluation for Graph Drawing

Edited by

Ulrik Brandes[1], Irene Finocchi[2], Martin Nöllenburg[3], and Aaron Quigley[4]

1 **University of Konstanz,** `ulrik.brandes@uni-konstanz.de`
2 **University of Rome "La Sapienza",** `finocchi@di.uniroma1.it`
3 **KIT – Karlsruhe Institute of Technology,** `noellenburg@kit.edu`
4 **University of St. Andrews,** `aquigley@st-andrews.ac.uk`

Abstract

This report documents the program and outcomes of Dagstuhl Seminar 15052 "Empirical Evaluation for Graph Drawing" which took place January 25–30, 2015. The goal of the seminar was to advance the state of the art in experimental evaluation within the wider field of graph drawing, both with respect to user studies and algorithmic experimentation.

Seminar January 25–30, 2015 – http://www.dagstuhl.de/15052
1998 ACM Subject Classification E.1 Data Structures: Graphs and Networks, F.2 Analysis of Algorithms and Problem Complexity, G.3 Probability and Statistics: Experimental Design, H.5 Information Interfaces and Presentation
Keywords and phrases graph drawing, experimental design, algorithm engineering, user studies, empirical evaluation, information visualization
Digital Object Identifier 10.4230/DagRep.5.1.243

1 Executive Summary

Graph Drawing provides, among other things, the algorithmic foundations for network information visualization. It has considered implementation and experimentation as integral aspects from its very inception and recent research has demonstrated varying approaches to empirical evaluation. Experimental standards, however, have never been established, and little progress toward higher levels of sophistication can be observed.

The seminar was a community effort organized as a hands-on training event. It brought together experts on experimentation from fields with an established experimental tradition (referred to as "trainers"), and a group of graph drawing researchers expected to act as exponents and multipliers ("participants"). After two days of invited lectures on experimental methodology in different disciplines and a problem selection session, participants spent three days in working groups designing experiments. Trainers moving between groups and intermittent reporting session facilitated knowledge dissemination.

Participant feedback in the Dagstuhl survey indicates that the inclusion of trainers was highly appreciated. A number of experimental designs for a broad range of problems have been developed, and it is expected that many of them will be implemented and carried out in collaborative follow-up work.

As everyone who has ever been to Schloss Dagstuhl knows, Dagstuhl seminars are the ideal forum for achieving such goals. The fact that a considerable part of the graph drawing community came together for a week to focus on experimentation is expected to lead to a rapid diffusion of the seminar results and foster the acceptance of new methodology and criteria within the community.

On behalf of all participants, the organizer express their sincere gratitude to the Dagstuhl staff for their outstanding service and support.

Empirical Evaluation for Graph Drawing, *Dagstuhl Reports*, Vol. 5, Issue 1, pp. 243–258
Editors: Ulrik Brandes, Irene Finocchi, Martin Nöllenburg, and Aaron Quigley

Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Table of Contents

3 Organization

3.1 Introduction

Graph Drawing has a long tradition of implementing algorithms and evaluating their performance, maybe longer than other areas in algorithmics. An early example is the almost 20-year-old work of Himsolt [1], who evaluated twelve representative graph drawing algorithms based on a statistical analysis of geometric and structural properties of the respective layouts. Or the work of Di Battista et al. [2], who experimentally compared four algorithms for orthogonal grid drawings based on nine different quality measures. They introduced a benchmark suite, which is known today as the *Rome Graphs* and still frequently used in experimental graph drawing. The collection consists of more than 11,500 sparse graphs with fewer than 100 vertices generated from real-world graphs in software engineering and database systems.

This tradition may not come as a surprise given that Graph Drawing is a particularly interesting area for experimentation – an area that combines combinatorics, geometry, topology, algorithmics, visualization, interaction, and human factors. In this seminar, we are interested in two types of experiments which exhibit characteristics that are particularly challenging in graph drawing:

1. Experiments that compare graph-drawing algorithms in terms of domain-specific aesthetic optimization criteria (such as number of crossings, number of bends, angular resolution, crossings angles, layout area, uniformity of edge lengths, vertex distribution, or number of symmetries), and also in terms of running time and other more usual performance criteria.
2. Experiments that test how certain drawing styles help or hinder users to fulfill certain graph reading tasks. The difficulty of controlling for layout features sets the problem apart from other, more routinely conducted user studies in information visualization and human-computer interaction.

