DAGSTUHL
REPORTS

Volume 8, Issue 5, May 2018

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
January, 2019

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.8.5.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 18201

Secure Compilation

Edited by

Amal Ahmed[1], Deepak Garg[2], Catalin Hritcu[3], and Frank Piessens[4]

1 **Northeastern University and INRIA – Paris, FR,** amal@ccs.neu.edu
2 **MPI-SWS – Saarbrücken, DE,** dg@mpi-sws.org
3 **INRIA – Paris, FR,** catalin.hritcu@gmail.com
4 **KU Leuven, BE,** frank.piessens@cs.kuleuven.be

Abstract

Secure compilation is an emerging field that puts together advances in security, programming languages, verification, systems, and hardware architectures in order to devise secure compilation chains that eliminate many of today's vulnerabilities. Secure compilation aims to protect a source language's abstractions in compiled code, even against low-level attacks. For a concrete example, all modern languages provide a notion of structured control flow and an invoked procedure is expected to return to the right place. However, today's compilation chains (compilers, linkers, loaders, runtime systems, hardware) cannot efficiently enforce this abstraction: linked low-level code can call and return to arbitrary instructions or smash the stack, blatantly violating the high-level abstraction. The emerging secure compilation community aims to address such problems by devising formal security criteria, efficient enforcement mechanisms, and effective proof techniques.

This seminar strived to take a broad and inclusive view of secure compilation and to provide a forum for discussion on the topic. The goal was to identify interesting research directions and open challenges by bringing together people working on building secure compilation chains, on developing proof techniques and verification tools, and on designing security mechanisms.

Seminar May 13–18, 2018 – http://www.dagstuhl.de/18201
2012 ACM Subject Classification Security and privacy → Formal security models
Keywords and phrases secure compilation, low-level attacks, source-level reasoning, attacker models, full abstraction, hyperproperties, enforcement mechanisms, compartmentalization, security architectures, side-channels
Digital Object Identifier 10.4230/DagRep.8.5.1
Edited in cooperation with Roberto Blanco (INRIA – Paris, FR)

1 Executive Summary

Amal Ahmed (Northeastern University and INRIA – Paris, FR)
Deepak Garg (MPI-SWS – Saarbrücken, DE)
Catalin Hritcu (INRIA – Paris, FR)
Frank Piessens (KU Leuven, BE)

Today's computer systems are distressingly insecure. The semantics of mainstream low-level languages like C and C++ is inherently insecure, and even for safer languages, establishing security with respect to a high-level semantics does not prevent devastating low-level attacks. In particular, all the abstraction and security guarantees of the source language are currently lost when interacting with lower-level code, for instance when using low-level libraries. For a

Secure Compilation, *Dagstuhl Reports*, Vol. 8, Issue 05, pp. 1–30
Editors: Amal Ahmed, Deepak Garg, Catalin Hritcu, and Frank Piessens
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

concrete example, all modern languages provide a notion of structured control flow and an invoked procedure is expected to return to the right place. However, today's compilation chains (compilers, linkers, loaders, runtime systems, hardware) cannot efficiently enforce this abstraction: linked low-level code can call and return to arbitrary instructions or smash the stack, blatantly violating the high-level abstraction.

Secure compilation is an emerging field that puts together advances in security, programming languages, compilers, verification, systems, and hardware architectures in order to devise secure compiler chains that eliminate many of today's low-level vulnerabilities. Secure compilation aims to protect high-level language abstractions in compiled code, even against low-level attacks, and to allow sound reasoning about security in the source language. The emerging secure compilation community aims to achieve this by:

1. **Identifying and formalizing secure compilation criteria and attacker models.** What are the properties we want secure compilers to have, and under what attacker models? Should a secure compilation chain preserve observational equivalence of programs? Should it preserve some class of security properties of the source programs? Should it guarantee invariants on the run-time state of the compiled program (like for instance well-formedness of the call-stack)? And what are realistic attacker models? Can attackers only interact with compiled programs by providing input and reading output? Or can they link arbitrary low-level code to the program? Well-studied notions like fully abstract compilation provide partial answers: a fully abstract compiler chain preserves observational equivalence under an attacker model where attackers are target-level contexts. Even where this is the desired end-to-end security goal, it can still be too hard to enforce, for instance in cases where target level contexts can measure time.
2. **Efficient enforcement mechanisms.** The main reason today's compiler chains are not secure is that enforcing abstractions in low-level compiled code can be very inefficient. In order to overcome this problem, the secure compilation community is investigating various efficient security enforcement mechanisms: from the use of static checking of low-level code to rule out linking with ill-behaved contexts, to software rewriting (e.g., software fault isolation), dynamic monitoring, and randomization. One key enabler is that hardware support for security is steadily increasing.
3. **Developing effective formal verification techniques.** Secure compilation properties like full abstraction are generally much harder to prove than compiler correctness. Intuitively, in order to show full abstraction one has to be able to back-translate any low-level context attacking the compiled code to an equivalent high-level context that can attack the original source code. This back-translation is, however, nontrivial, and while several proof techniques have been proposed (e.g., based on logical relations, bisimulations, game semantics, multi-language semantics, embedded interpreters, etc.), scaling these techniques to realistic secure compilers is a challenging research problem. This challenge becomes even more pronounced if one expects a strong level of assurance, as provided by formal verification using a proof assistant.

The Secure Compilation Dagstuhl Seminar 18201 attracted a large number of excellent researchers with diverse backgrounds. The 45 participants represented the programming languages, formal verification, security, and systems communities, which led to many interesting points of view and enriching discussions. Some of these discussions were ignited by the "guided discussions" on the 3 aspects above and by the 35 talks contributed by the participants. The contributed talks spanned a very large number of topics: investigating

various secure compilation criteria and attacker models, building prototype secure compilation chains, proposing different enforcement techniques, studying the relation to verified compilation and compositional compiler correctness, specifying and restricting undefined behavior, protecting against side-channels, studying intermediate representations, performing translation validation, securing multi-language interoperability, controlling information-flow, compartmentalizing software, enforcing memory safety, compiling constant-time cryptography, securing compiler optimizations, designing more secure (domain-specific) languages, enforcing security policies, formally specifying the semantics of realistic languages and ISAs, compartmentalization, capability machines, tagged architectures, integrating with existing compilation chains like LLVM, making exploits more difficult by diversification, multi-language interoperability, etc. Talks were interspersed with lively discussions, since by default each speaker could only use half of the time for presenting and had to use the other half for answering questions and engaging with the audience.

Given the high interest spurred by this first edition and the positive feedback received afterwards, we believe that this Dagstuhl Seminar should be repeated in the future. Particular aspects that could still be improved in future editions is focusing more on secure compilation and spurring more participation from the practical security and systems communities.

2 Table of Contents

3 Guided Discussions

3.1 What Is Secure Compilation? Security Goals and Attacker Models

Discussion led by Catalin Hritcu (INRIA – Paris, FR)

Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.CatalinHritcu.Slides.pdf

In the broadest sense, the goal of secure compilation research is to devise more secure compilation chains. Since there are many different ways to define "more secure," there are also many different notions of secure compilation. This discussion was aimed at identifying various different security goals and attacker models for secure compilation chains. Here we use the term "compilation chain" to include not just the compiler, but also the linker, loader, runtime, operating system, hardware, and security enforcement mechanisms at any of these levels. We do this since the responsibility of enforcing secure compilation often does not rest just with the compiler, but is shared by various parts of the compilation chain. For instance, achieving memory safety requires not only changing the compiler, but also most other components of the compilation chain have to at least be taught that pointers are not integers, and to achieve efficient enforcement the hardware needs to be extended as well.

So what are some of the possible security goals and attacker models for secure compilation chains? A first class of secure compilation chains aim at providing a "safer" semantics for unsafe low-level languages like C and C++, whose standard semantics call out a large set of undefined behaviors for which compilers can produce code that behaves arbitrarily, often leading to exploitable vulnerabilities. For instance, memory safety is aimed at turning spatial and/or temporal memory violations–e.g., buffer overflows, use after free–into safe behavior–e.g., raising an exception or terminating the program. Similarly, type safety can ensure that invalid casts are always detected and do not cause undefined behavior. The standard attacker model for type and memory safety protects against an external adversary that provides malicious, often malformed, inputs into the program and tries to hijack control, corrupt or disclose data, etc.

Ideally, one would like to turn as much undefined behavior in C and C++ as possible into safe behavior. However, especially when done solely in software, this can have a very high performance cost. So most security defenses that are widely deployed today are mitigations focused not at making languages like C and C++ safe, but instead at making exploiting security vulnerabilities more difficult: control-flow integrity, data-flow integrity, code-pointer integrity, lightweight stack protection, randomization. The attacker model for these mitigations is that the attacker can send inputs that exploit a particular class of vulnerabilities: for instance the attacker can use a buffer overflow to access memory via say contiguous writes or arbitrary reads. The goal of the attacker is then to inject code or behavior, to corrupt or leak data, while avoiding the mitigations in place. With enough effort a motivated attacker can usually achieve just this, and the goal here is only to increase the attacker effort, not to provide watertight guarantees.

In contrast, compartmentalization (e.g., software-fault isolation) is a mitigation technique that does provide watertight guarantees. The security goal of compartmentalization is to limit the damage of an attack only to the compromise of the components encountering undefined behavior. In particular, compartmentalization can be applied in unsafe low-level languages to structure large, performance-critical applications into mutually distrustful components that have clearly specified privileges and interact via well-defined interfaces. Intuitively,

protecting each component from all the others should bring strong security benefits, since a vulnerability in one component need not compromise the security of the whole application.

The applications of compartmentalization are, however, much broader. One can use compartmentalization to, for instance: (1) protect a trusted host application from untrusted plugins or libraries that could be malicious (e.g., as usually done for securing securing web browsers plugins, etc.); (2) protect a secure enclave from a malicious host (e.g., Intel SGX, ARM TrustZone, Sancus, Sanctum, etc.); or (3) protect mutually distrustful components written in an unsafe language against each other (e.g., as done for achieving least privilege design with process-based isolation, SFI, capability machines, tagged architectures, etc). In all these scenarios a minimal security goal is to preserve the integrity of the code and data of each component from malicious or compromised code in the other components. In addition one could also aim that no component can infer the secrets of other components, other than communicating with them through their high-level interface. This is particularly challenging though when bad components can also observe side-channels like execution time. Finally, one could also aim at protecting the availability of critical components, ensuring that others cannot cause crashes or hangs.

The common way of formalizing the security guarantees of compartmentalization is in terms of the source-level security reasoning principles it enables. Reasoning about the security in the source language (or "the safe part" of the source language, without undefined behaviors) is useful because then one does not need to worry about low-level attacks that can only happen at the target level, since one knows the abstractions of the source language are implemented in a watertight way by the secure compilation chain. A good way to formalize this is in terms of preserving various classes of security property during compilation: (1) trace properties such as safety and liveness, (2) hyperproperties such as noninterference, or (3) relational hyperproperties such as trace equivalence and observational equivalence.

An important point in the discussion was that specially designed source languages or source language extensions could make it easier to precisely specify the intended security properties, so that the secure compilation chain only needs to preserve those, and can thus be more efficient than if trying to preserve a large class of properties. For instance, explicitly annotating what is the secret data that external observers or other components should not be able to obtain, maybe even using side-channels like timing, gives the compilation chain the freedom to more efficiently handle any data that is not influenced by secrets.

We end this report summary with some interesting questions raised in our discussion:

- What are meaningful security properties to preserve in a particular application domain?
- When is it more meaningful for secure compilation chains to preserve large classes of properties (in which case, one doesn't don't need to specify much at the source level), and when is it more meaningful to preserve application-specific security properties?
- How much can program verification help and what are its scalability limitations?
- How does one go about preserving security intent all the way to the hardware and how can one convince hardware manufacturers to use this information for security?
- Can domain-specific languages make certain properties easier to achieve?
- In cases where security is not just binary, can we properly quantify the notion of attacker cost, taking maybe inspiration from cryptographic proofs?
- What are low-cost mitigations for (the lack of) full abstraction, maybe inspired by current mitigations for (the lack of) memory safety?

3.2 Effective Enforcement Mechanisms for Secure Compilation

Discussion led by Frank Piessens (KU Leuven, BE)

Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.FrankPiessens.Slides.pdf

This discussion session started from the observation that there is a wide variety of enforcement mechanisms, including hardware based mechanisms (such as processor privilege levels, virtual memory, capabilities, trusted computing, ...), software based mechanisms (including techniques such as type checking, static analysis, program verification, run-time monitoring, taint tracking, ...), and cryptographic mechanisms.

A first question that was discussed is where this wide variety of techniques is still insufficient. Several areas where there is need for novel kinds of enforcement mechanisms were discussed, the most prominent being the area of protecting against micro-architectural side-channel attacks.

A second topic that was addressed during the discussions is the issue of passing security information across abstraction layers, and in particular the question of what security information should be passed down to the compiler and further down to the hardware by software source code. Should software engineering inspired abstraction mechanisms be enforced as security boundaries after compilation? Or should specific security annotations be added to the source code to inform the compiler and the hardware about what security boundaries to enforce?

Finally, the discussion focused on trade-offs (for instance, expressivity versus performance versus complexity) of different enforcement techniques.

3.3 Formal Verification and Proof Techniques for Secure Compilation

Discussion led by Amal Ahmed (Northeastern University and INRIA – Paris, FR)

Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.AmalAhmed1.Slides.pdf

This session involved discussion of what kinds of formalisms and proof techniques might be needed for verifying secure compilation and compositional compiler correctness.

The first issue discussed was what it would take to extend existing verified compilers into secure compilers and compositionally correct compilers. Here "secure compiler" might encompass various different notions of security, e.g., resistant to side-channel attacks, satisfying robust safety preservation, fully abstract, and so on. Two central questions raised were: (1) when do we need entirely new proof architectures, and (2) what are good strategies for reusing mechanized proof efforts. The following points were raised during this part of the discussion:

- Taking CompCert as an exemplar, it was posited that a central challenge is stating the security properties we want. A related issue (discussed earlier at the seminar) is that C is not a language in which programmers can express their "security intent" so we either need to (a) have the compiler writer decide what security properties to enforce or (b) provide programmers with compiler flags or program annotations so they can communicate their security intent to the compiler.

- Proof architectures such as CompCert's have proved fairly reusable, but whether we can keep using refinement and simulation style proofs will likely depend on the security properties we wish to establish.
- There's a question of what security architectures are employed for enforcing security properties in a compiler like CompCert, e.g., capability machines like CHERI or tag-based architectures. There may be potential difficulties here since these mechanisms may be quite different from CompCert's memory model.
- A significant issue that must be taken into account when extending existing verified compilers into secure compilers is that the correctness of certain compiler optimizations will be influenced by the attacker model. This might complicate the statements of theorems as well as the proofs themselves.

The second issue discussed was about better techniques for compositional compiler correctness. While there are a number of existing techniques–e.g., multi-language semantics, cross-language logical relations, PILS, interaction semantics with structured simulations–it would be useful to have guidelines about which technique is suitable when. One example is that while the language-independent interaction semantics used by Compositional CompCert works for CompCert–where the source, intermediate, and target languages use the same memory model–it's unclear how to extend it to compilers where the languages have different memory models. There are also open questions of how to reduce the effort involved in multi-pass compilers–i.e., making it easier to prove transitivity (or vertical compositionality) for a compositionally correct compiler–and the related question of how to do prove transitivity when different passes of the compiler are verified using different proof techniques. Finally, it would be nice to have some common infrastructure for mechanizing proofs. The following additional points were raised in the discussion:

- Compositional compiler correctness requires reasoning about the behavior of the target-level (assembly) code that a compiled component is linked with. But if we take that target code to simply be assembly then we may have a difficult reasoning problem, as well as a highly powerful attacker model.
- Can we impose constraints on the attacker even when it is assembly code? Yes, we can either impose constraints statically or dynamically through mechanisms such as SFI, capabilities, or putting code in enclaves as in SGX.
- Once target contexts (attackers) are somewhat constrained using static or dynamic mechanisms, we must correctly model the power of target-level attackers. The multi-language semantics approach gives one way to do this. Different kinds of multi-language semantics can be set up to allow more restricted or less restricted interactions between target contexts and source (or compiled) components. Another strategy might be to come up with the right abstractions to add to the source level that correctly model the additional power of the target contexts/attackers. This is similar to the "linking types" idea presented at the seminar.
- The ultimate goal for secure compilation is to have compiled code that does not have attacks. If we could lift that specific security goal into a proof obligation at the source level, what would that proof obligation be? If we can specify what abstractions we must add to the source to model additional attacker power, then at least we know exactly what we must protect our source components against.
- Over time, as we encounter new attacks, we may want to adapt our proofs of compositional compiler correctness or secure compilation to new attacker models. But this will probably be quite challenging since this is an instance where the tension between horizontal and vertical compositionality might come into play.

The final issue brought up was regarding proof methods that help reason about the power of the adversary in secure compilation. In particular, context-based back-translation has been widely used, but back-translation techniques differ depending on how different the source and target languages are from each other, whether they are Turing-complete or not, and whether the back-translation used is syntax-based or trace-based. Trace-based back-translations might be easier to reuse across languages.

4 Working Groups

4.1 Meltdown and Spectre Attacks

Discussion led by Chris Hawblitzel (Microsoft Research – Redmond, US-WA)

Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.ChrisHawblitzel.Slides.pptx

The discussion was illustrated by a (contrived) Spectre example. Consider an array of flat pointers, not secret, each pointing to an integer, also not secret. In memory, the array is followed by some secret data. In performing a standard, safe loop over the elements of the array, the hardware may speculatively go beyond the bounds of the array and execute the next potential iteration, therefore loading the secret into the cache and exposing it to an adversary.

For the described example, potential mitigations include the modular indexing of elements in the array, which are then accessed modulo the length of the array. Attempts to implement such fixes can be performed at the source level or directly in the compiler (or even on hardware)–it should be noted that interval analysis, already implemented, say, in JavaScript compilers, allows a compiler to patch up this vulnerability.

The discussion revolved around two main questions:

4.1.1 What Can Software Do?

Several ideas were given:

- For conditional branches: clamping or sandboxing array accesses (as in the motivating example).
- For unconditional jumps: turning speculation off, other measures?
- In general, formal reasoning requires a hardware model: e.g., an operational semantics that nondeterministically speculates on conditional branches (however, the necessity of establishing a fruitful dialogue with architects was noted). A suggested example would involve adding a cache to the operational semantics and stating properties about that cache. The obvious question will be whether a model is good enough to prevent attacks in practice.

Regardless, it was noted that speculation may not need to be turned off completely, though these considerations may be application-dependent.

4.1.2 What Can Hardware Do?

Again, several lines of discussion were considered:

- Partitioning of shared resources: the software would decide which data goes into which partition (and the hardware would be responsible for providing partitions, each with its own cache and resources).
- Protection of secret data: the software would mark some addresses and data as sensitive.
- Side channel avoidance without software help: a more speculative idea, by which the hardware would avoid committing changes to shared resources until all relevant speculation were resolved.

As in the discussion of software-based strategies, the need to engage in discussion with more architects was noted.

An upcoming *Panel on the implications of the Meltdown & Spectre design flaws* (http://iscaconf.org/isca2018/panel.html) was mentioned, where the state of affairs was tentatively summarized as:

> Computer Architecture 1.0 specifies the timing-independent functional behavior of a computer, while Micro-Architecture is the implementation techniques that improve performace. What if a computer that is completely correct by Architecture 1.0 can be made to leak protected information via timing, a.k.a., Micro-Architecture?

4.2 C Semantics in Depth

Discussion led by Peter Sewell (University of Cambridge, GB)

The working group revolved around a discussion of pointer provenance and uninitialised reads in the C programming language. The attendance included the following participants:

- Frédéric Besson
- David Chisnall
- John T. Criswell
- Chung-Kil Hur
- Xavier Leroy
- Santosh Nagarakatte
- Steve Zdancewic
- Roberto Blanco
- Daniel Patterson
- Andrew Tolmach
- Peter Sewell

5 Overview of Talks

5.1 Compositional Compiler Correctness and Secure Compilation: Where We Are and Where We Want to Be

Amal Ahmed (Northeastern University – Boston, US and INRIA - Paris, FR)

Joint work of Amal Ahmed, Daniel Patterson
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.AmalAhmed.Slides.pdf

In this talk, I'll start with a brief but insightful survey of recent compositional compiler correctness results. I'll give a high-level perspective on what is good and bad about each of the existing compositional compiler correctness results and how their formalisms influence the required verification effort. I'll explain why *none* of the compositional compiler correctness results to date are where we want to be!

Then I'll present a generic compositional compiler correctness (CCC) theorem that abstracts away from existing formalisms. CCC gives us insight on what is required for modular verification of multi-pass compilers.

I will end with an insight for those working on secure compilation results that require "weaker" protection of compiled components than fully abstract compilation: when it comes to proving such compilers correct, truly modular verification of multi-pass compilers seems impossible.

5.2 Thoughts on Preserving Abstractions

Nick Benton (Facebook Research – London, GB)

The talk discussed the principle that high-level reasoning as performed by a programmer or a compiler should remain valid when the compiled code runs in a real environment. To this end, the behavioral characterization of interface contracts should be independent of the source language and the compiler, modulo calling and linking conventions. It is thus necessary to agree on a language to express the aforementioned interface contracts. The end result of a high-level denotational semantics with a low-level operational behavior will be a proof obligation to show that a piece of code behaves like some given mathematical function.

When reasoning about preservation of abstractions, it was observed that there is always, in fact, an appeal to some form of denotational semantics, whether this is explicitly acknowledged or not. In current practice, many such treatments are not fully abstract, but nonetheless, weaker, "good enough" notions of abstraction are routinely used to good effect, provided that they offer sufficient abstraction for the task at hand–in fact, some of the finer points of full abstraction are often not very useful in practice, nor are they well-understood in their full generality.

Finally, a mixed-language approach to the problem was discussed, noting the risk of breaking abstractions too severely, and the difficulties and costs incurred by some potential mitigations to that risk. However, the difficulties of preserving abstractions express themselves fully in the presence of higher-order and/or fairly strong notions of purity, whereas most foreign function interfaces are, rather, first-order in nature–and when they are not, the type system offers assistance in those parts that cross the boundary.

5.3 Secure Compilation of Safe Erasure

Frédéric Besson (INRIA – Rennes, FR)

Joint work of Frédéric Besson, Thomas Jensen, Alexandre Dang
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.Fr%C3%A9d%C3%A9ricBesson.Slides.pdf

Secure coding requires erasing secrets to limit the possibility for an attacker to probe the content of memory. At source level, erasure is typically performed by a `memset (secret,0)`. Yet, as secret is dead, compiler optimisations may remove this piece of code and therefore break the security.

In the talk, I tested on the audience a semantics definition of (preservation) of safe erasure phrased in terms of quantitative information flow. I then sketched how typical compiler optimisations (DSE, register allocation) need to be modified to preserve this property.

5.4 CompCertSFI

Frédéric Besson (INRIA – Rennes, FR)

Joint work of Frédéric Besson, Sandrine Blazy, Alexandre Dang, Thomas Jensen

We describe the design, implementation and proof of an efficient, machine-checked CompCert implementation of Portable Software Fault Isolation. We propose a novel sandboxing transformation that has a well-defined C semantics and which supports arbitrary function pointers. Our experiments show that our formally verified technique is a competitive way of implementing Software Fault Isolation.

5.5 Memory Safety for Shielded Execution

Pramod Bhatotia (The University of Edinburgh, GB)

Joint work of Dmitrii Kuvaiskii, Oleksii Oleksenko, Sergei Arnautov, Bohdan Trach, Pramod Bhatotia, Pascal Felber, Christof Fetzer
Main reference Dmitrii Kuvaiskii, Oleksii Oleksenko, Sergei Arnautov, Bohdan Trach, Pramod Bhatotia, Pascal Felber, Christof Fetzer: "SGXBOUNDS: Memory Safety for Shielded Execution", in Proc. of the Twelfth European Conference on Computer Systems, EuroSys 2017, Belgrade, Serbia, April 23-26, 2017, pp. 205–221, ACM, 2017.
URL http://dx.doi.org/10.1145/3064176.3064192

In this talk, I will first present our work on SGXBounds on how to achieve lightweight memory safety in the context of SGX Enclaves.

I will conclude the talk with our on-going work on Intel MPX Explained: https://intel-mpx.github.io/

5.6 Software Diversity vs. Side Channels

Stefan Brunthaler (Universität der Bundeswehr – Munich, DE)

Main reference Stephen Crane, Andrei Homescu, Stefan Brunthaler, Per Larsen, Michael Franz: "Thwarting Cache Side-Channel Attacks Through Dynamic Software Diversity", in Proc. of the 22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015, The Internet Society, 2015.