In spite of early and extensive work in these types of experiments, we think that it is time for the community to reconsider whether its experimental standards are up to date. We observe little progress in sophistication of experiments. One may ask whether this is because saturation has been reached early on but we doubt this. We simply think that knowledge about experimental methodology is not yet commonplace. Specifically, we identify the following problems:

1. In algorithmic studies, researchers often define the experimental region ad-hoc. They rarely ensure that the benchmark data is representative. Rather than generating good test instances or new benchmark sets, most researchers resort to the above-mentioned Rome graphs or to AT&T graphs, another benchmark suite.
2. It is rare that hypotheses are first explicitly formulated and then tested.
3. Phenomena that are observed during studies are usually explained post-hoc. Instead, such phenomena should lead to new hypotheses and to further experiments for validation.
4. There are only few specific graph generators.
5. Generally, there is a lack of user studies. Moreover, they tend to suffer from badly controlled factors in the instances presented to subjects.
6. Often, confounding factors and systematic biases are not identified.
7. Experiments are not convincingly randomized.
8. Statistical evaluation is generally rudimentary.

There were two main goals of this seminar. The first one was to increase the awareness of the need of high standards of empirical evaluations within the graph drawing community. We looked beyond graph drawing and learned from other fields with more advanced experimental research. Experimentation experts with experience in the closely related fields algorithm engineering, information visualization, and human-computer interaction, as well as experts from disciplines with more extensive experimental traditions who acted as trainers. By the close interaction with these experts, facilitated by a set of invited lectures and group discussions, we made a concerted effort for advancing the state of the art of experimental research in graph drawing. We aimed at establishing principles and experimental methodology by means of a guided knowledge import and an appropriate adaptation to the graph drawing context. Depending on progress with the second goal below, this shall result in a position paper.

The second goal was to actually design a set of empirical studies for answering experimental research questions in graph drawing. Within groups consisting of graph-drawing researchers as domain experts, and with experimentation experts floating among groups to offer advice, the newly acquired knowledge was applied to concrete problems. These problems had been collected in a special session following the invited tutorials. We hope that a large share of the ideas generated during this seminar will soon be implemented. Several groups and subgroups pledged to run some of the proposed experiments, evaluate them, and publish the results.

3.2 Schedule

As is customary, the schedule was organized around meals.

Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**
		breakfast		
welcome and introduction	invited talk discussion	topic selection group formation	trainer feedback working groups	working groups
		coffee		
invited talk discussion	topic collection	working groups	working groups	group reports final discussion
		lunch		
invited talk discussion		working groups	working groups	
		coffee & cake		
invited talk discussion	*excursion*	rapid feedback working groups	working groups	
		dinner		

3.3 Evaluation

According to the Dagstuhl survey conducted toward the end of the seminar, as well as informal feedback received by the organizers, the seminar was highly appreciated. Participants ranked the seminar higher than the (already hugely successful) average Dagstuhl seminar in virtually all dimensions, including scientific quality, group composition, the inspiration of new ideas for research and collaboration, and the acquisition of new knowledge. Increasing flexibility in the schedule and leaving more time for socializing are things to consider for future seminars.

Given the goals of the seminar it was by design that the group of participants had an overrepresentation of researchers who identify themselves as neither junior nor senior, and also happened to be relatively experienced Dagstuhl-goers, and an underrepresentation of industry.

The single most notable aspect in the qualitative feedback items was the excellent contribution of the trainers to the seminar.

We conclude that both the format and the content of the seminar worked as hoped for, despite adversarial circumstances that included the flu trying to compete with the spread of knowledge and new ideas.

References

1 Michael Himsolt. Comparing and evaluating layout algorithms within GraphEd. *J. Vis. Lang. Comput.*, 6(3):255–273, 1995.

2 Giuseppe Di Battista, Ashim Garg, Giuseppe Liotta, Roberto Tamassia, Emanuele Tassinari, and Francesco Vargiu. An experimental comparison of four graph drawing algorithms. *Computational Geometry*, 7(5–6):303–325, 1997.

4 Invited Presentations

We were fortunate to be joined by four enthusiastic and supportive domain experts in experimentation who fully embraced the goals of the seminar and immersed themselves into the group. One more trainer with a background in physics had to cancel on short notice.

4.1 The Art and Science of Evaluating Graph Layout Systems

Bernice E. Rogowitz (Visual Perspectives Research and Consulting – New York, US)

In graph layout evaluation, we measure the effectiveness of our computational methods and the degree to which our systems enable human effectiveness. In both cases, there are two evaluation strategies.

Methods	A-B Tests	Hypothesis-Generated Evaluations
Computational Evaluation		✓
Perceptual/Cognitive Evaluation		✓

One strategy is to evaluate performance using simple A-B tests, where design choices are compared for a particular set of stimuli using a particular task. In this paper, I make the case for hypothesis-driven evaluation, whose aim is to understand not only whether there is a difference between conditions, but why. In this more scientific approach, we select test conditions and tasks in order to test hypotheses about the underlying mechanisms driving observable distinctions, providing more generalizable results.

In this talk, I focus on perceptual and cognitive evaluation methods. Visualization systems, including graph layout methods, are designed to support human problem solving, judgment, decision-making, and pattern recognition. This means that every algorithm or tool we create embodies hypotheses about human information processing. Which perceptual and cognitive mechanisms are involved, however, depends on the task. For example, detection

and discrimination tasks involve early vision and tasks that require finding features in data require attention, semantic encoding, and memory. It is important to match the task to the perceptual and cognitive demands of the application.

I show examples of how evaluation experiments can be used to suggest new parameters for algorithms, test the ecological validity of different data representation schemes, and measure differences between intuitive and objective measures of visualization effectiveness. For example, I describe an experiment with Frank van Ham that shows how Gestalt principles of perceptual organization can enhance graph layout algorithms. When observers were allowed to move the nodes of a planar graph, they created configurations that emphasized clusters in the data. Their results looked quite similar to a force-minimization layout, but with less-uniform edge lengths, fewer line crossings, and increased distance between clusters. They even used edges to create convex hulls around clusters, enhancing their structure.

In an experiment with Mercan Topkara, Arum Hampapur and Bill Pfeiffer, I show how visual evaluation methods can be used to select the hit and false alarm rate of a surveillance algorithm to maximize human performance. And, in an experiment with Enrico Bertini, Aritra Dasgupta, Jorge Poco and Claudia Silva, I demonstrate how judgments of data magnitude and spatial distribution can be influenced by the choice of color scales. Surprisingly, the color scale with highest appeal, judged accuracy and familiarity provided the worst performance, demonstrating the importance of empirical testing.

4.2 Controlled Experiments in Software Engineering

Janet Siegmund (Universität Passau, DE)

Empirical research in psychology has come a long way. Thus, computer scientists who want to evaluate the human factor in their discipline, e.g., graph drawing or software engineering, can profit from the methodological advances in psychology. In this talk, I present a roadmap for conduction empirical studies of the human factor related to computer science research objectives. This roadmap is based on the state of the art of empirical research in psychology.

4.3 Designing Experiments in Political Science

Michael Stoffel (Universität Konstanz, DE)

This talk introduced the so-called "potential outcomes" framework that is the foundation of experimental research. Building on this, we discussed the two general assumptions that experimenters make and how to guarantee that they are satisfied: unconfoundedness and the stable unit treatment value assumption (SUTVA). In the empirical part of the talk, we then had a look at an experiment on principal-agent relations in the bureaucracy.

4.4 Experimental Algorithmics

Catherine C. McGeoch (Amherst College and D-Wave Systems Inc., US)

I talk about experimental methods that are aimed at algorithmic questions. These are more likely to be descriptive and exploratory (especially graphical) than confirmatory in nature – due to the common types of questions asked in algorithm studies. I review some possibilities for choosing performance indicators and show how variance reduction techniques can improve outcomes.

5 Working Groups

Initial ideas for topics to be worked on in groups were collected in the discussions following each invited presentation. The topics were reviewed, complemented, and consolidated in a special session at the end of the tutorial part. Self-assignment of participants to groups was surprisingly easy to deliberate.

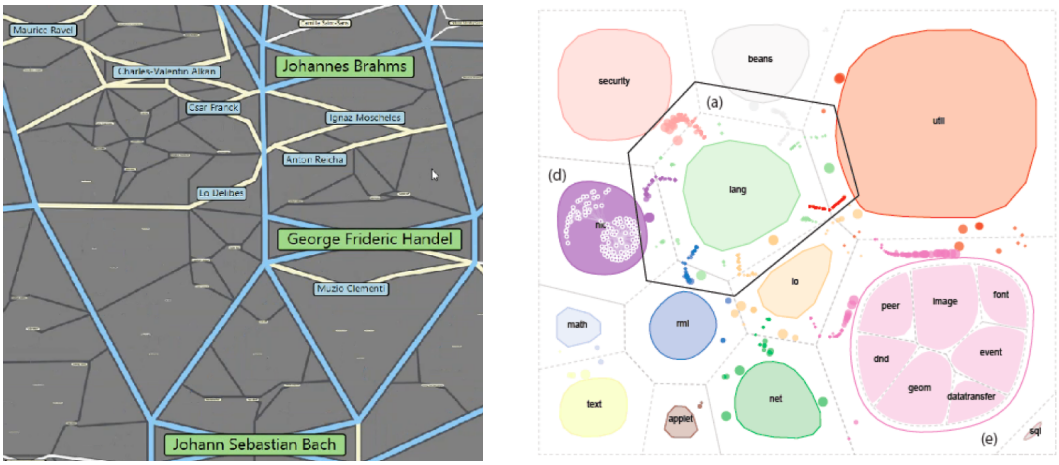
Working groups used the Dagstuhl Wiki environment to collect input, ideas, and outcomes, and the following are reports distilled from these entries.