URL https://www.ndss-symposium.org/ndss2015/thwarting-cache-side-channel-attacks-through-dynamic-software-diversity

Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.StefanBrunthaler.Slides.pdf

The past couple of years have seen attacks becoming increasingly sophisticated, primarily due to the discovery and incorporation of side channels. Among others, Drammer, AnC, and SPECTRE showed how predictable behavior enables modern side-channel attacks.

Based on my experience with using diversity to counter timing-based side-channel attacks, I will present new ideas and results of either mitigating or substantially lessening the impact of these side-channel attacks.

5.7 Preserving High-Level Invariants in the Presence of Low-Level Code

David Chisnall (University of Cambridge, GB)

Joint work of David Chisnall, Brooks Davis, Khilan Gudka, David Brazdil, Alexandre Joannouand Jonathan Woodruff, A. Theodore Markettos, J. Edward Maste, Robert Norton, Stacey Son, Michael Roe, Simon W. Moore, Peter G. Neumann, Ben Laurie, Robert N. M. Watson

Main reference David Chisnall, Brooks Davis, Khilan Gudka, David Brazdil, Alexandre Joannou, Jonathan Woodruff, A. Theodore Markettos, J. Edward Maste, Robert Norton, Stacey D. Son, Michael Roe, Simon W. Moore, Peter G. Neumann, Ben Laurie, Robert N. M. Watson: "CHERI JNI: Sinking the Java Security Model into the C", in Proc. of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2017, Xi'an, China, April 8-12, 2017, pp. 569–583, ACM, 2017.

URL http://dx.doi.org/10.1145/3037697.3037725

Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.DavidChisnall.Slides.pptx

Most complex programs contain a mixture of different languages, but the guarantees available in common implementations are those of the lowest-level language. A typical Java implementation includes well over a million lines of C/C++ code with no constraints on its abilities and the same is true for most other high-level languages.

In the CHERI JNI work presented at ASPLOS last year, we demonstrated one possible way of allowing untrusted native code (including unverified assembly code) to exist in the same process as Java code, with high performance and preserving all of the invariants on which the Java security model is built.

5.8 Teaching a Production Compiler That Integers Are Not Pointers

David Chisnall (University of Cambridge, GB)

Joint work of David Chisnall, Brooks Davis, Khilan Gudka, David Brazdil, Alexandre Joannouand Jonathan Woodruff, A. Theodore Markettos, J. Edward Maste, Robert Norton, Stacey Son, Michael Roe, Simon W. Moore, Peter G. Neumann, Ben Laurie, Robert N. M. Watson
Main reference David Chisnall: "No such thing as a general-purpose processor", Commun. ACM, Vol. 57(12), pp. 44–48, 2014.
URL http://dx.doi.org/10.1145/2677030
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.DavidChisnall1.Slides.pptx

Over the past six years, we have taught the clang front end for [Objective-]C/C++, the LLVM optimisation pipeline, and the MIPS back end, to understand that pointers are a distinct type from integers (though memory may contain either). With the CHERI extensions applied to MIPS, we are able to preserve the distinction between pointers and integers all of the way from a source language, which supports features such as untagged unions and untyped memory, all of the way through the compilation pipeline to hardware that can preserve this distinction at run time.

We support a single-provenance semantics for pointers and can discuss the changes required to the compiler and our design decisions for concrete choices allowed within the C/C++ abstract machine that maintain compatibility with large corpora of real-world code while preserving memory safety.

5.9 Virtual Instruction Set Computing with Secure Virtual Architecture

John Criswell (University of Rochester, GB)

This talk will present Secure Virtual Architecture (SVA): a virtual instruction set computing infrastructure which we have used to enforce security policies on both application and operating system kernel code. I will present how we have used SVA to enforce traditional policies like memory safety and control flow integrity as well as newer policies that mitigate side-channel attacks and Spectre/Meltdown attacks launched by compromised operating system kernels. I hope to solicit feedback on how to employ secure compilation techniques into SVA to further reduce its (already small) trusted computing base size and to discuss the use of secure compilation techniques on operating system kernel code.

5.10 Capability Machines as a Target for Secure Compilation

Dominique Devriese (KU Leuven, BE)

Joint work of Dominique Devriese, Thomas Van Strydonck, Frank Piessens, Lau Skorstengaard, Lars Birkedal, Akram El-Korashy, Stelios Tsampas, Marco Patrignani, Deepak Garg
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.DominiqueDevriese.Slides.pdf

A quick introduction to capability machines, and an overview of ideas about how different properties can be enforced using different extensions of capability machines

5.11 Defining Undefined Behavior in Rust

Derek Dreyer (MPI-SWS – Saarbrücken, DE)

Joint work of Derek Dreyer, Ralf Jung
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.DerekDreyer.Slides.pdf

In the RustBelt project, we have been building foundations for understanding the safety claims of the Rust programming language and for evolving the language safely. In so doing, we have thus far assumed a memory model in which the only forms of undefined behavior are data races and memory safety violations. However, this is too simplistic. The Rust developers would like to support more aggressive compiler optimizations that exploit non-aliasing assumptions derived from Rust's reference types, but in order for such optimizations to be sound, undefined behavior must be expanded to include unsafe code that violates such non-aliasing assumptions. In this talk, I will report on several avenues currently being explored for defining undefined behavior in Rust.

5.12 Compiling a Cecure Variant of C to Capabilities

Akram El-Korashy (MPI-SWS – Saarbrücken, DE)

Joint work of Akram El-Korashy, Dominique Devriese, Deepak Garg, Marco Patrignani, Frank Piessens, Stelios Tsampas
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.AkramEl-Korashy.Slides.pdf

Capability machines offer architectural support for fine-grained memory separation and controlled sharing. In this in-progress work, we leverage this support to compile a high-level data isolation primitive fully abstractly. We start from a safe subset of C extended with an abstraction for modules that may have private state. The language semantics prevent a module from accessing an element of another module's private state, unless it has been shared explicitly. We then describe a compiler from this language to CHERI, a modern capability machine. In ongoing work, we are proving that the compiler is fully abstract, i.e., it preserves and reflects observational equivalence and, hence, implements the source module abstraction securely.

5.13 Building Secure SGX Enclaves using F*, C/C++ and X64

Cédric Fournet (Microsoft Research – Cambridge, GB)

Joint work of Cédric Fournet, Anitha Gollamudi

Intel SGX offers hardware mechanisms to isolate code and data running within enclaves from the rest of the platform. This enables security verification on a relatively small software TCB, but the task still involves complex low-level code.

Relying on the Everest verification toolchain, we use F* for developing specifications, code, and proofs; and then safely compile F* code to standalone C code. However, this

does not account for all code running within the enclave, which also includes trusted C and assembly code for bootstrapping and for core libraries. Besides, we cannot expect all enclave applications to be rewritten in F*, so we also compile legacy C++ defensively, using variants of /guard that dynamically enforce their safety at runtime.

To reason about enclave security, we thus compose different sorts of code and verification styles, from fine-grained statically-verified F* to dynamically-monitored C++ and custom SGX instructions.

This involves two related program semantics: most of the verification is conducted within F* using the target semantics of Kremlin—a fragment of C with a structured memory—whereas SGX features and dynamic checks embedded by defensive C++ compilers require lower-level X64 code, for which we use the verified assembly language for Everest (VALE) and its embedding in F*.

5.14 How to Define Secure Compilation? (A Property-Centric View)

Deepak Garg (MPI-SWS – Saarbrücken, DE)

Joint work of Deepak Garg, Carmine Abate, Roberto Blanco, Catalin Hritcu, Marco Patrignani, Jérémy Thibault
Main reference Carmine Abate, Roberto Blanco, Deepak Garg, Catalin Hritcu, Marco Patrignani, Jérémy Thibault: "Exploring Robust Property Preservation for Secure Compilation", CoRR, Vol. abs/1807.04603, 2018.
URL http://arxiv.org/abs/1807.04603
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.DeepakGarg.Slides.pdf

This talk presents a possible approach to defining compiler security as the preservation of security properties despite adversarial contexts. The talk starts from the idea that compiler correctness can be defined as preservation of properties (in the absence of adversaries). Adversarial contexts are then introduced, and a notion of compiler security, parametrized by a class of security properties, is defined. Particularly interesting classes include safety properties, hyperproperties (e.g., non-interference), and relational hyperproperties (e.g., observational equivalence).

5.15 When Good Components Go Bad: Formally Secure Compilation Despite Dynamic Compromise

Catalin Hritcu (INRIA – Paris, FR)

Joint work of Catalin Hritcu, Carmine Abate, Arthur Azevedo de Amorim, Roberto Blanco, Ana Nora Evans, Guglielmo Fachini, Théo Laurent, Benjamin C. Pierce, Marco Stronati, Andrew Tolmach
Main reference Guglielmo Fachini, Catalin Hritcu, Marco Stronati, Arthur Azevedo de Amorim, Ana Nora Evans, Carmine Abate, Roberto Blanco, Théo Laurent, Benjamin C. Pierce, Andrew Tolmach: "When Good Components Go Bad: Formally Secure Compilation Despite Dynamic Compromise", CoRR, Vol. abs/1802.00588, 2018.
URL http://arxiv.org/abs/1802.00588
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.CatalinHritcu1.Slides.pdf

We propose a new formal criterion for evaluating secure compilation schemes for unsafe languages, expressing end-to-end security guarantees for software components that may

become compromised after encountering undefined behavior–for example, by accessing an array out of bounds.

Our criterion is the first to model dynamic compromise in a system of mutually distrustful components with clearly specified privileges. It articulates how each component should be protected from all the others–in particular, from components that have encountered undefined behavior and become compromised. Each component receives secure compilation guarantees–in particular, its internal invariants are protected from compromised components–up to the point when this component itself becomes compromised, after which we assume an attacker can take complete control and use this component's privileges to attack other components. More precisely, a secure compilation chain must ensure that a dynamically compromised component cannot break the safety properties of the system at the target level any more than an arbitrary attacker-controlled component (with the same interface and privileges, but without undefined behaviors) already could at the source level.

To illustrate the model, we construct a secure compilation chain for a small unsafe language with buffers, procedures, and components, targeting a simple abstract machine with built-in compartmentalization. We give a careful proof (mostly machine-checked in Coq) that this compiler satisfies our secure compilation criterion. Finally, we show that the protection guarantees offered by the compartmentalized abstract machine can be achieved at the machine-code level using either software fault isolation or a tag-based reference monitor.

5.16 Taming Undefined Behavior in LLVM

Chung-Kil Hur (Seoul National University, KR)

Joint work of Chung-Kil Hur, Juneyoung Lee, Yoonseung Kim, Youngju Song, Sanjoy Das, John Regehr, David Majnemer, Nuno P. Lopes
Main reference Juneyoung Lee, Yoonseung Kim, Youngju Song, Chung-Kil Hur, Sanjoy Das, David Majnemer, John Regehr, Nuno P. Lopes: "Taming undefined behavior in LLVM", in Proc. of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017, Barcelona, Spain, June 18-23, 2017, pp. 633–647, ACM, 2017.
URL http://dx.doi.org/10.1145/3062341.3062343
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.Chung-KilHur.Slides.pdf

A central concern for an optimizing compiler is the design of its intermediate representation (IR) for code. The IR should make it easy to perform transformations, and should also afford efficient and precise static analysis.

In this paper we study an aspect of IR design that has received little attention: the role of undefined behavior. The IR for every optimizing compiler we have looked at, including GCC, LLVM, Intel's, and Microsoft's, supports one or more forms of undefined behavior (UB), not only to reflect the semantics of UB-heavy programming languages such as C and C++, but also to model inherently unsafe low-level operations such as memory stores and to avoid over-constraining IR semantics to the point that desirable transformations become illegal. The current semantics of LLVM's IR fails to justify some cases of loop unswitching, global value numbering, and other important "textbook" optimizations, causing long-standing bugs.

We present solutions to the problems we have identified in LLVM's IR and show that most optimizations currently in LLVM remain sound, and that some desirable new transformations become permissible. Our solutions do not degrade compile time or performance of generated code.

5.17 Taming I/O in Intermittent Computing

Limin Jia (Carnegie Mellon University – Pittsburgh, US)

Joint work of Limin Jia, Brandon Lucia, Milijana Surbatovich
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.LiminJia.Slides.pdf

Energy harvesting enables novel devices and applications without batteries. However, intermittent operation under energy harvesting poses new challenges to preserving program semantics under power failures. I will first discuss unique challenges that existing checkpointing mechanisms for intermittent computing face in the presence of I/O operations. Then, I will talk about our ongoing work on developing a static analysis tool for automatically identifying bugs caused by I/O operations, methods for fixing such bugs, and formal models for intermittent computing.

5.18 Data Refinement for Cogent

Gabriele Keller (The University of New South Wales – Sydney, AU)

Joint work of Gabriele Keller, Christine Rizkallah

COGENT allows low-level operating system components to be modelled as pure mathematical functions operating on algebraic data types, suitable for verification in an interactive theorem prover. Further-more, it can compile these models into imperative C programs, and provide a proof that this compilation is a refinement of the functional model. Currently, however, there is still a gap between the C data structures used in the operating system, and the algebraic data types used by COGENT, which force the programmer to write a large amount of boilerplate marshalling code to connect the two.

In this talk, I'll outline our current work on adding a data description component to the framework, which will allow COGENT to be flexible in how it represents its algebraic data types, enabling models that operate on standard algebraic data types to be compiled into C programs that manipulate C data structures directly. Once fully realised, this extension will enable more code to be automatically verified by COGENT, smoother interoperability with C, and substantially improved performance of the generated code.

5.19 Secure Compilation of Side-Channel Countermeasures: The Case of Cryptographic "Constant-Time"

Vincent Laporte (IMDEA Software Institute – Madrid, ES)

Joint work of Vincent Laporte, Gilles Barthe, Benjamin Grégoire
Main reference Gilles Barthe, Benjamin Grégoire, Vincent Laporte: "Provably secure compilation of side-channel countermeasures", IACR Cryptology ePrint Archive, Vol. 2017, p. 1233, 2017.
URL http://eprint.iacr.org/2017/1233
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.VincentLaporte.Slides.pdf

Software-based countermeasures provide effective mitigation against side-channel attacks, often with minimal efficiency and deployment overheads. Their effectiveness is often amenable to rigorous analysis: specifically, several popular countermeasures can be formalized as information flow policies, and correct implementation of the countermeasures can be verified with state-of-the-art analysis and verification techniques. However, in absence of further justification, the guarantees only hold for the language (source, target, or intermediate representation) on which the analysis is performed.

We consider the problem of preserving side-channel countermeasures by compilation for cryptographic "constant-time," a popular countermeasure against cache-based timing attacks. We present a general method, based on the notion of constant-time-simulation, for proving that a compilation pass preserves the constant-time countermeasure. Using the Coq proof assistant, we verify the correctness of our method and of several representative instantiations.

5.20 The Formal Verification of Compilers and What It Doesn't Say About Security

Xavier Leroy (INRIA – Paris, FR)

Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.XavierLeroy.Slides1.pdf

This talk starts with an overview of the formal verification of compilers, as done in the CompCert and CakeML projects for example.

Verifying the soundness of a compiler means proving that the generated code behaves as prescribed by the semantics of the source program. There are many definitions of interest for "behaves as prescribed." Observational equivalence is appropriate for well-defined source languages such as Java. However, for C and C++, observational equivalence cannot be guaranteed because several evaluation orders are allowed for source programs, while the compiled code implements one of those evaluation orders. Moreover, C and C++ treat run-time errors such as integer division by zero or out-of-bound array accesses as undefined behaviors, meaning that the compiled code is allowed to perform any actions whatsoever, from aborting the program to continuing with random values to opening a security hole.

The CompCert compiler verification project builds on a notion of program refinement that enables the compiler to choose one among several possible evaluation orders, making the program "more deterministic," and also to optimize source-level undefined behaviors away, making the program "more defined." An example of the latter dimension of refinement is the elimination of an integer division `z = x / y` if `z` is unused later: if `y` is 0, the original program exhibits undefined behavior (division by zero), but not the optimized program.

As discussed in the second part of the talk, CompCert-style compiler verification shows the preservation of safety and liveness properties of the source code, but fails to establish the preservation of many security properties of interest. This is illustrated on two examples: constant-time code and unwanted optimizations.

Example 1. Cryptographic code is said to be "constant time" if secret data is never used as argument to conditional branches, memory addressing, or other operations whose execution time depends on the value of the arguments. This "constant time" property does not rule out all side-channel attacks, but avoids the most obvious timing attacks. But is the property preserved by compilation? If the source code is "constant time," is the compiled code "constant time" too? Compilers can destroy the property by introducing conditionals or memory lookups for optimization purposes. This does not invalidate a CompCert-style semantic preservation proof. To reason about constant-time preservation it seems necessary to add observable events for non-constant-time operations to the semantic trace, and reason about the preservation, or removal but not insertion, of such events during compilation.

Example 2. C compilers are allowed to optimize based on the assumption that the source code does not run into undefined behavior. Sometimes, this leads optimizers to amplify a programming error, removing security-relevant checks that follow a possibly-undefined operation. CVE 2009-1879 is an example of such a compiler-amplified security hole. Such misguided optimizations are hard to control because they are close to other desirable optimizations, and both fall out naturally from standard compiler passes such as value analysis and constant propagation. CompCert tries hard to degrade the precision of its value analysis to be conservative with respect to undefined behavior. However, this is a best effort and no formal proof is given that the analysis was degraded enough so that subsequent optimizations preserve security checks.

In conclusion, formal compiler verification in the style of CompCert or CakeML gives many guarantees relevant to safety, but few guarantees relevant to security beyond safety. CompCert tries to handle security code with care, but it's a best effort without confirmation by the proof. More work is needed to semantically characterize the security properties of interest and prove their preservation by compilation.

5.21 Verified Compilation of Noninterference for Shared-Memory Concurrent Programs

Toby Murray (The University of Melbourne, AU)

Joint work of Toby Murray, Robert Sison, Edward Pierzchalski, Christine Rizkallah
Main reference Toby C. Murray, Robert Sison, Edward Pierzchalski, Christine Rizkallah: "Compositional Verification and Refinement of Concurrent Value-Dependent Noninterference", in Proc. of the IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 – July 1, 2016, pp. 417–431, IEEE Computer Society, 2016.
URL http://dx.doi.org/10.1109/CSF.2016.36
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.TobyMurray.Slides.pdf

Shared-memory concurrency is ubiquitous in modern programming, including in security-critical embedded devices. Proofs of information flow control (IFC) for the software that controls such devices have recently become a reality. Yet most of this work to date operates at the level of the small-step semantics for the source programming language. In reality, such programs execute atop a thread scheduler (e.g., the OS kernel), executing binary instructions in fixed slices. We argue that verified noninterference-preserving compilation should be

employed to bridge this semantic gap, and present a theory for compositionally proving preservation of timing-sensitive noninterference for concurrent programs under refinement. We explain how this theory captures the semantics of compiled programs executing under an instruction-based scheduling discipline, and its instantiation in a verified compiler from a simple While language to an idealised RISC language. We report on the current state of this work, which is part of the COVERN project (https://covern.org), and directions for future research.

5.22 Is the Verified CakeML Compiler Secure?

Magnus Myreen (Chalmers University of Technology – Gothenburg, SE)

Joint work of Scott Owens, Yong Kiam Tan, Anthony Fox, Ramana Kumar, Michael Norrish
URL https://cakeml.org/
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.MagnusMyreen.Slides.pdf

I propose to (1) present the CakeML compiler at a high-level, then (2) zoom in on the exact details of the compiler correctness theorem, but leave plenty of time for (3) a discussion on whether the CakeML compiler is secure or not. The CakeML compiler starts from a safe language (unsafe out-of-bounds accesses are not possible) and compiles it to concrete machine code (x86, ARM, RISC-V etc.) with a semantics where the OS and other programs are allowed to interrupt the CakeML machine code. The CakeML compiler is probably safer than unverified compilers for ML, but is it more secure? In the discussion part of my talk, I'll talk about different attacker models and security questions regarding the target semantics which is at the level of machine code.

5.23 Compiler Optimizations with Retrofitting Transformations: Is There a Semantic Mismatch?

Santosh Nagarakatte (Rutgers University – New Brunswick, US)

Joint work of Santosh Nagarakatte, Jay P. Lim, Vinod Ganapathy
Main reference Jay P. Lim, Vinod Ganapathy, Santosh Nagarakatte: "Compiler Optimizations with Retrofitting Transformations: Is there a Semantic Mismatch?", in Proc. of the 2017 Workshop on Programming Languages and Analysis for Security, PLAS@CCS 2017, Dallas, TX, USA, October 30, 2017, pp. 37–42, ACM, 2017.
URL http://dx.doi.org/10.1145/3139337.3139343
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.SantoshNagarakatte.Slides.pdf

A retrofitting transformation modifies an input program by adding instrumentation to monitor security properties at runtime. These tools often transform the input program in complex ways. Compiler optimizations can erroneously remove the instrumentation added by a retrofitting transformation in the presence of semantic mismatches between the assumptions of retrofitting transformations and compiler optimizations. This talk will describe a generic strategy to ascertain that every event of interest that is checked in the retrofitted program is also checked after optimizations.

5.24 Plugging Information Leaks Introduced by Compiler Transformations

Kedar Namjoshi (Nokia Bell Labs – Murray Hill, US-NJ)

Joint work of Kedar Namjoshi, Chaoqiang Deng
Main reference Chaoqiang Deng, Kedar S. Namjoshi: "Securing a Compiler Transformation", in Proc. of the Static Analysis – 23rd International Symposium, SAS 2016, Edinburgh, UK, September 8-10, 2016, Proceedings, Lecture Notes in Computer Science, Vol. 9837, pp. 170–188, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-662-53413-7_9
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.KedarNamjoshi.Slides.pdf

Some compiler optimizations (e.g., dead store removal, or SSA conversion) can introduce new information leaks as they transform a program. I will talk about sound–but necessarily approximate–methods to produce leak-free forms of these optimizations. Not all optimizations introduce leaks; I will show how one can verify that an implementation of a transformation is leak-free by checking additional properties of a refinement relation (a "witness") that is produced originally to justify correctness.

There are several open questions (e.g., how to establish preservation of security properties other than information leakage?) which I hope to have the chance to discuss during the talk and in the seminar.

5.25 Relational Logic for Fine-grained Security Policy and Translation Validation

David A. Naumann (Stevens Institute of Technology – Hoboken, US)

Joint work of David A. Naumann, Anindya Banerjee, Mohammed Nikouei
Main reference Anindya Banerjee, David A. Naumann, Mohammad Nikouei: "Relational Logic with Framing and Hypotheses: Technical Report", CoRR, Vol. abs/1611.08992, 2016.
URL http://arxiv.org/abs/1611.08992
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.DavidA.Naumann.Slides.pdf

Relational Hoare Logics facilitate reasoning about information-flow properties of programs as well as relations between programs such as observational equivalence. Such logics might be used to specify sensitive information at source level and to specify what is considered observable at source and target levels, in order to define security-preserving compilation and support translation validation.

5.26 Specifications for Dynamic Enforcement of Relational Program Properties

Max S. New (Northeastern University – Boston, US)

Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.MaxS.New.Slides.pdf

Many security and reliability properties are phrased in terms of relations on programs, e.g., noninterference and representation independence. While all source-level programs respect

these relational properties due to syntactic restrictions such as linearity or type checking, when compiling securely to low-level programs, we need to interpose on the boundary between compiled code and low-level attackers to maintain our high-level security properties.

In this talk we present a simple specification for the interposition functions between compiled code and low-level attackers. The basic idea is to first provide a *refinement relation* between high level and low level behaviors. Some simple properties must be satisfied to ensure that the refinement relation is compatible with the relational properties of interest. Then functions that enforce high-level interfaces on low-level attackers and dually protect compiled code from low-level attackers can be given two dual specifications with respect to the refinement relation. An enforcement function is sound if its output refines its input, and *optimal* if it has the most behavior of any refinement of the input. Dually, a protection function is sound if its output is refined by its input, and *optimal* if it has the least behavior of any refinement of the input. Finally, to get security/full abstraction we need the protection function to be *injective*, which is here equivalent to saying that `enforce o protect = id`.

This fairly simple spec is the core of "Galois connection"-based approaches to security, but we argue that by focusing on the refinement relation first, the Galois connection properties become more intuitive. Furthermore, since the actual implementation of enforce and protect can be quite complex, it is useful to specify them first in terms of a simple refinement relation.

5.27 Closure Conversion is Safe-for-Space

Zoe Paraskevopoulou (Princeton University, US)

Joint work of Zoe Paraskevopoulou, Andrew Appel

Compiler transformations may fail to preserve the resource consumption of compiled programs. A notable example is closure conversion with linked closures which may introduce space leaks. In this talk I will present a (currently ongoing) proof that closure conversion with flat closure representation is safe-for-space, meaning that it preserves the space complexity of the compiled program. We develop a method based on step-indexed logical relations that allows us to conveniently reason about the resource consumption of the source and target programs, as well as the functional correctness of the transformation.