5.1 Large Graphs

Irene Finocchi, Seokhee Hong, Lev Nachmanson, Huamin Qu, Alexander Wolff, and Kai Xu

Due to the finite resolution of display devices, which represents a physical limitation on the size of graphs that can be conveniently displayed, designing effective visual representations of large graphs poses many challenges. In order to cope with the cluttering effects arising when visualizing huge quantities of data, it seems important to use information hiding techniques and decompositions of the visual space that reflect some structural view of the data. The working group focused on two different visualization paradigms, both inspired by graph maps, that appear to be promising when dealing with large graphs. The implementation of these paradigms – called *graph maps with highways* and *clustering with maps*, respectively – was available in software tools co-designed by two of the group participants. For each paradigm, a different user study has been designed identifying both hypotheses and tasks to be performed by end users. The goal of the user studies is to compare the effectiveness of drawings produced by different algorithms according to the specific tasks, highlighting for which tasks each paradigm turns out to be most useful.

Graph maps with highways. For many real-world graphs with substantial numbers of edges, traditional algorithms produce visually cluttered layouts [3]. The relations between the nodes are difficult to analyze by looking at such layouts. Graph maps with highways, designed by Nachmanson *et al.*, exploit techniques similar to edge bundling to solve this problem. A visualization example of a graph map with highways is shown in Figure 1(a). The hypothesis of the user study is that the highway metaphor improves node location *memorability*, i.e.,



(a) **(b)**

Figure 1 (a) Graph map with highways on the composers' network from [1]; (b) Clustering with maps: an example taken from [2].

the speed of users to find previously visited nodes. During the study, each user must visit a given number of nodes, using pan & zoom, and then find one of the previously visited nodes as quickly as possible. Data used for the experiment include social networks and the composers' network used in the Graph Drawing 2011 competition [1].

Clustering with maps. This visualization paradigm, described in [2], has been designed with the goal of representing the major communities in large social networks. A visualization example is shown in Figure 1(b). In the designed study, end users are required to perform either inter-cluster tasks (e.g., counting clusters or finding the "most related" pair of nodes) or intra-cluster tasks based on zooming (e.g., finding an opinion leader inside a certain cluster). The study should test the following three hypotheses:

- a reasonable gap between the clusters increases the speed of counting clusters;
- if there are many nodes belonging to two different clusters, the gap between those clusters in the visualization should be larger;
- a big gap is good for inter-cluster tasks, while a small gap is more convenient for intra-cluster tasks.

References

1 Christian A. Duncan, Carsten Gutwenger, Lev Nachmanson, and Georg Sander. Graph drawing contest report. In *Graph Drawing – 19th International Symposium, GD 2011*, pages 449–455, 2011.

2 Yanhong Wu, Wenbin Wu, Sixiao Yang, Youliang Yan, and Huamin Qu. Interactive visual summary of major communities in a large network. In *IEEE Pacific Visualization Symposium, PacificVis 2015*, 2015. To appear.

3 Sergey Pupyrev, Lev Nachmanson, Sergey Bereg, and Alexander E. Holroyd. Edge routing with ordered bundles. In *Graph Drawing*, volume 7034 of *Lecture Notes in Computer Science*, pages 136–147. Springer, 2012.

5.2 Bends, Curves, and Bundles

Daniel Archambault, Martin Fink, Martin Nöllenburg, Yoshio Okamoto, and Ignaz Rutter

Edge bundling is a popular technique for reducing visual clutter in layouts of dense graphs [1, 2, 3, 4, 5, 6]. It is based on the idea of grouping edges whose end-vertices have similar locations into *bundles* and using appropriate graphical deformations to draw the edges of each bundle along the same underlying trunk path. While bundling techniques are claimed to successfully reduce edge clutter, the effect of bundled layouts on human graph readability is not yet well investigated and only very few studies have been published [7]. It may be argued that bundled layouts represent global trends well on a coarser scale, but there is also an unavoidable trade-off between cleaning up the layout by edge bundling and losing low-level connectivity information in the graph. This is because in an edge bundle it is often very difficult to trace individual edges and a dense bundle may be indistinguishable from a complete bipartite subgraph linking all pairs of vertices on both sides of the bundle. This is the main difference to the graph drawing style of *confluent layouts* [8]. In a confluent layout of a graph $G = (V, E)$, edges merge and split in smooth confluent junctions such that there is an edge between two vertices in G if and only if there is a smooth path between them in the layout.