5.28 Linking Types: Bringing Fully Abstract Compilers and Flexible Linking Together

Daniel Patterson (Northeastern University – Boston, US)

Joint work of Daniel Patterson, Amal Ahmed
Main reference Daniel Patterson, Amal Ahmed: "Linking Types for Multi-Language Software: Have Your Cake and Eat It Too", in Proc. of the 2nd Summit on Advances in Programming Languages, SNAPL 2017, May 7-10, 2017, Asilomar, CA, USA, LIPIcs, Vol. 71, pp. 12:1–12:15, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.SNAPL.2017.12
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.DanielPatterson.Slides.pdf

Fully abstract compilers protect components from target-level attackers by ensuring that any observations or influence that a target attacker could have can also be done by a source-level attacker. This means that programmers need only reason about security properties in their own language, not additional interactions that may happen in lower level intermediate or target languages. While this is obviously an extremely valuable property for secure compilers, it rules out linking with target code that has features or restrictions that can not be represented in the source language that is being compiled.

While traditionally fully abstract compilation and flexible linking have been thought to be at odds, I'll present a novel idea called Linking Types that allows them to coexist. Linking Types enable a programmer to opt in to local violations of full abstraction that she needs in order to link with particular code without giving up the property globally. This fine-grained mechanism enables flexible interoperation with low-level features while preserving the high-level reasoning principles that fully abstract compilation offers.

The talk will give some brief background to the ideas, show how they play out in examples, and open a broader discussion as to how this idea could influence secure compilers and language design.

5.29 A Project on Secure Compilation in the Context of the Internet of Things

Tamara Rezk (INRIA – Sophia Antipolis, FR)

Joint work of Tamara Rezk, Frédéric Besson, Thomas Jensen, Alan Schmitt, Gérard Berry, Nataliia Bielova, Ilaria Castellani, Manuel Serrano, Claude Castelluccia, Daniel Le Métayer
URL http://cisc.gforge.inria.fr/

I will briefly present a new starting project which relies on the idea of using secure compilation for the Internet of Things (IoT). The talk will present new challenges in the IoT context, security risks, and speculations on how to address them.

5.30 A Formal Equational Theory for Call-By-Push-Value

Christine Rizkallah (The University of New South Wales – Sydney, AU)

Joint work of Christine Rizkallah, Dmitri Garbuzov, Steve Zdancewic
URL https://github.com/secure-compilation/ds-2018/raw/master/18201.ChristineRizkallah.Preprint.pdf
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.ChristineRizkallah.Slides1.pdf

Establishing that two programs are contextually equivalent is hard, yet essential for reasoning about semantics preserving program transformations such as compiler optimizations. The Vellvm project aims to use Coq to formalize and reason about LLVM program transformations and as part of this project we are using a variant of Levy's call-by-push-value language. I will talk about how we establish the soundness of an equational theory for call-by-push-value and about how we used our equational theory to significantly simplify the verification of classic optimizations.

5.31 Secure Compilation–Understanding the Endpoints

Peter Sewell (University of Cambridge, GB)

Joint work of many people
URL http://www.cl.cam.ac.uk/users/pes20/rems
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.PeterSewell.Slides.pdf

In this talk I described ongoing work in the REMS and CHERI projects to define the architecture and C-language abstractions, both for current mainstream architectures (especially ARMv8-A and RISC-V, with some work also for IBM POWER and x86) and mainstream ISO / de facto C, and for the research CHERI architecture and CHERI C language. I also described work on WebAssembly semantics.

5.32 Constant-Time Crypto Programming with FaCT

Deian Stefan (University of California, San Diego, US)

Joint work of Deian Stefan, Fraser Brown, Sunjay Cauligi, Ranjit Jhala, Brian Johannesmeyer, John Renner, Gary Soeller, Riad Wahby, Conrad Watt
Main reference Sunjay Cauligi, Gary Soeller, Fraser Brown, Brian Johannesmeyer, Yunlu Huang, Ranjit Jhala, Deian Stefan: "FaCT: A Flexible, Constant-Time Programming Language", in Proc. of the IEEE Cybersecurity Development, SecDev 2017, Cambridge, MA, USA, September 24-26, 2017, pp. 69–76, IEEE, 2017.
URL http://dx.doi.org/10.1109/SecDev.2017.24
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.DeianStefan.Slides.pdf

Implementing cryptographic algorithms that do not inadvertently leak secret information is notoriously difficult. Today's general-purpose programming languages and compilers do not account for data sensitivity; consequently, most real-world crypto code is written in a subset of C intended to predictably run in constant time. This C subset, however, forgoes structured programming as we know it–crypto developers, today, do not have the luxury of if-statements, efficient looping constructs, or procedural abstractions when handling sensitive

data. Unsurprisingly, even high-profile libraries, such as OpenSSL, have repeatedly suffered from bugs in such code.

In this talk, I will describe FaCT, a new domain-specific language that addresses the challenge of writing constant-time crypto code. With FaCT, developers write crypto code using standard, high-level language constructs; FaCT, in turn, compiles such high-level code into constant-time assembly. FaCT is not a standalone language. Rather, we designed FaCT to be embedded into existing, large projects and language. In this talk, I will describe how we integrated FaCT in several such projects (OpenSSL, libsodium, and mbedtls) and languages (C, Python, and Haskell).

5.33 C-Level Tag-Based Security Monitoring

Andrew Tolmach (Portland State University, US)

Joint work of Andrew Tolmach, Sean Anderson, Catalin Hritcu, Benjamin C. Pierce
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.AndrewTolmach.Slides.pdf

Recent work on security "micropolicies" uses hardware-level metadata tags to monitor individual machine operations. This talk will sketch preliminary ideas for how to raise the definition of tag-based policies to the level of C code. C-level polices should be useful both to express high-level properties that are tedious or impossible to specify at machine level (e.g., information flow control or compartmentalization) and to enforce particular variants of C semantics (e.g., differing flavors of memory safety based on differing pointer aliasing rules). C-level policies can be (verifiably) compiled to machine-level policies to be enforced by existing (prototype) hardware.

5.34 Verifying the Glasgow Haskell Compiler Core Language

Stephanie Weirich (University of Pennsylvania – Philadelphia, US)

Joint work of Stephanie Weirich, Joachim Breitner, Antal Spector-Zabusky, Yao Li, Christine Rizkallah, John Wiegley
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.StephanieWeirich.Slides.pdf

Verified compilers are one part of secure compilation. By developing a compiler within the language of a proof assistant, we can rigorously show that the semantics of the source language is preserved through compilation to the target. However, what about our existing compilers?

In this talk, I will present our preliminary work that uses the Coq theorem prover to reason about the implementation of the GHC Core intermediate language. Our goal is to show that Core optimization passes are correct: i.e., that these transformations preserve the invariants of the compiler AST and, ultimately, the semantics of the Core language. Our work uses the `hs-to-coq` tool to translate the source code of GHC from Haskell into Gallina, the language of the Coq proof assistant, taking advantage of the similarity between the languages. One discussion point is how much our proofs actually apply to GHC–what can we really prove about compilation and what guarantees can we conclude from our work?

5.35 Vellvm: Verifying the LLVM

Steve Zdancewic (University of Pennsylvania – Philadelphia, US)

Joint work of Steve Zdancewic, Dmitri Garbuzov, William Mansky, Christine Rizkallah, Yannick Zakowski
Slides https://github.com/secure-compilation/ds-2018/raw/master/18201.SteveZdancewic.Slides.pdf

I will describe the Vellvm project, which seeks to provide a formal framework for developing machine-checkable proofs about LLVM IR programs and translation passes. I'll highlight some of the "good," the "bad" and the "ugly" things about our prior LLVM developments, which motivates our ongoing work to re-engineer the Coq formalization.

In the Vellvm (Verified LLVM) project, we have been experimenting with representing SSA control-flow-graphs using terms of Levy's call-by-push-value (CBPV) variant of the lambda calculus. CBPV offers the benefits of a good equational theory based on the usual notions of beta-equivalence. By relating the operational semantics of the CBPV language to that of the SSA-control-flow graphs, we can transport reasoning and program transformations from one level to another, thereby allowing for very simple proofs of the correctness of many low-level optimizations such as function inlining.

This talk will explain our on-going work in this area and conections to the LLVM IR.

Participants

- Amal Ahmed
Northeastern University – Boston, US
- Gilles Barthe
IMDEA Software – Madrid, ES
- Nick Benton
Facebook – London, GB
- Frédéric Besson
INRIA – Rennes, FR
- Pramod Bhatotia
University of Edinburgh, GB
- Lars Birkedal
Aarhus University, DK
- Roberto Blanco
INRIA – Paris, FR
- William Bowman
Northeastern University – Boston, US
- Stefan Brunthaler
Universität der Bundeswehr – München, DE
- David Chisnall
University of Cambridge, GB
- John T. Criswell
University of Rochester, US
- Dominique Devriese
KU Leuven, BE
- Derek Dreyer
MPI-SWS – Saarbrücken, DE
- Akram El-Korashy
MPI-SWS – Saarbrücken, DE
- Cédric Fournet
Microsoft Research UK – Cambridge, GB
- Deepak Garg
MPI-SWS – Saarbrücken, DE
- Chris Hawblitzel
Microsoft Research – Redmond, US
- Catalin Hritcu
INRIA – Paris, FR
- Chung-Kil Hur
Seoul National University, KR
- Limin Jia
Carnegie Mellon University – Pittsburgh, US
- Gabriele Keller
UNSW – Sydney, AU
- Vincent Laporte
IMDEA Software – Madrid, ES
- Xavier Leroy
INRIA – Paris, FR
- Toby Murray
The University of Melbourne, AU
- Magnus Myreen
Chalmers University of Technology – Göteborg, SE
- Santosh Nagarakatte
Rutgers University – Piscataway, US
- Kedar Namjoshi
Nokia Bell Labs – Murray Hill, US
- David A. Naumann
Stevens Institute of Technology – Hoboken, US
- Max S. New
Northeastern University – Boston, US
- Scott Owens
University of Kent – Canterbury, GB
- Zoe Paraskevopoulou
Princeton University, US
- Marco Patrignani
Universität des Saarlandes, DE
- Daniel Patterson
Northeastern University – Boston, US
- Frank Piessens
KU Leuven, BE
- Tamara Rezk
INRIA Sophia Antipolis, FR
- Christine Rizkallah
UNSW – Sydney, AU
- Peter Sewell
University of Cambridge, GB
- Deian Stefan
University of California – San Diego, US
- Andrew Tolmach
Portland State University, US
- Stelios Tsampas
KU Leuven, BE
- Neline van Ginkel
KU Leuven, BE
- Stephanie Weirich
University of Pennsylvania – Philadelphia, US
- Steve Zdancewic
University of Pennsylvania – Philadelphia, US

Report from Dagstuhl Seminar 18202

Inter-Vehicular Communication Towards Cooperative Driving

Edited by

Onur Altintas[1], Suman Banerjee[2], Falko Dressler[3], and Geert Heijenk[4]

1 **TOYOTA InfoTechnology Center USA – Mountain V, US,** onur@us.toyota-itc.com
2 **University of Wisconsin – Madison, US,** suman@cs.wisc.edu
3 **Universität Paderborn, DE,** dressler@ccs-labs.org
4 **University of Twente, NL,** geert.heijenk@utwente.nl

Abstract

Looking back at the last decade, one can observe enormous progress in the domain of vehicular networking. In this growing community, many ongoing activities focus on the design of communication protocols to support safety applications, intelligent navigation, multi-player gaming and others. This seminar shifted the focus from basic networking principles to networked control applications. We were particularly interested in eSafety applications and traffic efficiency applications that are thought to yield substantial benefits for the emerging "cooperative automated driving" domain. The seminar brought together experts from several fields, including classical computer science (computer networking, simulation and modeling, operating system design), electrical engineering (digital signal processing, communication networks), and automated driving (mechanical engineering, image processing, control theory), to discuss the most challenging issues related to inter-vehicular communication and cooperative driving.

Seminar May 13–16, 2018 – http://www.dagstuhl.de/18202
2012 ACM Subject Classification Networks → Cyber-physical networks, Networks → Mobile networks, Networks → Network architectures, Networks → Network performance evaluation, Networks → Network protocols
Keywords and phrases automated driving, cooperative driving, road traffic safety, vehicular networking
Digital Object Identifier 10.4230/DagRep.8.5.31
Edited in cooperation with Isabel Wagner

1 Executive Summary

Falko Dressler (Universität Paderborn, DE)
Onur Altintas (TOYOTA InfoTechnology Center USA – Mountain V, US)
Suman Banerjee (University of Wisconsin – Madison, US)
Geert Heijenk (University of Twente, NL)
Katrin Sjoberg (Volvo, Sweden)

Looking back at the last decade, one can observe enormous progress in the domain of vehicular networking. In this growing community, many ongoing activities focus on the design of

Inter-Vehicular Communication Towards Cooperative Driving, *Dagstuhl Reports*, Vol. 8, Issue 05, pp. 31–59
Editors: Onur Altintas, Suman Banerjee, Falko Dressler, and Geert Heijenk
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

communication protocols to support safety applications, intelligent navigation, multi-player gaming and others. Very large projects have been initiated to validate the theoretic work in field tests and protocols are being standardized. With the increasing interest from industry, security and privacy have also become crucial aspects in the stage of protocol design in order to support a smooth and carefully planned roll-out. We are now entering an era that might change the game in road traffic management. This is supported by the U.S. federal government announcement in December 2016 that National Highway Traffic Safety Administration (NHTSA) plans to make V2V devices in new vehicles mandatory. This coincides with the final standardization of higher layer networking protocols in Europe by the ETSI.

The vehicular networking research also complements the ongoing activities towards automated driving. Very successful activities started with the Google and lead to first projects on the road such as the Singapore driverless taxi service or the platooning experiments in Scandinavia and now Germany.

The management and control of network connections among vehicles and between vehicles and an existing network infrastructure is currently one of the most challenging research fields in the networking domain. Using the terms Vehicular Ad-hoc Networks (VANETs), Inter-Vehicle Communication (IVC), Car-2-X (C2X), or Vehicle-2-X (V2X), many applications – as interesting as challenging – have been envisioned and (at least) partially realized. In this context, a very active research field has developed. There is a long list of desirable applications that can be grouped into four IVC categories:

1. eSafety applications that try to make driving safer, e.g. road hazard warning;
2. traffic efficiency applications aiming at more efficient and thus greener traffic, e.g., detection of traffic jams;
3. manufacturer oriented applications, e.g., automatic software updates; and
4. comfort applications, e.g. automatic map updates.

In 2010, a first Dagstuhl Seminar (10402) was organized on the topic of inter-vehicular communication. The motivation was to bring together experts in this field to investigate the state of the art and to highlight where sufficient solutions already existed. The main outcome of this very inspiring seminar was that there are indeed areas within this research where scientific findings are being consolidated and adopted by industry. This was the consensus of quite intriguing discussions among participants from both industry and academia. Yet, even more aspects have been identified where substantial research is still needed. These challenges have been summarized in the following IEEE Communications Magazine article [1].

A follow-up seminar (13392) was organized in 2013. The goal was to again bring together leading researchers both from academia and industry to discuss if and where the previously identified challenges have been adequately addressed, and to highlight where sufficient solutions exist today, where better alternatives need to be found, and also to give directions where to look for such alternatives. Furthermore, it was the goal of this workshop to go one step beyond and identify where IVC can contribute to the basic foundations of computer science or where previously unconsidered foundations can contribute to IVC. It turned out that quite a number of research questions were still open or insufficiently addressed. This particularly included scalability and real-time capabilities. These challenges have been summarized in the following IEEE Communications Magazine article [2].

We now shifted the focus of this seminar from basic networking principles to networked control applications. We were particularly interested in the first two IVC categories that are thought to yield substantial benefits for the emerging “cooperative automated driving” domain. It is of utmost importance to bring together expertise from classical computer

science (computer networking, simulation and modeling, operating system design), from electrical engineering (digital signal processing, communication networks), as well as from automated driving (mechanical engineering, image processing, control theory). Building upon the great success of the first two seminars, with this follow-up seminar, we aimed to again bring together experts from all these fields from both academia and industry.

The seminar focused intensively on discussions in several working groups. To kick-off these discussions, we invited two keynote talks "Cooperative Driving A Control of a Networking Problem?" by Renato Lo Cigno and "Cooperative driving – maneuvers, perception, and IVC" by Lars Wolf. These keynotes were complemented by four additional talks: Human-in-the-Loop: Towards Deeply Integrated Hybridized Systems (Falko Dressler), Machine Learning for Cooperative Driving (Geert Heijenk), Measuring Privacy in Vehicular Networks (Isabel Wagner), and Predictable V2X Networking for Application-Networking Co-Design (Hongwei Zhang). We finally organized the following working groups on some of the most challenging issues related to inter-vehicular communication and cooperative driving:

- Ultra-Reliable Low-Latency and Heterogeneous V2X Networking,
- Human-in-the-Loop,
- Safety-critical Vehicular Network Applications,
- Security and Privacy,
- Network and Cloud based Control, and
- Sensing and Data Management.

For most of these working groups, we provide in-depth feedback from the experts in this report.

References

1 Falko Dressler, Hannes Hartenstein, Onur Altintas, and Ozan K. Tonguz. Inter-Vehicle Communication – Quo Vadis. *IEEE Communications Magazine*, 52(6):170–177, June 2014.

2 Falko Dressler, Frank Kargl, Jörg Ott, Ozan K. Tonguz, and Lars Wischhof. Research Challenges in Inter-Vehicular Communication – Lessons of the 2010 Dagstuhl Seminar. *IEEE Communications Magazine*, 49(5):158–164, May 2011.

2 Table of Contents

3 Overview of Talks

3.1 Human-in-the-Loop: Towards Deeply Integrated Hybridized Systems

Falko Dressler (Universität Paderborn, DE)

Main reference Falko Dressler: "Cyber Physical Social Systems: Towards Deeply Integrated Hybridized Systems", in Proc. of the 2018 International Conference on Computing, Networking and Communications, ICNC 2018, Maui, HI, USA, March 5-8, 2018, pp. 420–424, IEEE Computer Society, 2018.
URL http://dx.doi.org/10.1109/ICCNC.2018.8390404

This talk is about issues raising up when considering not 100% optimal technical systems optimized for both individual behavior and global metrics but also considering the impact of the human-in-the-loop. Technically, we are observing a paradigm shift from classical Cyber Physical Systems (CPS) to Cyber Physical Social Systems (CPSS). Humans impact our technical systems, here we talk about (semi-)automated cooperative driving, in quite many dimensions. This includes the driving behavior that depends on the driver's demands or wishes, experiences, and capabilities that also vary over time. This is complemented by the often-citied incapability of humans to self-assess their abilities. A final frontier might be public acceptance on a global level, which might push or kill (optimal) technical solutions.

3.2 Machine Learning for Cooperative Driving

Geert Heijenk (University of Twente, NL)

Joint work of Geert Heijenk, Aashik Chandramohan

Machine learning is currently being introduced for self-driving cars. To a large extent the machine learning is used to interpret sensor information, including radar, lidar and video. In our research we are exploring to what extent machine learning can be used for vehicles to perform automated cooperative driving / maneuvering using information obtained through V2X communications. To this end, we need an environment for training systems and for testing systems. We therefore use a traffic simulation environment, SUMO, in which we can control the maneuvering of one, several, or all cars using machine learning agents.

In current experiments, we are using deep Q-learning to control the maneuvers of a vehicle on a 2-lane highway, where other cars are driven using one of the traditional SUMO driving models. The state input to the machine learning agent consists of the velocity and lane of the ego vehicle, and lane, and velocity of all the surrounding vehicles, assuming this is communicated using V2X. The actions the machine learning agent can take are lane change, acceleration and deceleration. The most critical part is the reward system. Currently, we give a strong negative reward for a collision. We also give negative rewards for near-collisions, and for violating traffic rules. Positive rewards are given depending on the speed, below the speed limit. In training periods of thousands of episodes, we can see the collision rate decreasing with the length of the training period, but not yet to an extent that we achieve reasonably low collision rates.

We plan to look into multi-agent learning as a way to improve the performance. Further, we are exploring other scenarios, such as intersection traffic. In that scenario, it is interesting

to see what happens if we extend the case where one car is driven by a machine learning agent and the others obey traffic rules, to the situation where all cars are driven by machine learning. From that situation maybe a new set of traffic rules will automatically emerge.

Overall, we are interested to see what is achievable using machine learning for cooperative driving, and what is the influence of all the design and parameter choices in machine learning on the learning outcome. Furthermore, we are interested to assess the potential of a simulated traffic environment for learning and for testing the outcome of the learning algorithms.

3.3 Cooperative Driving A Control of a Networking Problem?

Renato Lo Cigno (University of Trento, IT)

Joint work of Renato Lo Cigno, Michele Segata

This short talk wants to rise the attention on the fact that cooperative driving is a very complex, multi-disciplinary topic that requires a holistic approach to find a solution. All too often researchers from a specific discipline see cooperative driving shrinking the focus to their discipline, Control, Networking, Consensus, Automotive, ... loosing the big picture, and also doing modeling simplifications to tackle the problem that indeed introduce biases and errors, leading to partial, if not fully wrong, solutions.

It is clear that we are missing theoretical models that are able to grab the complexity of this system, and this aspect requires attention otherwise we risk doing research that is doomed to irrelevance, because solutions will find their way into life through other means.

Another topic of attention and interest is the lack of a sort of "standardization" at the coordination level. While networking and communications are used to have standards that define the minimum set of capabilities required to enable interaction and cooperation, it seems that at the consensus, control and coordination level there is nothing like this, so that we risk to have plenty of potential solutions from different automakers that, although they are formally compatible as they use the same networking and information exchange layer, they are not actually compatible, as they apply different logics and algorithms that lead to sub-optimal decisions, or even to contrasting decisions that may even lead to dangerous situations.

3.4 Measuring Privacy in Vehicular Networks

Isabel Wagner (De Montfort University – Leicester, GB)

Joint work of Isabel Wagner, Yuchen Zhao
Main reference Yuchen Zhao and Isabel Wagner: "On the Strength of Privacy Metrics for Vehicular Communication", IEEE Transactions on Mobile Computing, 2018.
URL https://doi.org/10.1109/TMC.2018.2830359

Vehicular communication plays a key role in near-future automotive transport, promising features such as increased traffic safety and wireless software updates. However, vehicular communication can expose drivers' locations and thus poses privacy risks. Many schemes have been proposed to protect privacy in vehicular communication, and their effectiveness is usually

evaluated with privacy metrics. However, different privacy metrics have not been compared to each other, and it is unknown how strong the metrics are. In this talk, I evaluate and compare the strength of 41 privacy metrics in terms of four novel criteria: Privacy metrics should be monotonic, i.e., indicate decreasing privacy for increasing adversary strength; their values should be spread evenly over a large value range to support within-scenario comparability; and they should share a large portion of their value range between traffic conditions to support between-scenario comparability. I evaluate all four criteria on real and synthetic traffic with state-of-the-art adversary models and create a ranking of privacy metrics. The results indicate that no single metric dominates across all criteria and traffic conditions. I therefore recommend to use metrics suites, i.e., combinations of privacy metrics, when evaluating new privacy-enhancing technologies.

3.5 Cooperative driving – maneuvers, perception, and IVC

Lars Wolf (TU Braunschweig, DE)

Joint work of Lars Wolf, Hendrik-Jörn Günther, Bernd Lehmann

Inter-Vehicular communication (IVC) can enable manifold types of cooperation between traffic participants, including human-driven vehicles, future autonomous vehicles, and also others like pedestrians and bicyclists. In daily live, humans cooperate and help each other in various ways, sometimes due to altruistic reasons or hoping for (indirect) reciprocity. This leads to many questions like: Can vehicular networks support such cooperation? What are the requirements for that and which new techniques are needed? Are specific methods for trust and reputation necessary? Vehicles may help others by cooperative sensing – how can such an architecture look like. Do autonomous vehicles, where no human assesses data, lead to additional demands?

Cooperative driving needs information about (i) the current situation consisting of the own perception / sensing as well as of collective perception / sensing (ii) intention of others, i.e., currently planned trajectories as well as potentially desired trajectories.

For (i) collective perception, several questions have to be solved such as: Which observations should be transmitted? At which granularity / which detail level (sensor data, objects, ...)? How and how often should transmission take place? How much does it improve the awareness ratio? How about reliability, trustworthiness, ...?

Information about (ii) intention of others enables maneuver coordination and, hence, extended cooperation. A general framework, supporting different kinds of scenarios, should be provided; thus, not for a specific traffic situation only. This requires the exchange of behavior composed of two components: a) the currently planned trajectory and b) a desired trajectory, representing a favored trajectory of a vehicle in case the need to deviate from the currently planned trajectory is detected. While maneuver coordination can be very helpful, it opens up many questions, e.g., regarding potential ambiguities, maneuver cascading and oscillation, complexity, and reliability. And there various IVC research concerns which need further study, e.g.: How to enable Maneuver Coordination Message exchange? What are the communication requirements ? Which communication technologies should be used? How to deal with the interrelation between coordination necessity for increasing traffic density?

3.6 Predictable V2X Networking for Application-Networking Co-Design

Hongwei Zhang (Iowa State University, US)

V2X communication is a basic enabler of the Connected-and-Automated-Vehicle (CAV) vision. In supporting safety-critical applications yet subject to complex dynamics and uncertainties, it is important to ensure predictable V2X communication (e.g., in reliability, timeliness, and throughput) so that predictable and trustworthy CAV systems can be developed. In this talk, I will present an integrated architecture for CAV applications and networks, and I will present field-deployable approaches to ensuring predictable communication reliability, timeliness, and throughput in highly-dynamic V2X networks. I will also present the applications of our architecture and algorithms to networked AR and networked control for CAVs.

References

1 Chuan Li, Hongwei Zhang, Jayanthi Rao, Le Yi Wang, George Yin, Cyber-Physical Scheduling for Predictable Reliability of Inter-Vehicle Communications, short paper, ACM/IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI), 2018
2 Yu Chen, Hongwei Zhang, Nathan Fisher, Le Yi Wang, George Yin, Probabilistic Per-Packet Real-Time Guarantees for Wireless Networked Sensing and Control, IEEE Transactions on Industrial Informatics, 14(5):2133–2145, 2018
3 Hongwei Zhang, Xiaohui Liu, Chuan Li, Yu Chen, Xin Che, Feng Lin, Le Yi Wang, George Yin, Scheduling with Predictable Link Reliability for Wireless Networked Control, IEEE Transactions on Wireless Communications (TWC), 16(9):6135–6150, 2017
4 Hongwei Zhang, Xin Che, Xiaohui Liu, Xi Ju, Adaptive Instantiation of the Protocol Interference Model in Wireless Networked Sensing and Control, ACM Transactions on Sensor Networks (TOSN), 10(2), January 2014

4 Working groups

4.1 Ultra-Reliable Low-Latency (URLL) and Heterogeneous V2X Networking

Onur Altintas (TOYOTA InfoTechnology Center USA – Mountain V, US), Ali Balador (RISE SICS – Västerås, SE), Suman Banerjee (University of Wisconsin – Madison, US), Claudia Campolo (University Mediterranea of Reggio Calabria, IT), Sinem Coleri Ergen (Koc University – Istanbul, TR), Eylem Ekici (Ohio State University – Columbus, US), Sonia Heemstra de Groot (TU Eindhoven, NL), Thorsten Hehn (Volkswagen AG – Wolfsburg, DE), Florian Klingler (Universität Paderborn, DE), Renato Lo Cigno (University of Trento, IT), Jörg Ott (TU München, DE), Elmar Schoch (BMW AG – München, DE), Jonathan Sprinkle (NSF – Alexandria, US), Erik Ström (Chalmers University of Technology – Göteborg, SE), Lars Wischhof (Hochschule München, DE), Andrea Zanella (University of Padova, IT), and Hongwei Zhang (Iowa State University, US)

4.1.1 CAV Applications and Co-Design with URLL V2X Networking

Vehicle-to-Everything (V2X) communication is a basic enabler to support applications for connected and automated vehicles (CAVs). CAV applications highly vary in terms of delivery requirements, e.g., reliability, timeliness, and throughput [1, 2, 3, 4]. Among them, many high-impact applications can benefit from predictable ultra-reliable and low-latency (URLL) V2X networking. The following are some examples:

- Network/cloud-assisted control of vehicles;
- Collaborative simultaneous-localization-and-mapping (SLAM) across traffic infrastructures and vehicles;
- Human-in-the-loop Augmented Reality (AR)-assisted driving;
- Real-time CAV control such as distributed collision avoidance and cooperative adaptive cruise control.

Achieving high reliability and low-latency is typically hindered by the fundamental trade-off between such metrics [5]. They can have different impact on CAV applications, and it is difficult to know what individual CAV applications require exactly at design time in general. Therefore, it is important to enable on-the-fly characterization of communication metrics and the control/choice of reliability-timeliness trade-off by applications.

The probabilistic nature of wireless communications and inherent uncertainties in V2X networks shall be considered in application and networking co-design. For instance, the interface between applications (e.g., CAV control) and networking shall capture the nature of random communication/networking topology, and this shall be differentiated from potentially random CAV coordination/control topologies, too. To better utilize V2X communication resources, it is also important to decide whether raw data or processed/fused data are to be exchanged between CAVs.

4.1.2 URLL V2X Networking: Models and Approaches

To support application-networking co-design and in capturing the probabilistic nature of wireless communication, it is crucial to model performance metrics. One approach is as follows:

$$Probability\{D \leq D_0\} \geq P_0, \tag{1}$$

where D is the communication delay from one CAV to another (via short-range single-/multi-hop links, and/or long-range cellular communication) and it may include potential retransmissions, D_0 is the packet delivery deadline, and P_0 is the minimum probability of delivering each packet before deadline. This model is particularly suitable for event-triggered Decentralized Environmental Notification Messages (DENMs) and real-time wireless-networked control in general [6]. Alternative approaches capturing attributes such as time between delivery of consecutive packets (a.k.a. update delay [7]) and age of information [8] may be particularly suitable for Cooperative Awareness Messages (CAMs), a.k.a. Basic Safety Messages (BSMs) periodically exchanged among vehicles.

The above models can be extended to capture the time-varying nature of V2X networking. For instance, Model (1) may be extended to the following:

$$Probability\{D(t) \leq D_0(t)\} \geq P_0(t), \tag{2}$$

which captures time-varying dynamics in both application requirements and V2X networking. In practice, it is important to understand the timescales of dynamics in wireless communications and vehicles and capture their impact on V2X modeling. It will also be interesting to explore how to best use such/similar models in vehicle control and sensing. Besides temporal variations and correlations in V2X communication, it will also be important to understand spatial variations and correlations in V2X networking and explore how to apply them in system design and analysis.

Major contributors to V2X communication delay include medium-access control (MAC) delay (including channel contention but also neighbor/link discovery delay for mmWave), propagation delay, and transmission delay. MAC delay is a factor that can be reduced by many mechanisms, and mechanisms should be explored such as infrastructure-assisted scheduling/access-control, distributed Time Division Multiple Access (TDMA) in highly-dynamic settings, transmit-time-aware sampling, and transmit power control (which impacts conflict sets in channel access). In V2X networks, a common primitive is for a vehicle to share its state/operation with close-by vehicles via broadcast. Thus, effectively capturing and controlling broadcast delay becomes an important issue. In fact, there may exist different notions of broadcast delay depending on applications, and one way of defining broadcast delay is to capture the delay in reaching a certain percentage/subset of receivers (i.e., intended receivers).

For high communication reliability, there also exist a wide range of mechanisms that can be exploited in practice. Examples include the following:

- Using multiple antennas and/or communication systems to leverage the enabled diversity.
- Designing effective physical layer solutions such as error control coding, channel estimation, and so on.
- Designing effective MAC mechanisms such as predictable interference control, infrastructure / network-assisted mode, TDMA, priority-aware scheduling, and power control.
- Designing proactive Automatic Repeat Request (ARQ) mechanisms for broadcast which do not require feedback from (all) receivers and leverages predictable broadcast communication reliability control mechanisms.

For high reliability and low latency in V2X communications, it is also critical to leverage redundancy and diversity provided by heterogeneous wireless media and networks, for instance, microwave, mmWave, visible-light, and free-space optical (FSO) communication, as well as the potential availability of the cellular infrastructure. To effectively leverage heterogeneous V2X networks and transmission media in URLL V2X communication, it is important to consider different application requirements as well as properties (e.g., real-time, throughput, and reliability) of different wireless media/networks. It is also important to control the complexity of the resulting system and to potentially fuse non-coherent information from different communication channels in a holistic manner.

4.1.3 Heterogeneous V2X Networking

As mentioned in Section 4.1.2, different technologies can be leveraged to fulfill the challenging requirements of vehicular applications [4]. The most frequently studied family of wireless access technologies are based on Dedicated Short-Range Communications (DSRC) and IEEE 802.11p. The latter ones are mainly intended to support localized vehicle-to-vehicle (V2V) communications. On the other hand, cellular technologies have been considered to support long-range connectivity with remote entities. Recently, 3GPP has sprinted forward by designing in Release 14 the Cellular V2X (C-V2X) technology, which supports V2V interactions over the PC5 sidelink interface in the 5.9 GHz spectrum. The sidelink interface is expected to further evolve with the New Radio (NR) technology expected to be specified in Release 16 for fifth generation (5G) systems. Outside of cellular and 802.11p-based technologies, many other potential technologies candidate themselves for the support of V2X services [4].

There is a wide consensus on the need of a heterogeneous networking solution combining multiple technologies, while outperforming the behavior of a technology alone [4]. Interesting recent examples can be found in [9] where DSRC messages are used to improve mmWave beamforming procedures, and in [10] where DSRC and VLC are used to boost the performance of platooning applications.

However, whenever multiple networking technologies are combined, the challenge of defining a suitable architecture arises.

4.1.3.1 V2X Applications Taxonomy

Applications conceived for improving passenger safety and comfort are typically classified in categories [2, 3, 4] such as:

- **Safety Applications**, e.g., collision avoidance, vulnerable road users warning;
- **Traffic Efficiency and Management**, e.g., local road traffic information exchange;
- **Infotainment**, e.g., Internet access, on-line streaming services;
- **Remote Diagnostics**, e.g., monitoring of the charging state of an electric vehicle;
- **Cooperative Driving**, e.g., platooning, cooperative maneuvering.

The requirements regarding latency, data rate, and reliability for these classes vary to a large extent – for example, safety-related applications based on a cooperative awareness by local communication often require a latency of less than 100 ms, whereas traffic efficiency or remote diagnostic applications can tolerate latencies in the order of tens of seconds.

While for the first four mentioned application classes products are already on the market, cooperative driving applications are not yet implemented in the field, being uniquely tightened to autonomous vehicle operation. They are a representative example of URLL applications [11, 1], that could benefit from a combination of multiple radio and networking technologies, as already discussed in Section 4.1.2.

4.1.3.2 Challenges in Heterogeneous Networking

Besides fulfilling the performance requirements of the respective applications by blending available networking technologies, several further challenges exist – often caused by the different concepts of the technologies.

Addressing

A common assumption for applications requiring communication in the local area (such as safety applications) is to rely on broadcasting, while long distance communication (e.g., for infotainment or remote diagnostics) is performed via a backend server and based on unicast addresses such as IPv4/IPv6 addresses. However, for cooperative driving some use-cases (e.g., negotiation of trajectories between two vehicles or within a platoon) might require a reliable unicast communication in the local area. In this case, a suitable addressing scheme is required, which may need to be complemented by a proper neighbor discovery approach. Some solutions are currently under discussion within IETF [12], but for 802.11p/WAVE networks, not for the C-V2X technology. Furthermore, the vehicle itself may have different addresses for different wireless technologies, leading to the need for a global identifier such as the Vehicle Identification Number (VIN). However, this global identifier might contradict privacy requirements – which could be solved by using temporary identifies (similar to the Temporary Mobile Subscriber Identity, TMSI, in cellular networks but technology independent).

Message Format

Currently, message formats, e.g. the format of a CAM, are often bound to a specific communication technology, despite their access-neutral design. Within a heterogeneous V2X network, a conversion of message formats as well as an aggregation of several messages coming from different communication interfaces can become necessary.

Network Selection

Each vehicle continuously monitors which communication networks are detected in the local situation. When multiple networks are available, selection criteria such as the network load or Quality of Service (QoS) guarantees need to be applied in order to select the optimal network for the local situation. If a handover to the selected technology is assumed, these criteria are sometimes referred to as handover triggers [13]. Instead of performing a handover, simultaneous usage of multiple technologies or per-packet selection of a communication technology [14] can lead to a better performance. The issue of where the network selection should be enforced has to be addressed (e.g., cloud-assisted hybrid vehicular networking [15] or a completely distributed approach.)

Application Model

Due to the wide range of existing and future V2X applications (as already mentioned in Section 4.1.3.1) it is still an open question which application model(s) should be assumed for V2X networking. Depending on the respective applications, a request-response model, a service-oriented model, or a publish-subscribe approach might be more suitable.

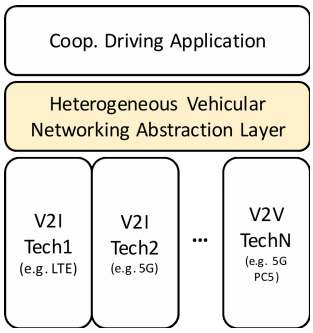

Figure 1 Heterogeneous networking abstraction layer avoiding a redundant implementation of technology selection and message format/address conversion.

4.1.3.3 Vehicular Network Architecture

For vehicular networking, several network architectures have been specified, for example ETSI ITS-G5 in Europe (ETSI EN 302 665), IEEE 1609.0/Wireless Access in Vehicular Environments (WAVE) in the US or ARIB STD-T109 in Japan [13]. Besides these standardization efforts, networking solutions for heterogeneous networks have also attracted a widespread interest in the research community which lead to a large number of publications and research projects (a survey of past and recent research efforts can, e.g., be found in [2]).

When considering hybrid vehicular networks where a vehicle has protocol stacks for several vehicular communication technologies on-board, three variants were discussed in the working group:

Class A Traditionally, in the on-board network of a vehicle, an application runs on a single, dedicated electronic-control unit (ECU). In this traditional approach an application, such as collision avoidance, transmitting and receiving CAMs would have the required protocol stack implemented on its ECU, for example the ITS-G5 stack. As a consequence, a single technology per vehicular networking application is used.

Class B The single communication per application approach (Class A) does not allow to leverage the benefits of combining different technologies for a single application. For example, the icy-road ahead warning application using cellular communication in case no other vehicles are in direct communication range can compensate a low market-penetration situation. This can be particularly important in the phase of market introduction of a communication system such as ETSI ITS-G5 or WAVE. One approach to overcome this restriction is to give an application direct access to multiple communication technologies and let the application decide which is the most appropriate technology for the current situation. Since this technology selection process might depend on information on the current status of the communication system (Section 4.1.3.2), the application needs access to all relevant parameters and to implement suitable message formats for all used technologies.

Class C In order to avoid a redundant implementation of status monitoring, message formats and address conversion, this class of hybrid networks introduces a Heterogeneous Vehicular Networking Abstraction Layer (HVNAL), as illustrated in Fig. 1.

The basic idea of the HVNAL is to hide the complexity of the heterogeneous network

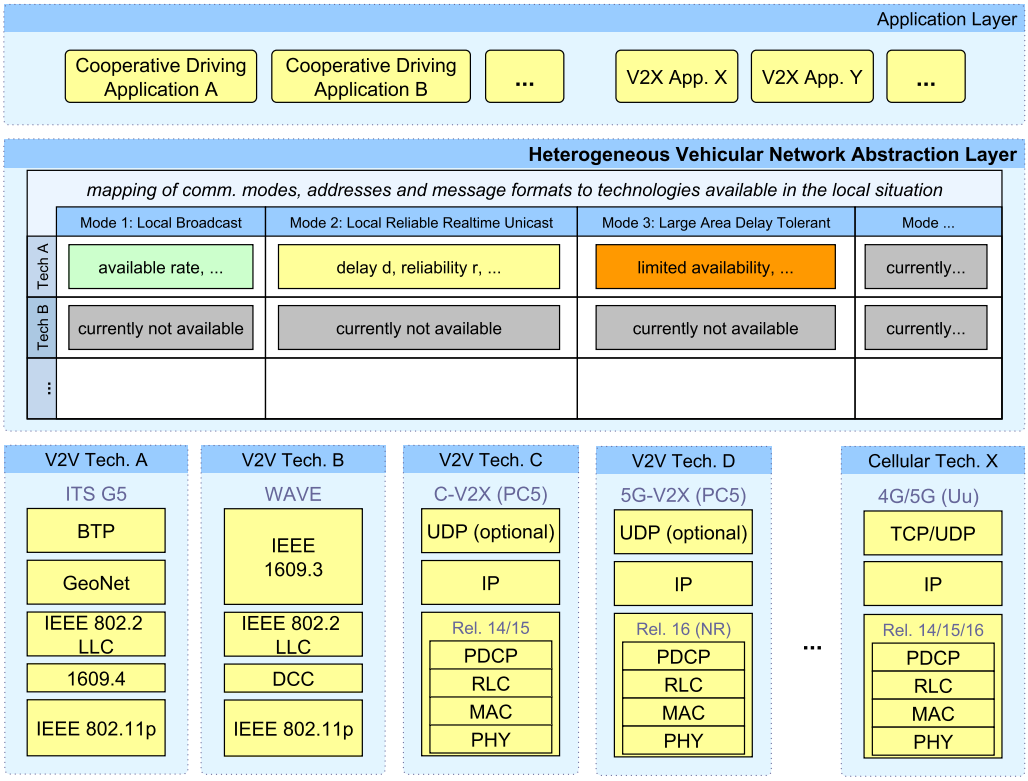

Figure 2 Example illustrating the realization of a vehicular networking architecture implementing a heterogeneous vehicular network abstraction layer.

and to be able to implement V2X applications (to some extent) independent of the detailed knowledge of underlying technologies. Thanks to the aforementioned layer in the V2X network architecture, on the one hand, future communication technologies could be introduced without requiring modifications in the individual V2X applications. On the other hand, novel applications, currently unknown, could be supported on top of existing technologies. However, as illustrated in the example in Fig. 2, with an increasing number of available V2X technologies, the complexity of the abstraction layer increases. Furthermore, it is still an open question – also known from classic Internet architectures – in which way the application requirements can be specified in a standardized format at the service access point between application and abstraction layer.

This could be one reason why often the currently discussed architectures for hybrid vehicular networks proposing a similar approach focus on single aspects, for example on load and resource sharing between cellular and direct/ad-hoc networks as in the system investigated by Zheng et. al. in [16]. Here, a Hybrid Link Layer (HLL) for load and resource sharing between cellular networks and IEEE 802.11p is introduced. An alternative approach could be an overlay protocol layer such as the Hybrid Overlay Protocol (HOP) layer in [14] which uses a concept of context indicators to select communication technologies and additionally provides services for data forwarding and aggregation.

4.1.4 Conclusions and Future Research

The requirements of future V2X and cooperative applications cannot be fulfilled by a single communication technology. Due to the large variance in V2X applications – including those envisioned for cooperative driving demanding ultra-reliable low-latency communication – multiple technologies will be required leading to heterogeneous vehicular communication networks.

Nonetheless the plenty of literature solutions, currently, there is no clear consensus on which approach should be implemented and if a hybrid architecture needs to be standardized or can be vendor-specific as long as the message formats and protocols for the individual communication technologies are standardized.

Future research is required to investigate promising solutions such as innovative URLL communication techniques and architectures supporting a heterogeneous vehicular network abstraction layer. It is worth observing that the idea of an abstraction layer is getting popular in the networking domain, with one of the most prominent instantiation being the Software-defined Networking (SDN) paradigm. The envisioned architecture could treasure SDN principles, currently investigated also a key solution for vehicular networks [17], and further advance them.

The design of the heterogeneous V2X networking architecture could also take inspiration by 5G systems. They face similar issues in the view of supporting multiple applications with different demands on top of the same but properly customized networking facilities, as for instance envisioned by network slicing solutions.

Overall, what clearly emerges from the breakout discussions is that the peculiarities of V2X applications and their continuous evolutions, especially in terms of strict delivery requirements, would require further efforts from the research community in the design of future-proof networking solutions.

References

1 R. Johri, J. Rao, H. Yua, and H. Zhang, "A multi-scale spatiotemporal perspective of connected and automated vehicles: Applications and wireless networking," *IEEE Intelligent Transportation Systems*, vol. 8, no. 2, pp. 65–73, 2016.

2 E. Ndashimye, S. K. Ray, N. I. Sarkar, and J. A. Gutiérrez, "Vehicle-to-infrastructure communication over multi-tier heterogeneous networks: A survey," *Computer Networks*, vol. 112, pp. 144–166, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128616303826

3 C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G network slicing for vehicle-to-everything services," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 38–45, 2017.

4 Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, "V2X access technologies: Regulation, research, and remaining challenges," *IEEE Communications Surveys & Tutorials*, 2018.

5 H. Zhang, X. Liu, C. Li, Y. Chen, X. Che, F. Lin, L. Y. Wang, and G. Yin, "Scheduling with predictable link reliability for wireless networked control," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 6135–6150, 2017.

6 Y. Chen, H. Zhang, N. Fisher, L. Y. Wang, and G. Yin, "Probabilistic per-packet real-time guarantees for wireless networked sensing and control," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2133–2145, 2018.

7 B. Kloiber, C. Rico-Garcia, J. Härri, and T. Strang, "Update delay: A new information-centric metric for a combined communication and application level reliability evaluation of cam based safety applications," 2012.

8 S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on.* IEEE, 2011, pp. 350–358.

9 J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, 2016.

10 M. Segata, R. L. Cigno, H.-M. M. Tsai, and F. Dressler, "On platooning control using IEEE 802.11 p in conjunction with visible light communications," in *Wireless On-demand Network Systems and Services (WONS), 2016 12th Annual Conference on.* IEEE, 2016, pp. 1–4.

11 5GPP, ERTICO ITS EUROPE, and European Commission, *5G Automotive Vision*, Oct. 2015. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf

12 "IPWAVE ipv6 neighbor discovery for prefix and service discovery in vehicular networks," March 2018.

13 K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and cellular network technologies for V2X communications: A survey," *IEEE transactions on vehicular technology*, vol. 65, no. 12, pp. 9457–9470, 2016.

14 S. Gopinath, L. Wischhof, C. Ponikwar, and H.-J. Hof, "Hybrid Solutions for Data Dissemination in Vehicular Networks," in *Proc. 8th International Wireless Days Conference*, Toulouse, France, Mar. 2016.

15 T. Higuchi and O. Altintas, "Leveraging cloud intelligence for hybrid vehicular communications," in *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on.* IEEE, 2017, pp. 15–20.

16 K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2377–2396, 2015.

17 G. Han, M. Guizani, Y. Bi, T. H. Luan, K. Ota, H. Zhou, W. Guibene, and A. Rayes, "Software-defined vehicular networks: Architecture, algorithms, and applications: Part 1," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 78–79, 2017.

4.2 Sensing and Data Management

Suman Banerjee (University of Wisconsin – Madison, US), Aruna Balasubramanian (Stony Brook University, US), Sonia Heemstra de Groot (TU Eindhoven, NL), Albert Held (Daimler AG – Ulm, DE), Frank Kargl (Universität Ulm, DE), Renato Lo Cigno (University of Trento, IT), Thomas Strang (German Aerospace Center-DLR, DE), Lars Wolf (TU Braunschweig, DE), and Andrea Zanella (University of Padova, IT)

Cooperative vehicular systems depend heavily on the ability of each vehicle to sense its neighborhood and use this information to interact with neighboring vehicles and the infrastructure. We anticipate that each vehicle is equipped with multi-modal sensors, e.g., to gather audio-visual data, for ranging and positioning, for inertial measurements, and to infer presence of various objects in the neighborhood in three dimensions. Common examples include LIDAR, RADAR, cameras, inertial measurement units, and more. Using these sensed

data, the vehicles aim to learn about other vehicles, pedestrians, various obstacles, and road signage. Given these goals, a number of challenges come to the fore, that are discussed next.

Given the raw sensed data can have a very high volume, is it practical to upload all such data to central repository for different applications. While archiving such raw data might be useful in certain cases, for most real-time applications it is likely adequate to process such data streams locally and only upload or share vectorized content. However, such a design raises a new question – can data captured by one vehicle and processed locally be trusted by another? This is particularly critical if a second vehicle makes various actuation decisions based on data it receives from non-local sources. Sensed data is often inherently noisy. Further, depending on the quality of the sensors and the nature of processing applied, additional inaccuracies maybe introduced. Hence, in situations where a vehicle wants to take an action based on data sourced from a different vehicle, the former might benefit from access to the raw data – especially if the first vehicle is more willing to trust its only processing capabilities to extract valuable information out of the sensors.

A next critical issue arises in understanding data ownership. This is a particularly complex issue as there are many stakeholders possible in the data that is sensed. The vehicle manufacturer, the vehicle owner, the objects being sensed, all may have different claims to the data being sensed. This will potentially impact who can do what with the data. Related to ownership is data privacy. For example, camera-based or LIDAR-based system provides raw input from which various contexts of a vehicle and its neighborhood can be extracted. Many vehicle-based video streams go through common privacy preserving techniques, such as face blurring. However, in some cases greater obfuscation techniques might be necessary. The nature of privacy preserving techniques might depend on the context and applications being considered over the data.