In this working group we set out to design an empirical user study to evaluate the influence of edge bundling strength on typical graph reading tasks, both of global nature and detail-oriented ones. We hypothesize that edge bundling has a positive effect on tasks that require more global reasoning about the layout, but that it has a negative effect on tasks that require detailed knowledge about local structures of the graph. In order to measure and control bundling strength on a continuous scale, we defined an ambiguity measure as well as a measure for the amount of edge deformation. We plan to evaluate the task performance for four basic tasks that have been extracted from practical applications of graph visualization. Graph data will contain both geographic networks with fixed vertex positions as well as force-directed layouts of non-spatial networks, e.g., social networks. Since our aim is to evaluate edge bundling as a general technique and not a particular bundling algorithm, we will focus on a few representative algorithms that create explicit edge bundles so that we can vary the bundling strength by interpolating between the unbundled input layout and the fully bundled layout as computed by the selected algorithms. This necessarily excludes bundling methods that include explicit edge disambiguation techniques [9, 10].

Currently, in preparation of the planned user study, we are implementing an interpolation feature for varying the bundling strength into our selected edge bundling algorithms and we are collecting suitable real-world data sets for the study.

References

1 Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li. Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1277–1284, Nov 2008.

2 Danny Holten and Jarke J. van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009.

3 A. Lambert, R. Bourqui, and D. Auber. Winding roads: Routing edges into bundles. *Computer Graphics Forum*, 29(3):853–862, 2010.

4 E.R. Gansner, Yifan Hu, S. North, and C. Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 187–194, March 2011.

5 Quan Nguyen, Seok-Hee Hong, and Peter Eades. TGI-EB: a new framework for edge bundling integrating topology, geometry and importance. In Marc van Kreveld and Bettina Speckmann, editors, *Graph Drawing*, volume 7034 of *Lecture Notes in Computer Science*, pages 123–135. Springer Berlin Heidelberg, 2012.

6 Hong Zhou, Panpan Xu, Xiaoru Yuan, and Huamin Qu. Edge bundling in information visualization. *Tsinghua Science and Technology*, 18(2):145–156, April 2013.

7 Fintan McGee and John Dingliana. An empirical study on the impact of edge bundling on user comprehension of graphs. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI'12, pages 620–627, New York, NY, USA, 2012. ACM.

8 Matthew Dickerson, David Eppstein, Michael T. Goodrich, and Jeremy Y. Meng. Confluent drawings: Visualizing non-planar diagrams in a planar way. *Journal of Graph Algorithms and Applications*, 9(1):31–52, 2005.

9 Sheng-Jie Luo, Chun-Liang Liu, Bing-Yu Chen, and Kwan-Liu Ma. Ambiguity-free edge-bundling for interactive graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):810–821, May 2012.

10 Quirijn Bouts and Bettina Speckmann. Clustered edge routing. In *Pacific Visualization Symposium (PacificVis'15)*. IEEE, 2015. To appear.

5.3 Cognition

Rudolf Fleischer, Stephen G. Kobourov, Tamara Mchedlidze, Wouter Meulemans, Aaron Quigley, and Bernice E. Rogowitz

The cognition group started its graph drawing [1, 2] work asking a series of broad questions before digging down into specific details. We started with questions relating to aesthetic experience [3, 4, 5] such as "What is the purpose of the visualisation?," "analysis vs. communication," "exploratory vs confirmatory vs communication," "is it to tell a message?" or "is this to allow people to explore?" we moved onto discussing how visualisations can be artist [6], engaging, beautiful [7], attractive [8] and even perhaps arresting to draw people into use not just for an immediate short term reaction but instead a long term use. The question of making a visualisation arresting raises a number of questions including, does familiarity with a visual language breed contempt, is this a suitable goal, and does the layout or rendering impact the arresting nature of a display overall. If a visualisation can be arresting how does this affect long term use and memorability for ongoing use. Many of the visual affects discussed relate to the notion of a "honeypot effect" which maybe enough to draw someone into use but not long term engagement.

Next, the working group moved onto the discussion of static versus dynamic displays and interactive versus non-interactive displays. The notion of making a visualisation arresting is interlinked with is the displayed content static or dynamic. Further, does an arresting visualisation draw someone in with the expectations they might transition into interactive engagement? The design space moving from initial engagement into these factors opens up a large space.

The combination of this broad design space with the goal of hooking users in with arresting visualisations require careful thought on the measures [9] we can employ in the evaluation of suitability. The group moved onto a specific goal of exploring clusters in force directed graph drawings and what are the key aspects of visual appeal and performance. This brings into questions of shape, colour harmony, symmetry, geometry [10, 11] and gestalt principles [12, 13, 14, 15].