Further, sharing such sensed data begs the question on incentives. What is the incentive to share data between different vehicles, especially when they belong to different manufacturers or fleet owners? Some natural incentives exist for sharing data between vehicles of the same manufacturer, e.g., to allow such vehicles to perform some functions like platooning better. Furthermore, the software and hardware subsystems in such vehicles from the same manufacturer are managed by a single entity and data trust is more practical in such scenarios. It is also possible that vehicles across manufacturers may share data with each other in scenarios that improve mutual safety, especially if sharing under such scenarios are mandated through regulations. It is also is possible to imagine a credit-based architecture that facilitate sharing across vehicles at a broader scale. The role of regulation might also play an important role in this context.

Sharing of data between vehicles and between vehicles and the infrastructure also require appropriate infrastructure support, especially at the edges of the networks. Requirements include suitable processing, storage, and communication channels to facilitate such sharing, especially when latency is critical.

Finally, to facilitate data sensing and sharing, it is important to define appropriate standards that describe the data and perhaps even policies that identify how different entities may utilize such data for different applications. Overall, sensing, sharing, and data management have many unique challenges that require significant further investigation from various technical standpoints.

4.3 New Use Cases

Sinem Coleri Ergen (Koc University – Istanbul, TR), Onur Altintas (TOYOTA InfoTechnology Center USA – Mountain V, US), Ali Balador (RISE SICS – Västerås, SE), Suman Banerjee (University of Wisconsin – Madison, US), Claudia Campolo (University Mediterranea of Reggio Calabria, IT), Falko Dressler (Universität Paderborn, DE), Eylem Ekici (Ohio State University – Columbus, US), Sonia Heemstra de Groot (TU Eindhoven, NL), Geert Heijenk (University of Twente, NL), Renato Lo Cigno (University of Trento, IT), Michele Segata (University of Trento, IT), Christoph Sommer (Universität Paderborn, DE), Jonathan Sprinkle (NSF – Alexandria, US), Andrea Zanella (University of Padova, IT), and Hongwei Zhang (Iowa State University, US)

The widespread usage of vehicular networking depends highly on developing a large number of use cases. This working group focused on brainstorming new use cases for vehicular networking. We have generated the following list:

1. **Emergency:** Earthquake and other disaster scenarios may destroy the cellular infrastructure. In those cases, cellular communication may not possible and vehicle-to-vehicle communication may be the only option.
2. **City-wide Surveillance:** Vehicular communication can be used to track people. This information can be further used to derive the movement pattern of people.
3. **Detection of bicycles and pedestrians:** A phone application on the bicycle and pedestrians can communicate with the cloud or directly with the vehicles. The vehicles can then collect this information and combine them with sensor data to detect bicycles and pedestrians.
4. **Distributed black box:** Each car can be considered as a black box, which combines sensor and communication data. When there is an event, such as accident, these data can be retrieved from the database to analyze the statistics related to the event.
5. **Socializing:** The drivers within vehicles close to each other can send warning messages or just to say hello to each other.
6. **Enforcing unwritten rules:** In some countries, there are some unwritten rules. For instance, in India, people give right of way to people of higher status. Vehicular communication can be used to enforce these rules.

4.4 Human-in-the-Loop

Falko Dressler (Universität Paderborn, DE), Eylem Ekici (Ohio State University – Columbus, US), Thorsten Hehn (Volkswagen AG – Wolfsburg, DE), Renato Lo Cigno (University of Trento, IT), Christoph Sommer (Universität Paderborn, DE), and Lars Wischhof (Hochschule München, DE)

4.4.1 Human-in-the-Loop

Despite the many advances in the fields of automated driving, vehicular networking, and now cooperative driving, one major component of the next generation transportation systems has often been ignored or only partially considered: the actual user. Human users as part of the technical system have a significant impact on the design and efficacy of such systems. A novel research domain incorporating human interactions has been termed Cyber Physical Social Systems (CPSS) [1, 2]. In addition to immediate issues related to transitioning from the current road usage to a fully automated one, most prominently the question how to deal with legacy systems, other important societal questions also arise. Considering that human users need to be put first, also to increase public acceptance, the (technical) system must be able to deal with these interactions – hopefully with little impact on efficiency and safety.

4.4.2 Human Beings as a Source of Errors

As cars are slowly moving towards full automation, more functionality is being taken over by the computer. The presence of humans in the decision loop, however, is a source of great uncertainty. This is supported by findings that accurate self-assessment of driving capabilities is massively biased – the well-known Dunning-Kruger effect [3] describes such cognitive bias, wherein persons of low ability suffer from illusory superiority when they mistakenly assess their cognitive ability as greater than it actually is.

So, taking humans out of the control loop has been proposed in our discussion group as a straight forward way of resolving this issue, which would also be the case in the steady state, i.e., when humans feel comfortable with the decisions exclusively made by vehicles. It has been observed that the ownership is one of the major factors in not releasing control of the vehicle – people tend to experience discomfort if *their* car is driven by a computer. However, changing trends in ownership (move towards shared mobility – the 'Robo-Taxi' concept) may alleviate some of these issues, leading to public acceptance of automatic driving systems.

It has also been argued that, as long as fully automated driving is not realized, inefficiencies will persist. As an easy way of transitioning to automated driving, back-seat driving options (the former 'driver' of the car now acting more as a 'captain', ordering a computer what to do) could be used as a transition to full autonomous driving. Similarly, an avatar interacting with the passenger (a natural evolution beyond simple indications of decision processes [4]) would help increase acceptance.

4.4.3 Interfacing Humans and Machines

The interface of computer-driven vehicles with humans and human-controlled vehicles is necessary to ensure harmonious coexistence. This is a longer term issue as it does not depend on acceptance by the driver. The challenges are in replacing interactions with pedestrians

and other human drivers. Such interactions include visual interactions (eye contact, gestures) within appropriate contexts (deployment location, societal norms and habits).

More specifically, interactions with pedestrians are very important: Existing examples include brake lights on the front of the car and projecting crossing lanes for pedestrians. In general, signaling and interacting with pedestrians has been identified as an important topic that need great attention and further research, as pedestrians are still a major components of road casualties. Indeed, research and attention should be extended to all vulnerable road users (VRU) like bicycle riders, but also moped and e-bike users, whose transport mean will not be automated in the foreseeable future. So, adding interfaces (e.g., cell phones) may resolve interaction issues in the short term, and also help adapt to various cultures and environments, but more advanced solutions should also be invented.

4.4.4 Automated Decision Making May Cause Harm

The final discussion was on the moral machine [5, 6], i.e., how to make decisions in critical decision junctures. An existing policy by Volvo, accepting all responsibility in case of an accident, may be relinquishing too much control to the OEM, which might try to reduce the cost rather than implementing other policies. Although automated cars would create a simpler pricing opportunity for insurers, this does not address criminal liability problems. From the human acceptance perspective, it may not be very easy to convince the driver that the controller's decision is better than any real-time decision the driver could have taken and executed.

At this point, we could envision that protecting the driver takes priority in control algorithm design. Protection of others, such as pedestrians, is also taken into account, but as secondary considerations. Legal systems are driven by what is acceptable by society – and legal systems will drive the algorithms controlling the vehicle behavior. Another approach would be to emulate human behavior (possibly including randomness) as a policy.

References

1 A. Sheth, P. Anantharam, and C. Henson, "Physical-Cyber-Social Computing: An Early 21st Century Approach," *IEEE Intelligent Systems*, vol. 28, no. 1, pp. 1541–1672, Feb. 2013.

2 F. Dressler, "Cyber Physical Social Systems: Towards Deeply Integrated Hybridized Systems," in *IEEE International Conference on Computing, Networking and Communications (ICNC 2018)*. Maui, HI: IEEE, Mar. 2018, pp. 420–424.

3 J. Kruger and D. Dunning, "Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments," *Journal of Personality and Social Psychology*, vol. 77, no. 6, pp. 1121–1134, Dec. 1999.

4 K. Sonoda and T. Wada, "Displaying System Situation Awareness Increases Driver Trust in Automated Driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 185–193, Sep. 2017.

5 J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," *Science*, vol. 352, no. 6293, pp. 1573–1576, Jun. 2016.

6 A. Shariff, J.-F. Bonnefon, and I. Rahwan, "Psychological roadblocks to the adoption of self-driving vehicles," *Nature Human Behaviour*, vol. 1, no. 10, pp. 694–696, 2017.

4.5 Safety-critical Vehicular Network Applications

Geert Heijenk (University of Twente, NL), Michele Segata (University of Trento, IT), Christoph Sommer (Universität Paderborn, DE), Thomas Strang (German Aerospace Center-DLR, DE), Lars Wolf (TU Braunschweig, DE), and Andrea Zanella (University of Padova, IT)

4.5.1 Introduction

Today's vehicles are built to very high standards of safety. Embedded systems in the cars, applications running on them, and networks connecting them are rigorously checked to ensure high reliability. As a next step, the safety of drivers will be further enhanced by networks between cars, supporting additional applications that afford the cooperation of multiple cars on the road, e.g., for intersection control or for platooning. However, as vehicle drivers rely more and more on these applications, the emerging behavior of cooperation between cars becomes safety-critical itself: Failure of the cooperation system to perform in a proper manner may directly result in death or serious injury to people. The working group on safety-critical vehicular network applications discussed how, and to what extent, vehicular network applications can be made as close as possible to 100% safe in the presence of faults.

For fully automated cooperative driving systems, foreseen in the future, there appears to be a trade-off between providing functionality and efficiency on the one hand and providing safety on the other hand. At the very extreme, providing 100% safety might mean providing no mobility at all. On the other hand, the fully automated cooperative paradigm can potentially enhance the road safety to levels that may not be otherwise reachable. It is a task of the vehicular networking community to provide insights into this trade-off – and to investigate which level of functionality and performance can be provided at which risk of major failure. Another task is to define measures to minimize the effects of system malfunctioning, especially those due to communication failures.

4.5.2 Comparison to other modes of transport

The working group discussed how the trade-off between performance and risk of failure is made for other modes of transport, especially railways and aviation. A key difference was obvious in the discussion: Whereas the railway fall back to 'fail-safe'(zero velocity) whenever safety requires, an aircraft cannot do the same – it has to remain in some 'fail-operational' mode which allows to continue flying (e.g., go-around maneuver in case of blocked runway). There are many further differences with these transport modes, e.g., the stakeholders, and the incentives for these stakeholders differ, their insights might be applicable to cooperative driving. In railway operation, safety and efficiency seem not to be addressed at the same level. The decision to install a certain safety system along a railway is made almost solely to increase safety, without taking efficiency into consideration too much. Nevertheless, railways do not seem to have a significantly lower capacity per track (12 trains with 800 passengers per hour per track[1] than road systems per lane (2400 cars with 4 passengers capacity per hour [1]). In railway operations, the human train driver is severely limited by what the (safety) system allows him/her to do. Also, in aviation, there seem to be many procedures in

[1] https://en.wikipedia.org/wiki/Route_capacity

place assuring the safety of the aviation as a system. As opposed to railway operation, here it is the pilot who is empowered to carry final responsibility of the aircraft. From another perspective, railway safety is mostly under the control of a single central entity, while in the aviation domain, it is provided in a hierarchical manner, where the aircraft's trajectories are planned and authorized well in advance by a central authority, but can be adjusted during the flight (upon permission) to cope with unpredictable situations, and finally the control can be taken by the pilot in case of immediate danger.

One possible reason for such a different approach to safety in railways and aviation may be the different 'trajectory plasticity' of two scenarios: the trains, indeed, are constrained to follow the train tracks, with basically no possibility of deviation from the planned trajectory, while aircraft can potentially move freely in the 3D space. The plasticity of road vehicle trajectory has its own characteristics, since it is limited by the presence of other nearby vehicles and, obviously, by the road/lane bounds, but can be dynamically and continuously changed within these constraints. Therefore, the approach to road safety in cooperative autonomous driving scenarios may also be different from those considered in railway and aviation scenarios to reflect the specific characteristics of trajectory plasticity of road vehicles.

4.5.3 Directions for solutions

Inspired by the analogy of aviation and railways, two directions for solutions were identified by the working group.

- If the operation of cooperative driving is based on (and depending on) situational awareness by means of sensing and communications, the system (that is, all vehicles in it) should have a good knowledge of the quality of the situational awareness – in terms of accuracy, trustworthiness, freshness, completeness, and correlations in these. Only if the quality of the situational awareness is close to 100%, full functionality and/or performance of the cooperative driving can be employed. If not, the system has to reduce its operation to a less functional, less efficient point of operation. As an example, in the case of platooning operation, headways can be increased depending on the quality of the situational awareness. Of course, degradation time does play an important role here. It should also be taken into account that future autonomous and cooperative systems will not consider the human driver as a possible fail-functional fallback option. As autonomous systems will more and more take over control, humans will progressively loose driving experience. Handing over control to humans might thus increase the likelihood of dangerous situations. This is completely the opposite of what happens in aviation, where pilots are required to manually land the aircraft to keep trained and will resort to automatic landing only if the weather conditions are not good enough. In addition, commercial pilots are required to renew their license every few years. This process is clearly not sustainable for road vehicles, as with autonomous and cooperative driving we are aiming to progressively reduce human intervention.
- Inspired by the railway and aviation scenarios, safety can be considered in a hierarchical way, where all vehicles within a group can operate at very small distances, tightly controlled with highly fault-tolerant operation (compare a series of aircraft in landing configuration on a final glidepath of a runway), whereas safety between groups can be ensured by a combination of less strict coordinated control and larger distances/headways (compare safety between trains). In such a scenario, homogeneity of group members will improve safety, but also introduce dilemmas such as how to enforce homogeneity, e.g., by limiting braking capacity of vehicles in a platoon.

References

1 National Research Council – Transportation Research Board, *Highway Capacity Manual 2000: HCM 2000 (metric units)*, 4th ed. TRB, 2000.

4.6 Security and Privacy

Frank Kargl (Universität Ulm, DE), Albert Held (Daimler AG – Ulm, DE), Elmar Schoch (BMW AG – München, DE), Christoph Sommer (Universität Paderborn, DE), Thomas Strang (German Aerospace Center-DLR, DE), Isabel Wagner (De Montfort University – Leicester, GB), and Andrea Zanella (University of Padova, IT)

The breakout group on security and privacy tackled three aspects: First, the role of security in systems engineering of automated connected vehicles (section 4.6.1). Second, questions regarding the privacy of cooperative Intelligent Transportation Systems (ITS) and smart cities (section 4.6.2). Third, security perimeters and attack vectors in automated connected driving (section 4.6.3).

4.6.1 The Role of Security in Systems Engineering

The breakout group started its discussions by tackling one of the fundamental questions: *Why do we require a car to be secure?*. Discussions quickly arrived at an interesting angle: dependability. In brief, users must be able to trust that the car does what it is supposed to do. Any attack that does not impact dependability is likely to receive little attention from users – not unlike how users are perfectly content with having their personal computers participate in bot nets (attacking servers, sending spam, ...) as long as their own performance is not impacted. Following this reasoning, one might arrive at the realization that (just like home computers) attacks on cooperative automatic cars might be treated by operators as fundamentally unavoidable. All that might be needed is ensuring that a system remains operational in the face of attacks – though possibly with reduced functionality (e.g., using backhaul control loops). What would be needed, though, are guarantees about the maximum impact of an attack on (safety) application performance (and, as a prerequisite, the ability to quantify the effect of attacks and to discriminate between attacks and failures).

One way towards this might be control theory that natively accounts for security through a true fusion of security engineering and control engineering. In a first step, this might take the shape of a new twist on error modeling: the use of error models that are representative of the effect of security attacks. Further on, it will be necessary to find 'resilience' boundaries of control systems to malicious information.

Formalizing the security problems of connected automated vehicles, however, is a complex task. Other than in related work like that of Meadows and Pavlovic [1] (where objectives are often straightforward and questions about a compromise of the system are often a simple, binary decision), some attacks on connected automated vehicles may only affect input data stochastically, with the effects adding up and the goal is to keep some specific processes within given bounds. So formalisms like those of Meadows and Pavlovic [1] would have to be extended to also cover such more complex tasks.

On the plus side, however, cooperative automated vehicles offer the opportunity to 'offload' phenomenological detection of attacks to surrounding vehicles. As long as a bare minimum of functionality remains active and untainted in the compromised vehicle, surrounding vehicles could be able to collaboratively issue a *failsafe* command to the vehicle's controller – an idea not unlike that of an *Air Marshal*.

4.6.2 Privacy of ITS and Smart Cities

The second topic discussed by the breakout group was that of privacy in cooperative ITS and smart cities. One of the key problems here is that location privacy is still not well understood and no well-established privacy metrics are commonly applied to compare solutions [2]. This is compounded by user interface issues: *'How can one empower users to take good privacy decisions?'* is a question that has – to date – no commonly accepted solutions.

In connected fully automated driving, however, the situation changes somewhat. Take, for example, a *Robo-Taxi* scenario. As a driver no longer exists and the identity of passengers is no longer intimately tied to that of the car owner, external attacks like overhearing or license plate recognition do not necessarily reveal information about the identity of passengers.

At the same time, however, the attack surface for internal attacks (by the operator) increases. Here, the amount of data generated, processed and stored will increase dramatically. On the plus side, traditional privacy preserving techniques may apply in this scenario.

Concerns can also be raised about the impact of connected automated vehicles on the privacy of others: With computer vision systems in all such cars (and manufacturers and operators likely recording all data in order to limit their liability) massive amounts of data will be collected about other road users, similar to the infamous Google Street View project. If created data could be bound to strong privacy policies, policy enforcement architectures like investigated by the PRECIOSA project [3] may be a viable option in this scenario to prevent data being abused.

4.6.3 Security Perimeters and Attack Vectors

Work in the breakout group concluded with a consideration of novel attack vectors on cooperative automated vehicles and a discussion on new security perimeter concepts (see Figure 3).

Two identified attack vectors that are specific to cooperative automated vehicles are attacks on sensors and sensor integrity, e.g., by feeding fake Lidar echoes to the car [4] or intelligent spoofing of GPS signals) and attacks on map data (some of which might be crowdsourced). Ways around these attack vectors can include communication of error ranges (in the case of sensors) and automated verification with measurements of many sensors (in the case of map data).

Another issue discussed in the working group was one of attack surfaces: In cooperative driving, many applications like CACC envision the interface to driving functions to be exposed to external entities. As a consequence, the physical boundaries of the vehicle can no longer serve as an isolation perimeter. Rather, semantically-aware input validation will be required. This, however, opens up the problem of misbehavior detection filtering out isolated, only locally-observed, but very relevant information such as about an accident or a small patch of black ice. Workarounds currently considered would be the explicit signaling of 'Here's some data, and I know it is hard to believe', i.e., an *accident flag* in messages – although the impact of this remains unclear.

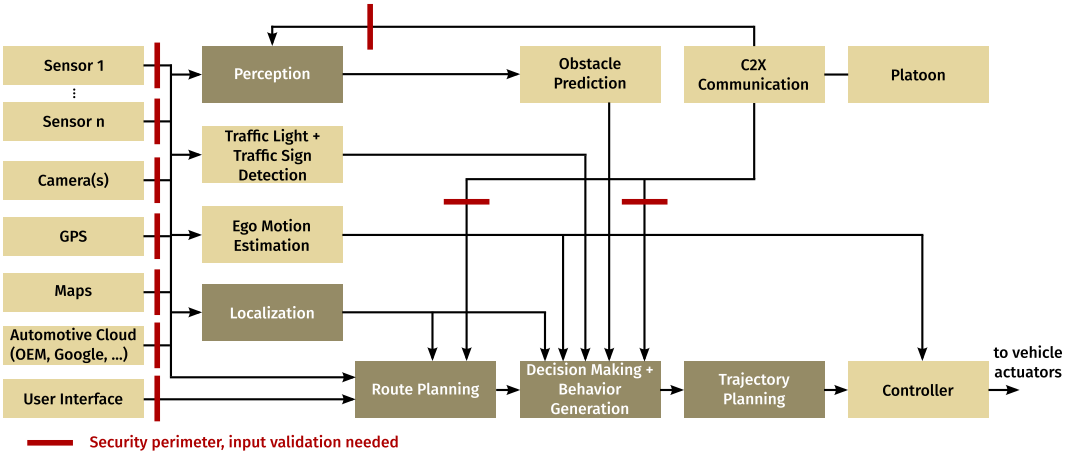

Figure 3 Arrows marked with a red bar are part of the vehicle's security perimeter and may be subject to novel attacks.

As concerns isolation of security domains inside the vehicle, it can be expected that new EE architectures lead to reduced isolation of formerly distinct components and functions. What would be needed here is are dynamic security perimeters that might extend beyond individual vehicles. Examples are *trust groups* with, e.g., a cohort of vehicles such as a platoon as a joint security parameter of all vehicles. This could be complemented by a weaker trust boundary between vehicles within the cohort – albeit this runs into the obvious problem of cooperation across OEM borders.

References

1 C. Meadows and D. Pavlovic, "Formalizing physical security procedures," in *Security and Trust Management*, A. Jøsang, P. Samarati, and M. Petrocchi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 193–208.

2 I. Wagner and D. Eckhoff, "Technical Privacy Metrics: A Systematic Survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 57:1–57:38, June 2018.

3 F. Kargl, F. Schaub, and S. Dietzel, "Mandatory enforcement of privacy policies using trusted computing principles," in *Intelligent Information Privacy Management Symposium (Privacy 2010)*. Stanford University, USA: AAAI, March 2010. [Online]. Available: http://vts.uni-ulm.de/doc.asp?id=7278

4 J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," in *Black Hat Europe*, November 2015. [Online]. Available: https://www.blackhat.com/docs/eu-15/materials/eu-15-Petit-Self-Driving-And-Connected-Cars-Fooling-Sensors-And-Tracking-Drivers-wp1.pdf

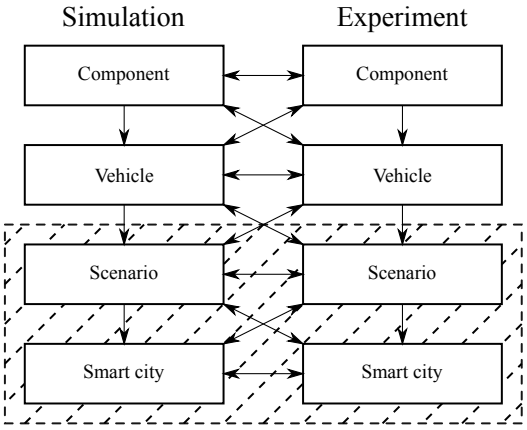

Figure 4 Extension of the validation process towards city-scale scenarios.

4.7 Simulation, Modeling, and Testing

Christoph Sommer (Universität Paderborn, DE), Wai Chen (China Mobile Research Institute – Beijing, CN), Geert Heijenk (University of Twente, NL), Michele Segata (University of Trento, IT), Jonathan Sprinkle (NSF – Alexandria, US), Erik Ström (Chalmers University of Technology – Göteborg, SE), Isabel Wagner (De Montfort University – Leicester, GB), and Hongwei Zhang (Iowa State University, US)

4.7.1 The new scale and dimensions of simulating cooperative mobile systems

Traditionally, system development of (communicating) cars starts with an idea about a component that gets implemented and validated in simulation at many different levels of abstraction. If successful, this component can then move on to lab tests of prototypes, lab tests of a complete car, then field operational tests for certification. This implies that, first, the design process stops at component level and, second, that certification and benchmarking is possible at the level of an individual car or a small group of cars. This process is proven for the development of individual systems (like the traditional communicating car).

For cooperative mobile systems, however, certification and benchmarking according to metrics like *fairness* or *safety* will no longer be possible without considering a large number of cooperating cars – up to city scale trials. In addition to requiring experimentation at scale, these trials will need to be perfectly controlled as well; thus, simulation will emerge as the prime means of both validation and certification of such systems.

As a consequence, the research community will need to find a way towards simulating city scale systems of cars with behavior that is identical to the system under study – ideally, provably so.

Moreover, other than the established approach of employing simulation only at the component or car level, simulation and experimentation will need to be employed for validation (and, even more importantly, for cross-validation) at each step of the composition process (from components, to cars, to individual convoys of cars, to smart cities). Figure 4 shows how the classic approach should be extended: The dashed area highlights the new validation domain that is inherently introduced with cooperative mobile systems.

This is particularly challenging for two main reasons: First, the assumption that a composition of (individually validated) systems can simply be considered valid without further testing is dubious at best. Second, in mixed traffic there will need to be a decidedly *human* component modeled in the system – an aspect that also needs further study.

Aside from this new scale of simulating cooperative mobile systems, simulation will also need to explore new dimensions:

In fully automated systems, simulation and benchmarking can no longer fall back on human behavior in a given situation as the gold standard against which to measure system performance. Ultimately, the objective is that these automated systems will outperform humans, in terms of safety and efficiency of the traffic system. For intermediate performance levels, human behavior could be used as a standard to test against.

Another aspect is that, for performance studies and compliance testing of such systems, typical behavior of a system as complex as a complete smart city (as well as individual, rare events) will need to be simulated using novel metrics: In addition to safety and efficiency (for which approaches have been established in the literature), security, privacy, fairness, and resilience are all qualities of a cooperative mobile system for which metrics will need to be defined and tested with.

Finally, as the level of complexity of the functionality provided and hence the scenarios to be tested is ever increasing, huge amounts of data have to be collected from realistic traffic situations to be able to (re)create testing scenarios. Especially, all data from challenging driving situations, including sensed and communicated data, should be made available for simulation and testing purposes.

4.7.2 Towards reproducible simulation studies

A cross-cutting concern of simulation as a tool for research is ensuring reproducible studies, that is, allowing other researchers to both *(1)* independently verify the validity of conclusions and to *(2)* build on the findings of others.

In the early days of small-scale simulation (that is, simulation of just a few aspects of isolated components), the simulation model and its underlying assumptions could well be documented within the few pages of text a scientific paper may allow. With today's simulations encompassing vastly complex systems of multiple components all the way up to trained neural networks and the like, however, writing up a text-form description of this system in a way that allows an interested researcher to reproduce the results (let alone in a timely fashion) has become close to impossible.

The research community will thus need to take the next step in sharing data: Where other fields are simply sharing *result* data (if at all), our community must share *input* data. This data takes two different, complementary forms: First, simulation data, i.e., input traces or training sets, probably collected from real-world challenging driving situations. Second, simulation models and tools – either as full source code of the model, the tool, and all necessary libraries or as a (future-proof) ready-to-run simulation. Such data bundles must also document all the assumptions that have gone into their design and that might restrict their validity (lest other researchers, who might not be domain experts in the particular field, misuse the simulation model, the tool, or other input data).

For the latter step, the sharing of simulation models and tools in a perfectly reproducible form, it might be possible to take a page from the playbook of the DevOps community, who have been creating a wealth of tools for fully automated, reproducible software installations (Docker, ansible, ...) as well as documenting assumptions about their design.

Participants

- Onur Altintas
TOYOTA InfoTechnology Center USA – Mountain V, US
- Ali Balador
RISE SICS – Västerås, SE
- Aruna Balasubramanian
Stony Brook University – US
- Suman Banerjee
University of Wisconsin – Madison, US
- Claudia Campolo
University Mediterranea of Reggio Calabria – IT
- Wai Chen
China Mobile Research Institute – Beijing, CN
- Sinem Coleri Ergen
Koc University – Istanbul, TR
- Falko Dressler
Universität Paderborn – DE
- Eylem Ekici
Ohio State University – Columbus, US
- Sonia Heemstra de Groot
TU Eindhoven – NL
- Thorsten Hehn
Volkswagen AG – Wolfsburg, DE
- Geert Heijenk
University of Twente – NL
- Albert Held
Daimler AG – Ulm, DE
- Frank Kargl
Universität Ulm – DE
- Florian Klingler
Universität Paderborn – DE
- Renato Lo Cigno
University of Trento – IT
- Jörg Ott
TU München – DE
- Elmar Schoch
BMW AG – München, DE
- Michele Segata
University of Trento – IT
- Christoph Sommer
Universität Paderborn – DE
- Jonathan Sprinkle
NSF – Alexandria, US
- Thomas Strang
German Aerospace Center-DLR – DE
- Erik Ström
Chalmers University of Technology – Göteborg, SE
- Isabel Wagner
De Montfort University – Leicester, GB
- Lars Wischhof
Hochschule München – DE
- Lars Wolf
TU Braunschweig – DE
- Andrea Zanella
University of Padova – IT
- Hongwei Zhang
Iowa State University – US

Report from Dagstuhl Seminar 18211

Formal Methods and Fault-Tolerant Distributed Computing: Forging an Alliance

Edited by

Javier Esparza[1], Pierre Fraigniaud[2], Anca Muscholl[3], and Sergio Rajsbaum[4]

1 **TU München, DE,** esparza@in.tum.de
2 **University Paris-Diderot and CNRS, FR,** pierre.fraigniaud@irif.fr
3 **University of Bordeaux, FR,** anca@labri.fr
4 **National Autonomous University of Mexico, MX,** rajsbaum@im.unam.mx

Abstract

The Dagstuhl Seminar "Formal Methods and Fault-Tolerant Distributed Computing: Forging an Alliance" took place May 22-25, 2018. Its goal was to strengthen the interaction between researchers from formal methods and from distributed computing, and help the two communities to better identify common research challenges.

Seminar May 21–25, 2018 – http://www.dagstuhl.de/18211
2012 ACM Subject Classification Theory of computation → Distributed algorithms, Theory of computation → Verification by model checking
Keywords and phrases distributed computing, distributed systems, formal verification
Digital Object Identifier 10.4230/DagRep.8.5.60
Edited in cooperation with Marie Fortin

1 Executive Summary

Anca Muscholl (University of Bordeaux, FR)
Javier Esparza (TU München, DE)
Pierre Fraigniaud (University Paris-Diderot and CNRS, FR)
Sergio Rajsbaum (National Autonomous University of Mexico, MX)

The original motivation of this workshop has to do with the evolution of research in Computer Science. The first ACM conference on Principles of Distributed Computing (PODC) was held in 1982. The proceedings of its first editions included papers on distributed algorithms[1], formal methods for distributed systems[2], or a combination of the two. However, in 1990 the area of formal methods for distributed computing branched out, and started its own conference, the International Conference on Concurrency Theory (CONCUR), now in its 27th edition. PODC and CONCUR have become the premier conferences in their respective fields, and, after over 20 years of almost independent evolution, feel the need to close a gap

[1] Algorithms designed to run on computer hardware constructed from interconnected processors.
[2] Mathematically based techniques for the specification, development and verification of software and hardware systems.

Formal Methods and Fault-Tolerant Distributed Computing: Forging an Alliance, *Dagstuhl Reports*, Vol. 8, Issue 05, pp. 60–79
Editors: Javier Esparza, Pierre Fraigniaud, Anca Muscholl, and Sergio Rajsbaum
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

that slows down progress, limits the applicability of the results, and causes repetitions and inconsistencies.

Our seminar aimed at achieving synergy by bringing together the two research areas, both with deep understanding of distributed computation, but different perspectives. We had two longer tutorials, one about concurrent data structures by Ph. Woelfel and one about verification of concurrent programs by A. Bouajjani. In addition, we had several survey talks, on correctness in concurrent programming (H. Attiya), distributed runtime verification (B. Bonakdarpour), distributed property testing (K. Censor-Hillel), distributed synthesis (B. Finkbeiner), and parametrized verification (I. Konnov).

The scientific programme was quite dense, given that we had only 4 days and almost all participants proposed to give a talk. Exchanges were very lively, and the discussion that we had with all participants showed that this kind of workshop is a great opportunity to compare our approaches and find new research directions, inspired by the perspectives of the other community. We warmly thank Marie Fortin for the editorial work on this report and the Dagstuhl staff for the excellent conditions provided for our seminar.

2 Table of Contents

3 Overview of Talks

3.1 On Verifying Robustness of Concurrent Systems

Ahmed Bouajjani (University Paris-Diderot, FR)

Joint work of Ahmed Bouajjani, Mohamed Faouzi Atig, Egor Derevenetc, Michael Emmi, Constantin Enea, Roland Meyer, Burcu Ozkan, Serdar Tasiran

Concurrent systems are in general used by their clients under strong assumptions on their visible behaviors. This allows a modular design approach: at the level of the client, these assumptions allow to reason in an abstract way about the behaviors of the invoked systems.

For instance, the users of a shared memory may assume that the implementation of the memory is sequentially consistent, which means that it behaves according to the standard interleaving model where write/read operations are considered to be atomic, and immediately visible to all parallel users. In an another context, the users of web services may consider that their requests are handled atomically in a serial way, and in yet another context, the designers of protocols and distributed algorithms may consider that interactions between components are happening in a synchronous way, etc.

However, for performance reasons, the implementations of concurrent systems tend to parallelize operations and to use various optimizations in order to increase the throughput of the system. This leads in general to relaxations in the semantics guaranteed by these implementations w.r.t. to strong consistency models. In this talk, we will address the issue of checking that a given program of the client is robust against this kind of relaxations, i.e., the observable behaviors of the client are the same under both the strong and relaxed consistency models. Robustness corresponds to a correctness criterion that ensures the preservation by the considered relaxations of all properties that can be proved assuming the strong consistency models.

We show that robustness can be checked efficiently in several cases by linear reductions to state reachability problems. These cases include robustness against the weak memory model TSO, and also checking robustness against concurrency and asynchrony in event-driven programs and message passing programs where we compare the behaviors of a same program under two different semantics, one being the asynchronous one, and the other one being a stronger semantics that is synchronous in some sense (that will be defined).

3.2 Visual/interactive design of fault-tolerant distributed algorithms

Paul C. Attie (American University of Beirut, LB)

Joint work of Paul Attie, Kinan Dak Al Bab, Mouhammad Sakr
Main reference Paul C. Attie, Kinan Dak-Al-Bab, Mouhammad Sakr: "Model and Program Repair via SAT Solving", ACM Trans. Embedded Comput. Syst., Vol. 17(2), pp. 32:1–32:25, 2018.
URL http://dx.doi.org/10.1145/3147426

I advocate the design and verification of fault-tolerant distributed algorithms via the direct manipulation of the state-transition relation. To deal with state explosion (in the finite state case), and with combinatoric explosion and infinite states (in the general case), I propose the following:

1. Pairwise composition: analyze the interaction of two processes at a time, to verify safety and liveness properties of process-pairs.
2. Small subsystems: analyze the postcondition of an action (in a small subsystem containing the action) to verify deadlock freedom.
3. Fault actions: model faults as actions which perturb the global state. Synthesize the needed recovery transitions.
4. Automatic repair of transition structures: delete states/transitions which violate a temporal logic specification.
5. Refine atomicity: use knowledge acquired by a process to replace test & set by atomic read/write.
6. Abstraction: equivalence relation on states specifies abstraction. Manipulate abstraction and then concretize.
7. Finitely representable infinite-state structure: a node labeled by a recursive predicate represents a set of states, a transition labeled by a guarded command represents an action.

I have implemented some of these methods, and am currently implementing the remainder, in the Eshmun tool, available at eshmuntool.blogspot.com. The combination of these methods enables rapid semantic feedback and interaction for the distributed algorithm designer. The use of methods to combat complexity is not only for computational reasons, but also for visualization reasons: it helps the designer visualize the behavior of the algorithm.

3.3 Formal Analysis of Population Protocols

Michael Blondin (TU München, DE)

Joint work of Michael Blondin, Javier Esparza, Stefan Jaax, Antonín Kučera
Main reference Michael Blondin, Javier Esparza, Stefan Jaax, Antonín Kučera: "Black Ninjas in the Dark: Formal Analysis of Population Protocols", in Proc. of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018, pp. 1–10, ACM, 2018.
URL http://dx.doi.org/10.1145/3209108.3209110

Population protocols are a model of distributed computation by anonymous mobile agents with little computational power. Such protocols allow for modeling systems such as networks of passively mobile sensors and chemical reaction networks. Agents of a population protocol interact by meeting at random. In well-designed protocols, for every initial configuration of agents and every computation starting from this configuration, all agents eventually agree on a consensus value.

In this talk, I will give an overview of recent advances on the formal analysis of population protocols. In particular, I will discuss the problem of automatically determining whether a protocol is correct, and the problem of computing an asymptotic bound on the expected time a protocol needs to reach consensus.

3.4 Synthesis of Distributed Systems from Logical Specifications

Benedikt Bollig (ENS – Cachan, FR) and Marie Fortin (ENS – Cachan, FR)

Joint work of Benedikt Bollig, Marie Fortin, Paul Gastin
Main reference Benedikt Bollig, Marie Fortin, Paul Gastin: "It Is Easy to Be Wise After the Event: Communicating Finite-State Machines Capture First-Order Logic with 'Happened Before' ", CoRR, Vol. abs/1804.10076, 2018.
URL http://arxiv.org/abs/1804.10076

We are concerned with formally modeling and specifying distributed systems, with the aim of ensuring their correctness. As a system model, we consider communicating finite-state machines (CFMs), in which finite-state processes exchange messages through unbounded FIFO channels. On the specification side, we focus on the first-order logic of message sequence charts (MSCs). MSCs, also known as space-time diagrams, arise naturally as executions of CFMs and feature Lamport's happened-before relation. First-order logic captures many interesting properties of distributed systems, and it subsumes various temporal logics. This presentation consists of two parts:

Part I: Logics over Message Sequence Charts (M. Fortin). In the first part, we study the expressive power of first-order logic, establish connections with temporal logics and propositional dynamic logic, and present a normal-form construction. As a corollary, we establish that first-order logic has the three-variable property.

Part II: From Logic to Communicating Finite-State Machines (B. Bollig). In the second part, we address the synthesis problem: Relying on the normal-form construction of Part I and a (nondeterministic) gossip protocol, we show that every first-order specification can be transformed into a CFM. The latter can then be considered as a system model that is correct by construction. Moreover, the translation is useful in the automata-theoretic approach to model checking distributed systems.

3.5 Automated Fine-Tuning of Probabilistic Self-Stabilizing Algorithms

Borzoo Bonakdarpour (McMaster University – Hamilton, CA)

Although randomized algorithms have widely been used in distributed computing as a means to tackle impossibility results, it is currently unclear what type of randomization leads to the best performance in such algorithms. In this talk, I propose automated techniques to find the probability distribution that achieves minimum average recovery time for an input randomized distributed self-stabilizing protocol without changing the behavior of the algorithm. Our first technique is based on solving symbolic linear algebraic equations in order to identify fastest state reachability in parametric discrete-time Markov chains. The second approach applies parameter synthesis techniques from probabilistic model checking to compute the rational function describing the average recovery time and then uses dedicated solvers to find the optimal parameter valuation. The third approach computes over- and under-approximations of the result for a given parameter region and iteratively refines the regions with minimal recovery time up to the desired precision. The latter approach finds sub-optimal solutions with negligible errors, but it is significantly more scalable in orders of magnitude as compared to the other approaches.

3.6 Tutorial: Distributed Runtime Verification

Borzoo Bonakdarpour (McMaster University – Hamilton, CA)

This tutorial surveys the most prominent works on distributed runtime verification.

3.7 Distributed Property Testing

Keren Censor-Hillel (Technion – Haifa, IL)

This survey talk will overview the recent achievements in the area of distributed property testing.

Background will be given on the computational model, the related distributed decision tasks, and the relaxations that allow overcoming expensive computations in settings of limited bandwidth, within a small number of local queries.

3.8 Parameterized Verification of Topology-sensitive Distributed Protocols

Giorgio Delzanno (University of Genova, IT)

Joint work of Giorgio Delzanno, Sylvain Conchon, Angelo Ferrando
Main reference Sylvain Conchon, Giorgio Delzanno, Angelo Ferrando: "Declarative Parameterized Verification of Topology-Sensitive Distributed Protocols," to appear in NETYS 2018.

We show that Cubicle, an SMT-based infinite-state model checker, can be applied as a verification engine for GLog, a logic-based specification language for topology-sensitive distributed protocols with asynchronous communication. Existential coverability queries in GLog can be translated into verification judgements in Cubicle by encoding relational updates rules as unbounded array transitions. We apply the resulting framework to automatically verify a distributed version of the Dining Philosopher mutual exclusion protocol formulated for an arbitrary number of nodes and communication buffers.

3.9 Communication-closed asynchronous protocols

Cezara Dragoi (ENS – Paris, FR)

Joint work of Cezara Dragoi, Josef Widder

Communication closed round-based models are a particular type of synchronous models that simplify the verification of fault-tolerant distributed systems. We present a sound method to check that an asynchronous protocol is communication closed. The verification conditions

implied by this method can be automatically discarded using of the self SMT-solvers or static analysers. Provided that an asynchronous protocol is communication close we define a code-to-code translation into the Heard-Of computational model, which is a communication closed round-based model.

3.10 Verification of a Fault-Tolerant Cache-Coherency Protocol

Jo Ebergen (Oracle Labs – Redwood Shores, US)

This short presentation tells some of the lessons we learned while verifying a fault-tolerant cache-coherency protocol.

References

1 D. J. Sorin, M. D. Hill, and D. A. Wood. *A Primer on Memory Consistency and Cache Coherence*. Morgan and Claypool Publishers, 2011.

3.11 Distributed Monitoring of Controlled Events

Yuval Emek (Technion – Haifa, IL)

Joint work of Yuval Emek, Amos Korman, Shimon Bitton, Shay Kutten

Monitoring is a fundamental task in many distributed systems. In its most basic form, monitoring is concerned with counting the number of events and detecting when this number reaches some threshold. A good monitoring protocol should run in the background without consuming too many network resources. The challenge in this regard is that the events to be counted may occur in different locations and at unpredicted times.

In this talk, we focus on the task of monitoring *controlled events*, namely, events that actually take place (or *commit*) only after they receive a permit from the monitoring protocol. We will discuss scenarios involving this kind of events and explore the connections between the task of monitoring them and the classic distributed *controller* problem including some recent advances in the study of this problem.

The talk will be self contained.

References

1 Yuval Emek and Amos Korman. *Efficient Threshold Detection in a Distributed Environment*. In Proceedings of the 29th ACM Symposium on Principles of Distributed Computing (PODC), pages 183–191, 2010.
2 Yuval Emek, Amos Korman. *New bounds for the controller problem*. Distributed Computing 24(3-4): 177-186 (2011).
3 Shimon Bitton, Yuval Emek, and Shay Kutten. *Efficient Dispatching of Job Batches in Emerging Clouds*. To appear in Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), 2018.

3.12 Specifying and Verifying Concurrent Objects

Constantin Enea (University Paris-Diderot, FR)

Modern software developments kits simplify the programming of concurrent applications by providing shared state abstractions which encapsulate low-level accesses into higher-level abstract data types (ADTs). Programming such abstractions is however error prone. To minimize synchronization overhead between concurrent ADT invocations, implementors avoid blocking operations like lock acquisition, allowing methods to execute concurrently. However, concurrency risks unintended inter-operation interference, and risks conformance to well-established correctness criteria like linearizability. We present several results concerning the theoretical limits of verifying such concurrent ADTs and testing-based methods for discovering violations in practical implementations.

3.13 Distributed Synthesis

Bernd Finkbeiner (Universität des Saarlandes, DE)

Joint work of Bernd Finkbeiner, Christopher Hahn, Philip Lukert, Marvin Stenger, Leander Tentrup
Main reference Bernd Finkbeiner, Christopher Hahn, Philip Lukert, Marvin Stenger, Leander Tentrup: "Synthesizing Reactive Systems from Hyperproperties", in Proc. of the Computer Aided Verification – 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 10981, pp. 289–306, Springer, 2018.
URL http://dx.doi.org/10.1007/978-3-319-96145-3_16

Distributed synthesis automates the construction of distributed systems. Instead of programming an implementation, the developer writes a formal specification of the desired system properties, for example in a temporal logic. The check whether the specified properties are realizable and the construction of the actual implementation is taken care of by the synthesis algorithm. In this talk, I give an overview on decidability results and algorithms for the two prominent models for distributed synthesis, the Pnueli/Rosner model and the Causal Memory model. The talk concludes with an outlook on the synthesis problem for HyperLTL, a temporal logic for hyperproperties. HyperLTL makes it possible to synthesize distributed systems that additionally satisfy conditions such as symmetric responses, secrecy, and fault tolerance.

3.14 Faithful Delay Models in Circuits and Distributed Systems

Matthias Függer (ENS – Cachan, FR)

Joint work of Matthias Függer, Stephan Friedrichs, Christoph Lenzen, Jürgen Maier, Robert Najvirt, Thomas Nowak, Ulrich Schmid

It is well known that the communication delay model assumed for a distributed system has large impact on the solvability of problems within it. The same is true for signal propagation

delay models in circuits. In the talk we discuss solvability issues for several circuit delay models, and draw the relation to distributed computing models and verification of such systems.

References

1 Matthias Függer, Thomas Nowak, and Ulrich Schmid. *Unfaithful glitch propagation in existing binary circuit models*. ASYNC'13 & IEEE Trans. on Computers'16
2 Matthias Függer, Robert Najvirt, Thomas Nowak, and Ulrich Schmid. *Towards binary circuit models that faithfully capture physical solvability*. DATE'15
3 Robert Najvirt, Matthias Függer, Thomas Nowak, Ulrich Schmid, Michael Hofbauer, and Kurt Schweiger. *Experimental validation of a faithful binary circuit model*. GLSVLSI'15
4 Matthias Függer, Jürgen Maier, Robert Najvirt, Thomas Nowak, and Ulrich Schmid. *A faithful binary circuit model with adversarial noise*. DATE'18
5 Stephan Friedrichs, Matthias Függer, and Christoph Lenzen. *Metastability-Containing Circuits*. IEEE Trans. on Computers'18

3.15 Indistinguishability: Friend and Foe in Concurrent Programming

Hagit Attiya (Technion – Haifa, IL)

Joint work of Hagit Attiya, Ramalingam, Noam Rinetzky, Rachid Guerraoui, Danny Hendler, Peter Kuznetsov, Maged Michael, Martin Vechev, Sandeep Hans, Alexey Gotsman

Uncertainty about the global state is a major obstacle for achieving synchronization in concurrent systems. Formally, uncertainly is captured by showing that a process cannot distinguish two different global states. Indistiguishability arguments play a key role in many lower bounds for concurrent data structures, one of them, on the need for memory barriers, will be presented in this talk. Surprisingly, however, indistiguishability can also help in the verification of concurrent data structures, as demonstrated by a reduction theorem we will describe, or in understanding their specification, as we will show in the context of transactional memory.

(Overview talk.)

3.16 Cutoff Results for Parameterized Verification and Synthesis

Swen Jacobs (Universität des Saarlandes, DE)

Joint work of Simon Außerlechner, Swen Jacobs, Ayrat Khalimov, Mouhammad Sakr
Main reference Swen Jacobs, Mouhammad Sakr: "Analyzing Guarded Protocols: Better Cutoffs, More Systems, More Expressivity", in Proc. of the Verification, Model Checking, and Abstract Interpretation – 19th International Conference, VMCAI 2018, Los Angeles, CA, USA, January 7-9, 2018, Proceedings, Lecture Notes in Computer Science, Vol. 10747, pp. 247–268, Springer, 2018.
URL http://dx.doi.org/10.1007/978-3-319-73721-8_12

In this talk, I highlight some of the principles and challenges of the cutoff-based approach to the verification and synthesis of systems of parametric size. I give an overview of some of our recent results that tackle these challenges, specifically in the framework of guarded protocols. Finally, I talk about our ongoing work on extensions of these techniques.

References

1 E. Allen Emerson and Vineet Kahlon. *Reducing Model Checking of the Many to the Few*. CADE 2000.
2 Swen Jacobs and Roderick Bloem. *Parameterized Synthesis*. Logical Methods in Computer Science 10(1), 2014.
3 Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, Josef Widder. *Decidability of Parameterized Verification*. Synthesis Lectures on Distributed Computing Theory, Morgan & Claypool Publishers, 2015.
4 Simon Außerlechner, Swen Jacobs and Ayrat Khalimov. *Tight Cutoffs for Guarded Protocols with Fairness*. VMCAI 2016.
5 Swen Jacobs and Mouhammad Sakr. *Analyzing Guarded Protocols: Better Cutoffs, More Systems, More Expressivity*. VMCAI 2018.

3.17 What my computer can find about your distributed algorithm

Igor Konnov (INRIA Nancy – Grand Est, FR)

Joint work of Igor Konnov, Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Marijan Lazic, Sasha Rubin, Helmut Veith, Josef Widder
Main reference Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, Josef Widder: "Decidability of Parameterized Verification", Morgan & Claypool Publishers, 2015.
URL http://dx.doi.org/10.2200/S00658ED1V01Y201508DCT013

Parameterized model checking is an active research field that addresses automated verification of distributed or concurrent systems, for all numbers of participating processes. The system models that are studied in this field are inspired by those from distributed computing. In this talk, I summarize the prominent techniques for parameterized model checking. Starting with the first undecidability results. Continuing with techniques such as cut-off proofs and abstraction. Finishing with our recent results on verification of threshold-guarded distributed algorithms.

Based on joint work with Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Marijana Lazic, Sasha Rubin, Helmut Veith, and Josef Widder.

References

1 Igor Konnov, Marijana Lazić, Helmut Veith, Josef Widder. *A short counterexample property for safety and liveness verification of fault-tolerant distributed algorithms*. In: POPL. pp. 719–734 (2017).