The outcome from the group is the design for an experiment which can be run in a lab and also online. A set of hypothesis around, appeal, visual principles, familiarity, subjective measures, display properties, geometric principles [16] have been formed. From this a series of experiments with small world graphs will measure, attraction, engagement and memorability [17, 18].

Our methods are based on the generation of drawings of small world-type graphs, a set of independent measures, dependent measures with various conditions and over 200 stimuli.

References

1. Frank van Ham and Bernice Rogowitz. Perceptual organization in user-generated graph layouts. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1333–1339, Nov 2008.
2. Miro Spönemann, Björn Duderstadt, and Reinhard von Hanxleden. Evolutionary meta layout of graphs. In Tim Dwyer, Helen Purchase, and Aidan Delaney, editors, *Diagrammatic Representation and Inference*, volume 8578 of *Lecture Notes in Computer Science*, pages 16–30. Springer Berlin Heidelberg, 2014.
3. Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. The aesthetics of graph visualization. In *Proceedings of the Third Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, Computational Aesthetics'07, pages 57–64, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
4. A. Lau and A. Vande Moere. Towards a model of information aesthetics in information visualization. In *Information Visualization, 2007. IV'07. 11th International Conference*, pages 87–92, July 2007.
5. Edward Vessel, Gabrielle Starr, and Nava Rubin. The brain on art: intense aesthetic experience activates the default mode network. *Frontiers in Human Neuroscience*, 66(6), 2012.
6. Werner Vogels. Graphs as art, 2006. Blog post, http://www.allthingsdistributed.com/2006/07/graphs_as_art.html.
7. Edward Vessel and Nava Rubin. Beauty and the beholder: Highly individual taste for abstract, but not real-world images. *Journal of Vision*, 10(2), 2010.
8. Julie Steele and Noah Iliinsk. *Beautiful Visualization: Looking at Data Through the Eyes of Experts*. 1st ed. O'Reilly Media, 2010.
9. Colin Ware, Helen Purchase, Linda Colpoys, and Matthew McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, June 2002.
10. Moshe Bar and Maital Neta. Humans Prefer Curved Visual Objects. *Psychological Science*, 17:645–648, 2006.
11. Roman Chernobelskiy, Kathryn I. Cunningham, Michael T. Goodrich, Stephen G. Kobourov, and Lowell Trott. Force-directed lombardi-style graph drawing. In Marc van Kreveld and Bettina Speckmann, editors, *Graph Drawing*, volume 7034 of *Lecture Notes in Computer Science*, pages 320–331. Springer Berlin Heidelberg, 2012.
12. Nadia Ali and David Peebles. The effect of gestalt laws of perceptual organization on the comprehension of three-variable bar and line graphs. *The Journal of the Human Factors and Ergonomics Society*, 55(1):183–203, Feb 2013.

13 Marlies de Brouwer. The influence of the gestalt principles similarity and proximity on the processing of information in graphs: An eye tracking study. Master's thesis, Tilburg University, 2014. http://arno.uvt.nl/show.cgi?fid=134020.

14 C.T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *Computers, IEEE Transactions on*, C-20(1):68–86, Jan 1971.

15 A. Rusu, A.J. Fabian, R. Jianu, and A. Rusu. Using the gestalt principle of closure to alleviate the edge crossing problem in graph drawings. In *Information Visualisation (IV), 2011 15th International Conference on*, pages 488–493, July 2011.

16 A. van Goethem, W. Meulemans, B. Speckmann, and J. Wood. Exploring curved schematization. In *2014 IEEE Pacific Visualization Symp. (PacificVis)*, pp. 1–8, March 2014.

17 Scott Bateman, Regan Mandryk, Carl Gutwin, Aaron Genest, David McDine, and Christopher Brooks. Useful junk? the effects of visual embellishment on comprehension and memorability of charts. In *ACM Conference on Human Factors in Computing Systems (CHI 2010)*, pages 2573–2582, Atlanta, GA, USA, 2010. Best paper award.

18 Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics (Proceedings of InfoVis 2013)*, 2013.

5.4 Computational Experiments

Ulrik Brandes, Emilio Di Giacomo, Andreas Karrenbauer, Karsten Klein, and Maurizio Patrignani

The group decided to design computational experiments for one specific research question. The question was identified based on various criteria such as relevance to the field of graph drawing, difficulty to be addressed analytically, expertise of the group members, and likelihood of leading to an actual study. This is what we came up with:

> Why do force-directed layouts exhibit relatively few crossings?

The approach taken to address this question is to fix a layout algorithm, identify structural features that are drawn with a crossings in optimal layouts, and relate the occurrence of such features in the input to crossings in the output.

Theoretical considerations allowed us to identify several families of problematic subgraphs, even though we conjecture that there exist trees without any of these subgraphs that still result in crossings. Based on these insights we derived four concrete hypothesis that are sufficiently specific to be testable.