3.18 Synthesizing Thresholds for Fault-Tolerant Distributed Algorithms

Marijana Lazic (TU Wien, AT)

Joint work of Marijana Lazic, Igor Konnov, Josef Widder, Roderick Bloem
Main reference Marijana Lazic, Igor Konnov, Josef Widder, Roderick Bloem: "Synthesis of Distributed Algorithms with Parameterized Threshold Guards", in Proc. of the 21st International Conference on Principles of Distributed Systems, OPODIS 2017, Lisbon, Portugal, December 18-20, 2017, LIPIcs, Vol. 95, pp. 32:1–32:20, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.OPODIS.2017.32

We focus on threshold-based distributed algorithms, where a process has to wait until the number of messages it receives reaches a certain threshold, in order to perform an action. Examples of such distributed algorithms include fault-tolerant broadcast, non-blocking atomic commitment, and consensus. I present an automated method for synthesizing these thresholds, given a sketch of a distributed algorithm and specifications. In this way we synthesize distributed algorithms that are correct for every number n of processes and every number t of faults, provided some resilience condition holds, e.g. $n > 3t$.

3.19 Breaking and (Partly) Fixing Pastry

Stephan Merz (INRIA Nancy – Grand Est, FR)

Joint work of Stephan Merz, Noran Azmy, Tianxiang Lu, Christoph Weidenbach
Main reference Noran Azmy, Stephan Merz, Christoph Weidenbach: "A machine-checked correctness proof for Pastry", Sci. Comput. Program., Vol. 158, pp. 64–80, 2018.
URL http://dx.doi.org/10.1016/j.scico.2017.08.003

Pastry [1] is a well-known algorithm for maintaining a distributed hash table over a peer-to-peer overlay network. A key correctness requirement is that the algorithm must ensure a sufficiently consistent view among the participating nodes of which nodes are live members of the network, in the absence of centralized control. In particular, this is necessary for requests to be routed to the intended destination. This property represents an interesting target for formal verification.

We analyzed formal models of Pastry using the TLA$^+$ model checker and identified problems in the different published versions of the algorithm that can lead to unrepairable loss of connectivity among the nodes in the Pastry ring, even in the absence of spontaneous node departures. Identifying the root cause of the problem, we suggest a variant of the algorithm and formally prove, using the TLA$^+$ proof system, that it ensures that requests are routed correctly, assuming that nodes do not fail [2]. We do not know to what extent our Pastry variant is robust to spontaneous node departures.

References

1 Antony I. T. Rowstron, Peter Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. Middleware 2001, pp. 329-350 (2001).
2 Noran Azmy, Stephan Merz, Christoph Weidenbach. A Machine-Checked Correctness Proof for Pastry. Sci. Comput. Program. 158, pp. 64-80 (2018).

3.20 Indistinguishability, Duality, and Coordination

Yoram Moses (Technion – Haifa, IL)

Indistinguishability is a fundamental notion in distributed systems. It serves as the central tool in impossibility proofs and lower bounds. Indeed, indistinguishability can be used to determine when actions are disallowed. Its dual, which corresponds to the knowledge that a process has, plays the opposite role, and determines when actions are allowed. This talk will discuss the relation between knowledge and action in distributed systems, and present several theorems that apply across all models of distributed computation. The connections drawn also relate a semantic approach, which can be viewed in terms a modal logic, and algorithmic issues.

References

1 Yoram Moses. *Relating knowledge and coordinated action: The knowledge of preconditions principle*. Proc. of TARK 2015, arXiv preprint arXiv:1606.07525, 2016 – arxiv.org.
2 Armando Castañeda, Yannai A. Gonczarowski, Yoram Moses. *Unbeatable consensus*. Proc. of DISC 2014.

3.21 Interactive Distributed Proofs

Rotem Oshman (Tel Aviv University, IL)

Interactive proof systems allow a resource-bounded verifier to decide an intractable language (or compute a hard function) by communicating with a powerful but untrusted prover. Such systems guarantee that the prover can only convince the verifier of true statements. In the context of centralized computation, a celebrated result shows that interactive proofs are extremely powerful, allowing polynomial-time verifiers to decide any language in PSPACE.

In this work we initiate the study of distributed interactive proofs: a network of nodes interacts with a single untrusted prover, who sees the entire network graph, to decide whether the graph satisfies some property. We focus on the communication cost of the protocol — the number of bits the nodes must exchange with the prover and each other. Our model can also be viewed as a generalization of the various models of "distributed NP" (proof labeling schemes, etc.) which received significant attention recently: while these models only allow the prover to present each network node with a string of advice, our model allows for back-and-forth interaction. We prove both upper and lower bounds for the new model. We show that for some problems, interaction can exponentially decrease the communication cost compared to a non-interactive prover, but on the other hand, some problems retain non-trivial cost even with interaction.

3.22 Proof-Labeling Schemes: Broadcast, Unicast and In Between

Mor Perry (Tel Aviv University, IL)

Joint work of Mor Perry, Boaz Patt-Shamir
Main reference Boaz Patt-Shamir, Mor Perry: "Proof-Labeling Schemes: Broadcast, Unicast and in Between", in Proc. of the Stabilization, Safety, and Security of Distributed Systems – 19th International Symposium, SSS 2017, Boston, MA, USA, November 5-8, 2017, Proceedings, Lecture Notes in Computer Science, Vol. 10616, pp. 1–17, Springer, 2017.
URL http://dx.doi.org/10.1007/978-3-319-69084-1_1

We study the effect of limiting the number of different messages a node can transmit simultaneously on the verification complexity of proof-labeling schemes (PLS). In a PLS, each node is given a label, and the goal is to verify, by exchanging messages over each link in each direction, that a certain global predicate is satisfied by the system configuration. We consider a single parameter r that bounds the number of distinct messages that can be sent concurrently by any node: in the case $r = 1$, each node may only send the same message to all its neighbors (the broadcast model), in the case r is at least Δ, where Δ is the largest node degree in the system, each neighbor may be sent a distinct message (the unicast model), and in general, for r between 1 and Δ, each of the r messages is destined to a subset of the neighbors.

We show that message compression linear in r is possible for verifying fundamental problems such as the agreement between edge endpoints on the edge state. Some problems, including verification of maximal matching, exhibit a large gap in complexity between $r = 1$ and $r > 1$. For some other important predicates, the verification complexity is insensitive to r, e.g., the question whether a subset of edges constitutes a spanning-tree. We also consider the congested clique model. We show that the crossing technique for proving lower bounds on the verification complexity can be applied in the case of congested clique only if $r = 1$. Together with a new upper bound, this allows us to determine the verification complexity of MST in the broadcast clique.

3.23 Pretend Synchrony- some distributed computing approaches

Sergio Rajsbaum (National Autonomous University of Mexico, MX)

Joint work of Srgio Rajsbaum, Eli Gafni, Maurice Herlihy, Yoram Moses, Michel Raynal

Pretend Synchrony is the title of a recent talk at VDS in Essaouira, Morocco 2018 by Ranjit Jhala where a restricted computational model is shown to be sufficient to verify correctness assertions for several distributed problems. In addition to Ranjit, others discussed related approaches at VDS, including Josef Widder, Cezara Dragoi, Bernhard Kragl and Ahmed Bouajjani, sometimes emphasizing the importance of the classic Communication-Closed Layers paradigm of Elrad and Frances. Motivated by these works, I will describe some of the research (not as recent, some dating back to 1998) which we have done on pretending synchrony from the distributed computing perspective, in the hope that this topics serves as a good point for exchanging ideas between the verification and distributed computing communities. I will discuss work on layering analysis for consensus, generalizations to other problems using topology [1], and iterated models together with recursive distributed algorithms [3, 4].

References

1 Maurice Herlihy, Dmitry N. Kozlov, Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann 2013.
2 Yoram Moses, Sergio Rajsbaum. *A Layered Analysis of Consensus*. SIAM J. Comput. 31(4):989–1021 (2002).
3 Sergio Rajsbaum. *Iterated Shared Memory Models*. LATIN 2010, Lecture Notes in Computer Science 6034, Springer 2010.
4 Sergio Rajsbaum, Michel Raynal. *An Introductory Tutorial to Concurrency-Related Distributed Recursion*. Bulletin of the EATCS 111 (2013).

3.24 Biased Clocks: A way to Improve Effectiveness of Run Time Monitoring of Distributed Systems

Sandeep S. Kulkarni (Michigan State University – East Lansing, US)

Joint work of Sandeep S. Kulkarni, Vidhya Tekken Valapil
Main reference Vidhya Tekken Valapil and Sandeep S. Kulkarni, "Biased Clocks: A Novel Approach to Improve the Ability to Perform Predicate Detection with O(1) Clocks", SIROCCO 2018.

Runtime Monitoring of distributed systems requires $O(n)$ sized timestamps given that events in a system cannot be partitioned into a total order. $O(n)$ sized timestamps severely limit the ability to utilize them in practice. $O(1)$ sized timestamps such as logical clocks or hybrid logical clocks can be used for runtime monitoring. However, they miss several instances where the property of interest is violated but the violation is not detected. We propose a new type of clocks, biased clocks, that improve the effectiveness of clocks in monitoring. Biased clocks treat local events on a process differently than messages. In particular, by adding a bias to the timestamp received in a message, we show that it substantially improves the ability to detect violations of desired system properties.

3.25 Playing with scheduling policies

Arnaud Sangnier (University Paris-Diderot, FR)

Joint work of Arnaud Sangnier, Carole Delporte-Galler, Hugues Fauconnier, Yann Jurski and François Laroussinie

In order to develop distributed algorithms, assumptions are made on their execution context: will the entities behave synchronously or in an asynchronous way, will the entities execution be scheduled in a round-robin way or will its order be completely non-deterministic, will the entities crash, will they be dependent one from each other or should they be able to run the algorithm independently, etc.

As a matter of fact, some tasks may be achieved in some executions contexts and changing an hypothesis on these contexts may lead to impossibility results. For instance, consensus cannot be achieved with 2 processes running a wait-free algorithm on a shared memory system, but this task is feasible when considering obstruction-free algorithms. One difficulty is however to find a formal way to define executive contexts. In this talk, I will present a

recent approach which consists in using automata to represent some executive contexts for shared memory systems and two-player games to detect the possibility or impossibility of achieving consensus in such contexts.

3.26 Linearizability via Order-extension Results

Ana Sokolova (Universität Salzburg, AT)

Joint work of Ana Sokolova, Harald Woracek

The semantics of concurrent data structures is usually given by a sequential specification and a consistency condition. Linearizability is the most popular consistency condition due to its simplicity and general applicability. Verifying linearizability is a difficult, in general undecidable, problem.

In this talk, I will discuss the semantics of concurrent data structures and (1) give an overview of work done on this topic by myself and a group of coauthors, as well as (2) present recent order extension results (joint work with Harald Woracek) that lead to characterizations of linearizability in terms of violations, a.k.a. aspects. The approach works for pools, queues, and priority queues; finding other applications is ongoing work. In the case of pools and queues we obtain already known characterizations, but the proof method is new, elegant, and simple, and we expect that it will lead to deeper understanding of linearizability.

3.27 Model checking of incomplete systems

Paola Spoletini (Kennesaw State University – Marietta, US)

Joint work of Paola Spoletini, Claudio Menghi, Carlo Ghezzi, Marsha Chechik, Anna Bernasconi, Lenore Zuck
Main reference Claudio Menghi, Paola Spoletini, Carlo Ghezzi: "Dealing with Incompleteness in Automata-Based Model Checking", in Proc. of the FM 2016: Formal Methods – 21st International Symposium, Limassol, Cyprus, November 9-11, 2016, Proceedings, Lecture Notes in Computer Science, Vol. 9995, pp. 531–550, 2016.
URL http://dx.doi.org/10.1007/978-3-319-48989-6_32

Incomplete models [3] describe the behavior of systems where some components or functionalities are still unspecified. These models can be used in different scenarios; examples are (1) analysis the trade-offs among alternative solutions for the unspecified parts, (2) development of component-based and distributed systems. Classic model checking assumes that a complete model of the system is available and does not support the verification of incomplete models. This is an obstacle to early detection of design errors since in early phases of the system design models are often incomplete.

In this talk, I present a novel automata-based model checking approach that supports verification of incomplete models. I explore two complementary solutions for handling cases in which the satisfaction of a given property depends on the yet unspecified parts of the model.

The first solution enables the computation of constraints that must be satisfied by future replacements of the unspecified components to guarantee the satisfaction of the given property. The satisfaction of these constraints by the replacements of the unspecified components,

that can be checked in isolation, ensures the fulfilment of the property of interest [3]. This approach can be complemented with a framework [1] that helps developers understanding why a property of interest is satisfied or "possibly" satisfied (i.e., its satisfaction depends on unknown parts) by enriching the model checker outcome with a proof of satisfaction or "possibly" satisfaction in these cases.

While the presented approach was developed to deal with incomplete systems, it could be also used to distribute the complexity of the verification of very large systems. This may be obtained though an iterative decomposition of the system into smaller parts that are encapsulated into unspecified components.

The second solution is based on a framework that supports (1) incompleteness through a formal specification of pre- and post-conditions and (2) independent development, reuse of off-the-shelf components, synthesis and verification of sub-components [2].

References

1 Anna Bernasconi, Claudio Menghi, Paola Spoletini, Lenore D. Zuck, Carlo Ghezzi. *From Model Checking to a Temporal Proof for Partial Models*. SEFM 2017:54–69
2 Claudio Menghi, Paola Spoletini, Marsha Chechik, Carlo Ghezzi. *Supporting Verification-Driven Incremental Distributed Design of Components*. FASE 2018:169–188
3 Claudio Menghi, Paola Spoletini, Carlo Ghezzi. *Dealing with Incompleteness in Automata-Based Model Checking*. FM 2016:531–550

3.28 Distributed Encoding of the Integers

Corentin Travers (University of Bordeaux, FR)

Joint work of Corentin Travers, Pierre Fraigniaud, Sergio Rajsbaum, Petr Kuznetsov, Thibault Rieutord
Main reference Pierre Fraigniaud, Sergio Rajsbaum, Corentin Travers: "Minimizing the Number of Opinions for Fault-Tolerant Distributed Decision Using Well-Quasi Orderings", in Proc. of the LATIN 2016: Theoretical Informatics – 12th Latin American Symposium, Ensenada, Mexico, April 11-15, 2016, Proceedings, Lecture Notes in Computer Science, Vol. 9644, pp. 497–508, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-662-49529-2_37

A distributed encoding of the integer is a distributed structure that encodes each positive integer n with a word w of length n over some (non-necessarily finite) alphabet A, such that any for any $n' < n$, any subword w' of w of length n' is not the distributed code of n'. Relying on well-quasi ordre theory, we show that the first N integers can be distributedly encoded using words on an alphabet with letters on $O(\log(\alpha(n)))$ bits, where α is a function growing at least as slowly as the inverse-Ackerman function.

We then show that distributed encoding of the integers can be applied in failure prone distributed systems to build failure detector outputting very few bits and to construct short certificate for distributed decision.

References

1 Pierre Fraigniaud, Sergio Rajsbaum, Corentin Travers, Petr Kuznetsov, Thibault Rieutord. *Perfect Failure Detection with Very Few Bits*. SSS 2016:154–169.

3.29 Towards verification of distributed algorithms in the Heard-of model

Igor Walukiewicz (University of Bordeaux, FR)

Joint work of Anca Muscholl, Corentin Travers, Igor Walukiewicz

We consider algorithms in the Heard-of model of distributed computation proposed by Charron-Bost and Schiper in 2009. We aim at verifying automatically if a given algorithm solves the consensus problem. In order to state the problem formally we need to fix what operations can algorithms perform. We propose to consider operations that are definable by existentially quantified linear inequalities. We call such algorithms tame. We show that even for tame algorithms the problem is undecidable. Then we present two decidable special cases. One when algorithms use only two values. The other is based on a short run property. We show that every run is equivalent to a short run if the algorithm has what we call stability property.

3.30 (Strong) Linearizability – A Tutorial

Philipp Woelfel (University of Calgary, CA)

This is a tutorial on linearizability, the gold standard of correctness conditions for shared memory algorithms. The first part of the talk will cover necessary definitions, examples, properties, and why linearizability is so important. The second part of the talk will show why linearizability is not enough for randomized algorithms, and will introduce a stronger correctness condition, strong linearizability, that resolves the issues with linearizability in certain randomized models.

Participants

- Paul C. Attie
American University of Beirut, LB
- Hagit Attiya
Technion – Haifa, IL
- A. R. Balasubramanian
Chennai Mathematical Institute, IN
- Michael Blondin
TU München, DE
- Benedikt Bollig
ENS – Cachan, FR
- Borzoo Bonakdarpour
McMaster University – Hamilton, CA
- Ahmed Bouajjani
University Paris-Diderot, FR
- Dan Brownstein
Ben Gurion University – Beer Sheva, IL
- Keren Censor-Hillel
Technion – Haifa, IL
- Aiswarya Cyriac
Chennai Mathematical Institute, IN
- Giorgio Delzanno
University of Genova, IT
- Cezara Dragoi
ENS – Paris, FR
- Jo Ebergen
Oracle Labs – Redwood Shores, US
- Yuval Emek
Technion – Haifa, IL
- Constantin Enea
University Paris-Diderot, FR
- Javier Esparza
TU München, DE
- Bernd Finkbeiner
Universität des Saarlandes, DE
- Marie Fortin
ENS – Cachan, FR
- Pierre Fraigniaud
University Paris-Diderot and CNRS, FR
- Matthias Függer
ENS – Cachan, FR
- Paul Gastin
ENS – Cachan, FR
- Swen Jacobs
Universität des Saarlandes, DE
- Igor Konnov
INRIA Nancy – Grand Est, FR
- Sandeep Kulkarni
Michigan State University – East Lansing, US
- Marijana Lazic
TU Wien, AT
- Jérémy Ledent
Ecole Polytechnique – Palaiseau, FR
- Martin Leucker
Universität Lübeck, DE
- Stephan Merz
INRIA Nancy – Grand Est, FR
- Roland Meyer
TU Braunschweig, DE
- Yoram Moses
Technion – Haifa, IL
- Anca Muscholl
University of Bordeaux, FR
- Rotem Oshman
Tel Aviv University, IL
- Mor Perry
Tel Aviv University, IL
- Sergio Rajsbaum
National Autonomous University of Mexico, MX
- David A. Rosenblueth
National Autonomous University of Mexico, MX
- Arnaud Sangnier
University Paris-Diderot, FR
- Ana Sokolova
Universität Salzburg, AT
- Paola Spoletini
Kennesaw State University – Marietta, US
- Corentin Travers
University of Bordeaux, FR
- Igor Walukiewicz
University of Bordeaux, FR
- Philipp Woelfel
University of Calgary, CA

Report from Dagstuhl Seminar 18212

On-Body Interaction: Embodied Cognition Meets Sensor/Actuator Engineering to Design New Interfaces

Edited by

Kasper Hornbaek[1], David Kirsh[2], Joseph A. Paradiso[3], and Jürgen Steimle[4]

1 **University of Copenhagen, DK,** kash@di.ku.dk
2 **University of California – San Diego, US,** dkirsh@gmail.com
3 **MIT – Cambridge, US,** joep@media.mit.edu
4 **Universität des Saarlandes, DE,** steimle@cs.uni-saarland.de

Abstract

On-body technologies are emerging as a new paradigm in human-computer interaction. Instead of moving a mouse or tapping a touch surface, people can use whole-body movements to navigate in games, gesture in mid-air to interact with large displays, or touch their forearm to control a mobile phone. First promising applications are being investigated or have been demonstrated in mobile computing, healthcare, or sports.

Two areas of research have been contributing to this paradigm. Research on embodied cognition suggests that the body should no longer be treated as a passive actuator of input devices but as something that needs to be carefully designed for and as something that offers unique new possibilities in interaction. Embodied cognition has become a prominent candidate for outlining what we can and cannot do in on-body interaction. Research on interactive technologies for the body is opening up new avenues for human-computer interaction, by contributing body-based sensing input and output modalities with more body compatible form factors. Together, these areas allow the design and implementation of new user interfaces; however, they are rarely in direct contact with each other.

The intended outcome of the seminar was a research agenda for on-body technologies based on synergies between these two views. We therefore brought together a group of researchers from embodied cognition (including psychology, robotics, human-computer interaction, and sociology) as well as sensor/actuator engineering (including computer science, materials science, electrical engineering). These groups worked together toward outlining a research agenda for on-body technologies, in part using a bottom-up process at the seminar, in part using structured answers to questions in advance of the seminar. Key topics for discussion included (1) advances in on-body sensors and actuators, in particular how to drive the technical development from work on embodied cognition and the body, (2) cognitive consequences of on-body technologies, (3) how to take the peculiarities and possibilities of the body into consideration, (4) how to evaluate on-body technology, and (5) application areas of on-body technologies.

Seminar May 21–24, 2018 – http://www.dagstuhl.de/18212

2012 ACM Subject Classification Human-centered computing → Interaction paradigms, Human-centered computing → Interaction devices, Human-centered computing → HCI theory, concepts and models, Human-centered computing → Interaction techniques

Keywords and phrases Human-Computer interaction, Embodied cognition, User interface software and technology

Digital Object Identifier 10.4230/DagRep.8.5.80

On-Body Interaction: Embodied Cognition Meets Sensor/Actuator Engineering to Design New Interfaces, *Dagstuhl Reports*, Vol. 8, Issue 05, pp. 80–101
Editors: Kasper Hornbaek, David Kirsh, Joseph A. Paradiso, and Jürgen Steimle

Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Kasper Hornbaek
David Kirsh
Joseph A. Paradiso
Jürgen Steimle

Motivation

For the past 40 years, input to computers has been given with mouse and keyboard. Over the last decade, multi-touch has become popular for small devices (e.g., phones and tablets) as well as for large displays (e.g., interactive tabletops and wall-sized screens). All these forms of input require the user to hold or touch a device. Conversely, output has happened on large screens external to the body (e.g., a desktop) or small ones on the body (e.g., smartwatches). The field of human-computer interaction (HCI) has worked to understand these user interfaces (UIs) and how people use them, in addition to establishing principles of design and models of performance to help design them so that they are useful and usable.

Recently, however, HCI researchers have been interested in allowing new forms of on-body technologies. One vision is to integrate technology with the body so as to use and supplement its capabilities. In particular, researchers have focused on sensing users' movement and gestures, aiming to allow users to interact using their body rather than by using a device. Early work included Bolt's put-that-there system developed in the late 1970s, and recent advances in computer vision have allowed the tracking of users' hands, arms, and bodies, leading to a flurry of motion-based gaming controls and inventive, body-based games. The number and variety of research prototypes of non-device UIs have also exploded over the past few years, showing how movements in front of a large display can control navigation, how users can gesture in mid-air, how scratching or poking the skin of one's forearm can be a means of input, and how electric muscle stimulation can be used to move users' limbs as output. Further, HCI researchers have been exploring the theoretical opportunities in using the body for interaction, describing principles for whole-body interaction, embodied interaction, and body-centric interaction, as well as highlighting some of the philosophical and psychological challenges associated with using the body as an interface. First promising applications are being investigated or have been demonstrated in mobile computing, healthcare, or sports. A new UI paradigm seems to be emerging.

The main objective of the seminar was to explore on-body interaction through two research areas: embodied cognition and sensor/actuator engineering. The former has driven a lot of thinking and models around on-body technologies and the potential of body-based interaction. The latter has been behind many of the sensors and actuators that have enabled prototypes to be built and to demonstrate the potential of on-body technology. We did this bringing together a group of researchers from embodied cognition (including psychology, robotics, human-computer interaction, art/design, and sociology) as well as sensor/actuator engineering (including computer science, materials science, electrical engineering). Second, we had this diverse group of researchers outline a research agenda for on-body technologies, in part using a bottom-up process at the seminar, in part using structured answers to questions in advance of the seminar.