Fortunately, these hypotheses give us a reason to test on planar graphs as for them all observed crossing are caused by the layout algorithm. A particularly important observation was made only because of the preceding discussion on experimental design: instead of simply charting the number of crossings we will determine the matches of problematic subgraphs and crossings, because these provide the evidence whether the crossings are actually caused by assumed mechanism.

Current challenges include fixing the set of problematic subgraphs to study, proving the above conjectures, random sampling of planar graphs, efficient counting of subgraphs, and identifying and controlling random and systematic biases due to the layout algorithm implementation.

5.5 Experiments Involving Humans

Markus Chimani, Walter Didimo, Michael Kaufmann, Giuseppe Liotta, and Dorothea Wagner

The group's main goal is to design an experiment centered around human understanding of graph drawings. A main goal thereby is to devise an experimental setup based on the fundamentals discussed in this workshop, to avoid traditional shortcomings often found in GD user studies. The study should hence not be an afterthought to theoretical research, but spur interest in and shed light on a topic that is typically rather left to intuition than to scientific rigor. Our naïve sounding question is:

Where should we put the arrow heads in directed graph drawings?

Despite the fact that we all draw directed graphs on a day-to-day basis, and that most (but not all) of us tend to draw the arrow heads at the edges' ends, it is unclear if this drawing paradigm is in fact the most suitable one. There are previous user studies discussing the drawing of directed edges, devising very different and diverse drawing approaches (even animated ones, unsuitable for printouts).

In contrast to those, we want to stick to the traditional arrow head paradigm, as there seems to be a large consensus that this paradigm is the most natural. However, when thinking about a vertex with large degree, traditionally end-placed arrow heads will overlap, making it hard or even impossible to identify the direction of a specific edge. We consider multiple different placement strategies to mitigate these effects. Some of these strategies give rise to interesting combinatorial optimization problems. However, our study will not discuss this; its outcome may, however, help to understand whether a detailed theoretical investigation of such a placement paradigm is at all worthwhile.

We spent a large percentage of our time at Dagstuhl devising our hypotheses and the tasks given to the users, discussing their interplay, and trying to find an as small set of tasks as possible, while still covering all our hypotheses in a meaningful way. Especially this minimization – necessary to end up with a feasibly conductible experiment – turned out to be harder than anticipated.

A further important fact for our user study is to specify our underlying graphs and drawings, which are to be based on real-world scenarios but yet controllable enough for a user study. Detailed discussions have taken place with respect to the various confounding factors, the experimental design and setup, and a time-plan for the next steps, including pilot studies to hammer out the finer choice details prior to the main experiment. The final steps towards the experimental study are currently in progress.

5.6 Tasks Linked to Representations

Tim Dwyer, Tamara Munzner, Falk Schreiber, Bettina Speckmann, and Matt F. Stallmann

In discussing "Experimental Graph Drawing," we felt that before a meaningful experiment could be designed, it was necessary to understand what the most important tasks in graph

drawing and network visualization really are. Once the important tasks and challenges facing network analysts are properly understood then we can begin to test various methods for creating visualizations of networks that actually support those tasks.

We began with a cursory literature search for task surveys and taxonomies, we found a couple but not that were very deep. Lee et al. [1] give a brief and rather overview of some reasonable sounding tasks, particularly *low-level tasks* such as path following, common neighbours, etc. However, high-level tasks and in-particular, how these translate into real-world problem solving we felt was missing. Pretorius et al. [2] give a longer discussion of tasks for *multivariate network analysis*, however, again it seems unclear precisely when these tasks translate into "aha moments" in analysis. Munzner [3] gives a "how/what/why" framework for problem solving with graph visualization that perhaps gets closer to consideration of real-world applications, but this part of the book is much briefer in relation to network visualization than other types of data visualization.

In summary then, we feel that there is room for a deeper analysis of tasks starting from applications and working down to a generalizable taxonomy. Such a taxonomy should identify not only very low-level tasks (ones that could be considered atomic, e.g. "is node A connected to node B?" but also mid-level tasks that are very application agnostic but composed of multiple low-level operations (e.g., "what is the shortest path between A and B?") and high-level tasks more specific to applications (e.g., "what is the critical path in this workflow, which deadlines can slip without jeopardizing the entire project?" Armed with such a taxonomy we are ready to consider how visualization can help or even if it is always the best method for particular scenarios.

The example above is quite a concrete connectivity task that requires close inspection of precise connectivity information, but it also seems clear that people are interested in understanding much larger-scale graph structure. Examples of applications where the practitioners want some insight into the gross structure of very large networks with thousands, tens-of-thousands or even millions of nodes abound. For example, in Biology metabolic networks contain thousands of nodes and gene activity correlation networks contain tens-of-thousands of nodes, in neuroscience neural networks, economics... really any complex system considered by modern scientists and other analysts has a scale issue. Much work has been done in making algorithms and rendering processes scale to thousands or millions of nodes and links in an efficient way. Less well understood is exactly how these large-scale visualizations help practitioners, especially when large, naturally occurring networks when rendered as node link diagrams, tend to appear as "hairballs" or when rendered as matrices, as "white noise".

On this note, our discussion then diverged to consideration of alternatives for understanding gross network structure that might avoid many of these drawing pitfalls entirely. We discussed the possibility of computing summary statistics for graphs that can convey the high-level structure of networks concisely yet adequately to give practitioners the information they need about large networks. For example, a force-directed layout of a very large network may turn out to be a hairball which only tells you that the network has many nodes and links and it is likely small-world and scale free. However, you can't be sure of even these properties without further investigation of the node-degree distribution and the network diameter. So, why not by-pass the precise node-link diagram entirely and make the first visualization be a succinct dashboard display of statistics such as these?

This led to an extensive study of network-diagnostic statistics, or *NetNostics* or even *NetGnostics*. We found that there is extensive literature and the theory and practice of statistical analysis of networks, a book by Kolaczyk [4] gives a good overview and introduction.

Yet, such statistics are rarely the focus of visualization – particularly not in the "dashboard" view we envisage.

In summary then, we plan to proceed on two fronts. First, we will continue to work towards a deeper survey and taxonomy of tasks which we feel will be an important practical contribution to the emerging field of experimental graph drawing in giving a solid motivation and foundation for designing studies with ecological validity. Second, we will survey the field of statistical analysis of network structure – with a short term goal of publishing a survey that is useful to information visualization researchers but in the longer term, will would like to produce a practical "dash-board" system as described above.

William Hill and Bernice Rogowitz contributed to subsequent discussion.

References

1 Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV'06, pages 1–5, New York, NY, USA, 2006. ACM.

2 A. Johannes Pretorius, Helen C. Purchase, and John T. Stasko. Tasks for multivariate network analysis. In *Multivariate Network Visualization*, volume 8380 of *Lecture Notes in Computer Science*, pages 77–95. Springer-Verlag, 2014.

3 Tamara Munzner. *Visualization Analysis and Design*. A.K. Peters visualization series. CRC Press, 2014.

4 Eric D. Kolaczyk. *Statistical Analysis of Network Data*. Springer-Verlag, 2009.

Participants

- Daniel Archambault
Swansea University, GB
- Ulrik Brandes
Universität Konstanz, DE
- Markus Chimani
Universität Osnabrück, DE
- Emilio Di Giacomo
University of Perugia, IT
- Walter Didimo
University of Perugia, IT
- Tim Dwyer
Monash Univ. Melbourne, AU
- Martin Fink
University of California – Santa Barbara, US
- Irene Finocchi
University of Rome 'La Sapienza", IT
- Rudolf Fleischer
German University of Technology – Oman, OM
- Seok-Hee Hong
The University of Sydney, AU
- Andreas Karrenbauer
MPI für Informatik – Saarbrücken, DE
- Michael Kaufmann
Universität Tübingen, DE
- Karsten Klein
Monash University, AU
- Stephen G. Kobourov
Univ. of Arizona – Tucson, US
- Giuseppe Liotta
University of Perugia, IT
- Catherine C. McGeoch
D-Wave Systems Inc. & Amherst College, US
- Tamara Mchedlidze
KIT – Karlsruher Institut für Technologie, DE
- Wouter Meulemans
Universität Münster, DE
- Tamara Munzner
University of British Columbia – Vancouver, CA
- Lev Nachmanson
Microsoft Corp. – Redmond, US
- Martin Nöllenburg
KIT – Karlsruher Institut für Technologie, DE
- Yoshio Okamoto
The University of Electro-Communications – Tokyo, JP
- Maurizio Patrignani
Roma Tre University, IT
- Huamin Qu
HKUST – Kowloon, HK
- Aaron Quigley
University of St. Andrews, GB
- Bernice E. Rogowitz
Visual Perspective – New York, US
- Ignaz Rutter
KIT – Karlsruher Institut für Technologie, DE
- Falk Schreiber
Monash University, AU
- Janet Siegmund
Universität Passau, DE
- Bettina Speckmann
TU Eindhoven, NL
- Matthias F. Stallmann
North Carolina State Univ., US
- Michael Stoffel
Universität Konstanz, DE
- Dorothea Wagner
KIT – Karlsruher Institut für Technologie, DE
- Alexander Wolff
Universität Würzburg, DE
- Kai Xu
Middlesex University, GB