Topics

In line with the objectives above, the seminar focused on three areas of investigation:

- **Embodied Cognition**: Embodied cognition is a term covering research in linguistics, robotics, artificial intelligence, philosophy, and psychology (e.g., Anderson 2003, Wilson 2002). The core idea in embodied cognition is that our bodies shape thinking broadly understood (including reasoning, memory, and emotion). In contrast to most psychological foundations of HCI, embodied cognition argues that one cannot study the human as a system comprising input (senses), processing (thinking), and output (motor activity), because sensor-motor activity affects thinking fundamentally and, conversely but less radically, because our body reflects more about our thinking than is commonly expected. Thus, bodies and thinking are intertwined, as reflected in embodied cognition book titles like "How the Body Shapes the Way We Think" [2] and "How the Body Shapes the Mind" [1]. Embodied cognition has become a prominent candidate for outlining what we can and cannot do in on-body interaction.
- **Sensor/Actuator Engineering**: The engineering of technologies that transform the human body into an interface is a very active research area. A widely used approach uses techniques from visual computing for capturing body gestures and touch input on the body using RGB or depth cameras, while projecting visual output with a body-worn projector. Other approaches build on the transdermal propagation of ultrasound or electromagnetic waves to identify the location of touch contact on human skin. EMG can be used to capture human muscle movement, while Electrical Muscle Stimulation can generate muscle output. Radar is another technology that has been successfully demonstrated very recently for capturing gestural input. A further recent strand in research uses slim skin electronics for sensing and output on the body. These technologies are opening up new avenues for human-computer interaction, by contributing body-based sensing input and output modalities with an increasing resolution and more body compatible form factors.
- **New On-Body Technologies**: This area concerns how we can combine embodied cognition and sensor/actuator engineering to design on-body technologies. The design of on-body technologies was a key discussion topic, in particular, how to drive the technical development from work on embodied cognition and the body, how to evaluate on-body technology, and how to take the peculiarities and possibilities of the body into consideration. The application areas of on-body technologies were another consideration.

Activities

The first day of the seminar was reserved for presentations, to establish common ground for discussions. All participants introduced themselves, their background, and their vision in short position talks.

Four long talks reviewed the state-of-the-art and presented recent work in key areas. In his talk "Embodied Cognition: What does having a body gives us?", David Kirsh emphasized on four topics: Effectivity, Enactive perception, Interactive Cognition, and Experience. They all explore what having a body gives us that goes beyond just having a sensor in space. Katia Vega's talk, entitled "Beauty Technologies", focused on the possibilities to embed technology on and inside the skin. Nadia Bianchi-Berthouze gave a talk entitled "The Affective Body in Interaction", discussing the high-level principles of affective computing and creating body-affective-aware-computing technology, which involves sensing the affect and emotion of the users and using them for interaction. In his talk "Cosmetic Computing: Actions and

Figure 1 Demo session featuring latest body-based technologies, held in the historical Music Hall of Dagstuhl castle.

Urgencies towards an Inclusive, Equitable Landscape of On-Body Technologies", Eric Paulos urged the need for transdisciplinary and interdisciplinary approaches and proposed a framing around "Cosmetic Computing".

The evening featured a demo session. An impressive total number of 8 interactive demos and exhibits were demonstrated in the historical ambiance of the Rokoko-style music hall. Those demos comprised, amongst other, e-textiles, interactive tattoos and make-up, new bio-inspired materials and tactile actuation technologies.

The second day consisted of work in **breakout groups**. First, groups identified **challenges** for future work in the field of on-body interaction, grouped into four main areas: Integration of the body and the device; Cognition and Affect; Interaction; and Applications. Next, the participants worked together to identify **positive visions** of a future with body-based interfaces. Promising aspects that were identified include sensory augmentation of human body for graceful ageing, personalized medication and the idea of legal/democratic framework for controlling wearable technology.. To identify potential risks associated with body-based technologies and interaction, the group also developed **negative visions**. Key problems and risks that were identified include a loss of physical embodiment and substantial security risks of our bodies (and potentially even emotions) being externally controlled.

In an session, entitled academic speed-dating, we randomly paired two participants with each other. Their goal was to developed within 7 minutes an idea and a title for a paper they would write together. The format turned out to be very well-received and to stimulate research ideas at unforeseen intersections between the participants' interest and expertise.

Conclusion

The seminar set out to bring together diverse researchers to discuss the overlap between embodied cognition and sensor/actuator engineering. The group managed to cover advances in on-body sensors and actuator, some of the cognitive consequences of on-body technologies, and open issues in applications of on-body technologies. Further, a range of open questions and exciting research questions were discussed, which will likely foster future collaboration and serve as a generator of future research on on-body technologies.

References

1 Shaun Gallagher. *How the body shapes the mind*. Clarendon Press, 2006.
2 Rolf Pfeifer and Josh Bongard. *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006.

2 Table of Contents

3 Overview of Talks

3.1 Haptically-enhanced Body Interaction

Liwei Chan (National Chiao-Tung University – Hsinchu, TW)

Body-based interaction should rely on the body, not vision and hearing though also not excluding them. The key challenge is to enable interaction loops centering haptic channel, if not alone. I consider the bandwidth of body-based interaction is limited by the imprecision of haptic sensation that can hider the user's input capability. Can we boost input capability with artificial haptic output that enhances our awareness of body motion?

3.2 Toward Extended Intelligence in Connected Environments

Clément Duhart (MIT – Cambridge, US)

Extended Intelligence is a new field introduced at the MIT Media Lab by Joi Ito which considers new cognitive dimensions as a fundamentally distributed phenomenon between artificial and biological intelligences. In this talk, we present an early stage of experimentation in which Hear There an auditive augmentation device with a visual attention mechanism interacts with Tidzam, a deep learning acoustic scene analyzer able also to classify wildlife sound like bird species. In such scenario, Hear There is able to detect where the user sight is oriented and so, able to play an audio stream from microphones close to the its region of interest which gives a kind of auditive super power. When combining with Tidzam which can detect that there is, for example, a canada goose, Hear There can play additional audio streams close to the right or left ear in order to invite the user to turn his head in another direction because there is other canada goose in his back for example. Depending of the user reaction, such system can refine his inference about user interest or learn more the acoustic scene. In this example, the intelligence is not in a particular system but more in their interaction in which the user is at the same time the subject and the actor.

3.3 Textile Interfaces

Michael Haller (University of Applied Sciences Upper Austria, AT)

The overall goal of this talk to show the possibilities for designing and creating an interactive knitted/woven textile sensor capable of sensing touch gestures and deformation input in real-time. Knitted/Woven Wearables combine tactile pressure sensitivity with conventional wearables, leading to an "Imperceptible Wearable Textile Interface". Its sensing capabilities enable the detection of pressure and deformation, and thus expands the potential gesture interaction space and possibilities for novel forms of expression. We propose a generic textile sensing platform, which includes the whole value chain ranging from material research and textile fabrication to hardware and software.

3.4 Textiles as Skin

Nur Hamdan (RWTH Aachen, DE)

In this talk, I propose textiles, garments, as an additional skin layer; one that can be digitally augmented to sense and actuate, and leverage our sense of proprioception. I describe two applications for textile-based on-body interaction. The first is a smart training shirt that enables runners to trigger actions by tapping at different locations on their upper body. The second is a seat cover with embedded vibration motors, specifically designed to send subtle messages to drivers to encourage mindful physical practices and breathing during a commute. I briefly demonstrate embroidered textiles sensors that can detect hover, touch, pressure, and fold. For actuation, I show how shape memory alloys can be incorporated in small monolithic structures and create rich tactile feedback on the skin, such as tap, stroke, twist, stretch, scratch, and pinch. I end the talk with a proof of concept prototype that proposers how textiles can add another dimension to our skin: sound.

3.5 Devices in, through or underneath the skin: Insertables

Kayla J. Heffernan (The University of Melbourne, AU)

The human body has emerged as a platform for devices—both for wearable wellbeing devices, and implantable medical devices (IMDs). IMDs include pacemakers, cochlear implants, deep brain stimulation for the treatment of Multiple Sclerosis and Parkinson's Disease, dental implants, orthodontics and implantable contraceptive to name only a few. Technological size and cost reductions, along with power and battery improvements, has seen items that were once strictly external become wearable, and even insertable. Instead of placing a device on the body when needed, and taking it off again when no longer required, it is now possible to augment the body in a semi-permanent way with an insertable device. This augmentation is typically not visible to others and is comparable to those who insert contact lenses rather than wearing glasses. In recent years, we have seen the emergence of non-life-threatening health products becoming insertable, such as female intrauterine devices (IUD) and sub-dermal contraceptive implants. As individuals become more comfortable with devices inside the body, as well as body modifications, we are beginning to see voluntary use of insertable devices outside of the health sphere. We define insertables as objects that go in, through, or underneath the skin. Our choice of the word 'insertable', over 'implantable', for these devices is deliberate. Implantable is used in the medical context to refer to an object fixed inside a person's body by surgery. Therefore, implantables are more difficult, if not impossible, to remove while insertables can be inserted and removed with minimal invasiveness. An implant is often something done to a person out of need, whereas an insertable implies a strong sense of personal agency and choice. Insertables are differentiated by their voluntary and non-medical nature. The arena of insertables has received little academic attention, particularly in the field of human-computer interaction (HCI). This project focuses on understanding the emerging field of insertable devices, looking as what devices people are putting into their bodies and why, classifying public opinions and propensity to insertables, and understanding

how to design and develop for them. It will provide an understanding of the current state of insertables, and compare and contrast their design and development to implantable devices to identify why insertables are different. This knowledge will inform future use and design and position insertables as a device mode of choice for users and a legitimate category for hardware manufactures, HCI researchers and interaction designers alike.

3.6 Continuous Physiological Sensing for On-Body Interfaces

Christian Holz (Microsoft Research – Redmond, US)

Current interactive technology strives towards better understanding users and their contexts. I'm proposing continuously monitoring the user's physiological signals to get a sense of their state, using mobile and convenient form factors. I demonstrate how this work impacts the future of holistic and preventive healthcare in the wild as well as its implications for technology on modern touch systems.

3.7 Embodied Cognition: What does having a body gives us?

David Kirsh (University of California – San Diego, US)

The overview of embodied cognition that I presented focused on four topics, all explorations of what having a body gives us that goes beyond just having sensors in space. These included:

1. Effectivity: bodies give us capacities that come from having actuators and sensors tied to a single structure (the body) that enables performing action (i.e. agency), joint activity (i.e. doing things in a coordinated manner with others through shared attention) and coadapting to the built environment (i.e. having effects on the constructed world in a manner that is sensitive to the physical attributes of the local environment). Each of these potentialities is special and essentially requires a bodily agent with attention directing capacities, and effectors. An important point to understand about bodies is how the neural system determines the boundary of the body and how this 'body schema' can be altered by practice with a tool, such as a cane or hammer or even an articulated instrument such as nunchucks or violin.
2. Enactive perception: is the view that perception has evolved to pick up dynamic invariants that emerge from the way we interact with things. For instance, we have learned to identify a cup perceptually by having developed saccadic and eye movement strategies that continuously produce predictable cup sensations. Using this model of enactive perception we can ask how adding sensors to our body and adding actuators or tools that alter our behavioral repertoire can lead us to understand cups, other objects, processes and properties in a new way. With a hammer in our hands we can encounter nails and wood in a new way. Another thing to appreciate about enactive perception is that we perceive our environments in a goal and interest relative manner. If you smoke you see potential ashtrays or places to dump ash. If you do not you never even notice those.

3. Interactive Cognition: refers to the way humans and animals interact with things outside their nervous system – their body and other things – to facilitate cognition. A core interactive strategy that humans use – and arguably animals too – is to project future possibilities onto the world and determine whether that 'augmented' world has properties that interest them. A lion might project where a buck might be in a few minutes and scan that region for hiding places to go to now. What would the buck see from there? A person might imagine a diagram or constructions on a diagram in order to solve a geometric problem. Similarly people may projectively try out what performing an action might do to the environment prior to executing that action. There are many ways people project and many ways they alter the environment precisely to increase the power of their projection. This is means that we must design to support or scaffold projection.
4. Experience: having a body means you always have a point of view. It also means that our perception is sensitive to what we might do. We see things by unconsciously considering counterfactuals – what would I see if I look over there or there. Since most of the time we are not looking in those places but we nonetheless have expectations about what we would see were we to look, our current experience includes elements of these counterfactual expectations. We see the couch as having sides (upholstered arms) on both ends despite not really checking, or we see Andy Warhol's 'Wall of Marilyns' as being made up of facial images of Marilryn uniquely – no Jayne Mansfields, even though there might an image in there (Jayne) that is not of Marilyn. Another feature of experience is the way we experience our body as not being in space as much as defining the origin of space. This origin is not like a mathematical centroid; it is where my body ends. This has odd consequences. If I wear glasses I see through them, I never see them. They are part of me. The same for canes and other artifacts we absorb into our body schema. This sort of reflection is relevant when thinking about the consequences of adding actuators and sensory extenders to humans.

3.8 Devices That Overlap With the User's Body

Pedro Lopes (Hasso-Plattner-Institut – Potsdam, DE)

How can interactive devices connect with users in the most immediate and intimate way? This question has driven interactive computing for decades. If we think back to the early days of computing, user and device were quite distant, often located in separate rooms. Then, in the '70s, personal computers "moved in" with users. In the '90s, mobile devices moved computing into users' pockets. More recently, wearables brought computing into constant physical contact with the user's skin. These transitions proved to be useful: moving closer to users and spending more time with them allowed devices to perceive more of the user, allowing devices to act more personal. The main question that drives my research is: what is the next logical step? How can computing devices become even more personal? Some researchers argue that the next generation of interactive devices will move past the user's skin, and be directly implanted inside the user's body. This has already happened in that we have pacemakers, insulin pumps, etc. However, I argue that what we see is not devices moving towards the inside of the user's body but towards the "interface" of the user's body they need to address in order to perform their function. This idea holds the key to more

immediate and personal communication between device and user. The question is how to increase this immediacy? My approach is to create devices that intentionally borrow parts of the user's body for input and output, rather than adding more technology to the body. I call this concept "devices that overlap with the user's body". I'll demonstrate my work in which I explored one specific flavor of such devices, i.e., devices that borrow the user's muscles. In my research I create computing devices that interact with the user by reading and controlling muscle activity. My devices are based on medical-grade signal generators and electrodes attached to the user's skin that send electrical impulses to the user's muscles; these impulses then cause the user's muscles to contract. While electrical muscle stimulation (EMS) devices have been used to regenerate lost motor functions in rehabilitation medicine since the '60s, during my PhD I explored EMS as a means for creating interactive systems. My devices form two main categories: (1) Devices that allow users eyes-free access to information by means of their proprioceptive sense, such as a variable, a tool, or a plot. (2) Devices that increase immersion in virtual reality by simulating large forces, such as wind, physical impact, or walls and heavy objects.

3.9 Inferring Emotion from Touch through analysis of On-Object Sensing

Karon MacLean (University of British Columbia – Vancouver, CA)

We've used low-cost, stretchy touch sensors and machine learning touch recognition to raise the 'emotional intelligence' of social human-robot interaction through bidirectional communication, by inferring changes in emotion state through sensed touch gestures and authoring believable emotional responses to them. This sensing, combined with simple outputs can transform a wide variety of interactions that are situated in the physical world rather than on a traditional computing device.

3.10 Towards Expressive Input Modalities for On-Skin Interaction

Aditya Shekhar Nittala (Universität des Saarlandes, DE)

The human body offers a vast, alwayas available, and quickly accessible real-estate for interaction. For these, reasons, interaction on body has received considerable attention in the HCI community. More recently, a new class of devices which we refer as Interactive Skin devices have emerged, which augement the human body with input and output capabilities. These devices are thin, flexible, can be easily worn on the body, are conformal to the body geometry[1, 2] and enable expressive ways of interaction on the body. However, the current state-of-the art Interactive Skin devices are only limited in terms of interaction and do not leverage all the natural affordances that the human skin offers. In this talk, I present the open challenges[3] and questions for enabling and understanding the new, expressive input modalities on the body, taking into account the various natural physical affordances that the human skin offers. Specifically, I focus on the deformation sensing(pressure, force, shear) on the skin leveraging the stretchability and deformability of the human skin.

References

1 J. Steimle, J. Bergstrom-Lehtovirta, M. Weigel, A. S. Nittala, S. Boring, A. Olwal, and K. Hornbak. On-skin interaction using body landmarks. *Computer*, 50(10):19–27, 2017.
2 Martin Weigel, Aditya Shekhar Nittala, Alex Olwal, and Jürgen Steimle. SkinMarks: Enabling Interactions on Body Landmarks Using Conformal Skin Electronics On-body interaction; on-skin sensing; on-skin display.
3 Aditya Shekhar Nittala and Jürgen Steimle. Digital fabrication pipeline for on-body sensors: Design goals and challenges. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, pages 950–953, New York, NY, USA, 2016. ACM.

3.11 What is a Wearable?

Joseph A. Paradiso (MIT – Cambridge, US)

We have already witnessed profound and often unanticipated developments as IoT is built out and the world is mediated via mainly graphical wireless devices held at arm's length. But what will happen once the world is precognitively interpreted by what we term 'sensory prosthetics' that change what and how humans physically perceive, a world where your own intelligence is split ever more seamlessly between your brain and the cloud? In this talk, I outlined a few research initiatives in my group that anticipate the broad theme of interfacing humans to the ubiquitous electronic "nervous system" that sensor networks will soon extend across things, places, and people, going well beyond the 'Internet of Things'. I started by outlining a few projects we did sensing finger and wrist gestures for more direct manipulation, then described how we now use the voice channel for many things that soon won't be appropriate for it – IoT will evolve into more an extension of self vs. having a dialog with an 'other'. I gave an example of the 'Mediated Atmospheres' research project in my team, where we have an entire room transform (projection, lighting, sound) according to how the occupant reacts to the stimulus (as measured by an array of sensors), as an example of trans-corporeal actuation. I then shifted to perception, describing our Tidmarsh project (where we manifest sensor data from across a restored wetland in different ways), outlining the 'HearThere' device that gives the user sensory (auditory) 'superpowers', sensing with they are attentive, then enhancing sound (via a bone conduction headset) in the direction they are looking. Then I discussed dynamic wearables, that move across the user, pointing to our Rovables project, then introducing another new project in my team with micro-robots that walk on skin using dynamic suction, aimed mainly for medical purposes. I then posed some questions – first 'will we need to wear anything?', pointing to wireless sensing of people using RF (Katabe, Afib) and vitals from computer vision (Picard, Poh). I then posed a bunch of broader questions – Implantables vs Wearables (an issue in the next decades)? – What will be grown vs. what will be wired?(biology is good for some things, wires/silicon for others; can we grow the boundary instead of just wire it, including programming cells, etc.) – How will human presence generalize (when you can plug into ubiquitous sensing)? – What happens when everybody sees their own reality (look at the issues we have with 'fake news' already when it's still at arm's length)? – Where does 'self' stop and 'other' begin (as we physically couple more into the ubiquitous network)? My final question was 'Where are the Aliens?' – if life exists elsewhere (which we'll know in 20 years at most from atmospheres

of exosolar planets), you'd think intelligence would be favored via evolution. Hence, why don't we see signs of life in other solar systems (you'd think advanced civilizations would do observable things). Answers include that either we're alone (my current belief due to the improbability of life – the universe is just a bunch of phase space for probability to play out in), intelligent life quickly self-destructs (highly dystopic view), or we retreat into noncoporeal (virtual) existence – e.g., we do our job too well, and people live in virtual rather than physical worlds (somewhere between dystopia and utopia perhaps, at least from our current understanding).

3.12 Cosmetic Computing: Actions and Urgencies towards a Inclusive, Equitable Landscape of On-Body Technologies

Eric Paulos (University of California – Berkeley, US)

The body as a site for new and exciting innovation. Through this presentation, I articulate a need for transdisciplinary and interdisciplinary approaches to advance the culture and state o the art within OnBody Interactions. Thinking back and building from a historical framing is essential and I present a list of body and performative artists starting from the Triadic Ballet by Oskar Schlemmer and moving through Rebecca Horn, Chris Burden, Yoko Ono, Stelarc, and others. I present an argument for a framing around "Cosmetic Computing" as a vociferous expression of radical individuality and an opportunity for deviance from binary gender norms. It is a catalyst towards an open, playful, and creative expression of individuality through wearable technologies. It's a liberation call across gender, race, and body types. Leveraging the term "cosmetics", originally meaning "technique of dress", we envision how intentionally designed new-wearables, specifically those that integrate with fashionable materials and overlays applied directly atop the skin or body, can (and should) empower individuals towards novel explorations of body and self- expression. Unlike many modern traditional cosmetics that are culturally laden with prescriptive social norms of required usage that are restrictive, sexually binary, and oppressive, we desire a new attitude and creative engagement with wearable technologies that can empower individuals with a more personal, playful, performative, and meaningful "technique of dress" – Cosmetic Computing. Throughout the talk, I presented exemplars of such on-body interactions through I wide range of materiality – hair, fingernails, skin, dynamic clothing, and beyond. Beyond the technical, the philosophical all to action is to operationalize the research through a lens that emphasizes a balance across the personal, performative, provocative, and poetic.

3.13 The Importance of Vestibular and Proprioceptive Signals on Perspective- Taking

Anastasia Pavlidou (MPI für biologische Kybernetik – Tübingen, DE)

The ability to adopt the visuo-spatial perspective of others is fundamental for successful social interactions. Here, we measured how vestibular (Experiment 1) and proprioceptive

(Experiment 2) signals influence perspective-taking abilities. For each experiment, participants completed the "dot-counting task": they evaluated if a number (0-3) presented at the start of each trial matched or mismatched the number of balls visible from their perspective in a visual scene of a 3D virtual room that followed. A task-irrelevant human avatar or arrow was also present in the center of the room that either shared the same or different viewpoint as the participant's. This allowed us to examine the likelihood that participants would implicitly adopt the perspective of the object even though they were not required to. In Experiment 1, participants performed the task while they received low-intensity (1mA) galvanic vestibular stimulation (GVS). Analysis of reaction times between same and different viewpoints revealed that GVS reduced the likelihood that participants implicitly adopted the avatar's perspective, promoting an egocentric viewpoint. In Experiment 2, we manipulated the congruency between the participant's body orientation (e.g. their entire body was facing the right side of the screen) and that of the avatar. When participants and avatars shared the same body orientation, participants were more likely to implicitly adopt the avatar's perspective, resulting in longer response times in the dot-counting task. For both experiments, the effects were not observed for the arrow. Altogether, the results indicate that implicit simulation of another person's viewpoint requires vestibular and proprioceptive signals.

3.14 The Body as a Casual Interaction Device

Henning Pohl (University of Copenhagen, DK)

Interaction with the body can be ubiquitous and subtle, yet is less suited for focused and complex interactions. For example, a body-based UI is likely not great for writing a novel, but pretty good for intermittent tasks or notifications. Interaction on and with the body can integrate and blend in. One variant of this integration is feedback that directly uses the body's own output channels. For example, itching skin is used by the body to steer our attention, but can also be repurposed as a channel for an interactive system.

3.15 Trying to augment the human experience

Joan Sol Roo (Universität des Saarlandes, DE)

This short presentation contains a brief overview of my previous work that led me to the field of body-based interaction. Rather than providing answers, it just frames my current hopes and concerns regarding this type of interfaces and their impact on how we experience our bodies.

3.16 Fashion motivated wearables

Chris Schmandt (MIT – Cambridge, US)

Our skin is the boundary between self and the world, and for millennia our species have decorated our bodies using many methods. We suggest that on-skin computing can benefit by means of design which builds on these existing "beauty practices". Examples include DuoSkin, which provides user interfaces based on metallic tattoos, NailO,which converts decorative finger nail paste on art to capacitive touch sensitive surfaces, and SkinMorph, which affords flexible body armor which stiffens when electrically heated.

3.17 Interactive Skin

Jürgen Steimle (Universität des Saarlandes, DE)

Using human skin as an interactive surface presents unique opportunities for body-based interaction: skin offers a large surface that is always available and easy-to-reach, even during demanding mobility tasks. Skin is inherently multi-modal and lends itself naturally to tactile and visual input and output. It is also a promising platform for continuous monitoring of physiological parameters.

We foresee a new generation of wearable devices, which we call Interactive Skin. These devices reside right on the user's skin and transform it into an input/output surface for computing. By seamlessly fusing natural functions of skin and computational augmentations, they shall enable interactions that are more direct, eyes-free, and more expressive than existing approaches.

However, turning human skin into an I/O surface is demanding. Skin is curved, covers complex geometries, can deform and stretch. This stands in stark contrast to conventional, rigid interactive devices. Skin also has a multitude of physiological functions, including tactile perception, thermal management and transport of vapor, which a skin-worn device must be compatible with. Last but not least, since every user's body is unique and body-worn devices have an important aesthetic component, it will be necessary to personalize such devices to a considerably larger extent than it is common with existing devices.

Together with my team, I address the challenges of Interactive Skin in the following main areas:

Fabrication and Personalization of Interactive Skin Devices: We have developed a suite of fabrication techniques to realize very thin interactive devices that are worn as overlays on human skin. With iSkin [1], we have presented a silicone-based approach for customized stretchable touch sensors that are fabricated using laser patterning. This approach enables new types of body-worn devices, including a) wrappables that are wrapped around body-parts, such as a finger, b) skin stickers that are attached on a desired body location, such as the forearm, and c) on-demand extensions for conventional wearable devices, such as a roll-out keyboard for a smartwatch. In follow-up work [2], we could considerably reduce the thickness of devices by using temporary tattoo paper and multi-layer screen printing with functional

inks. The resulting devices are between 3 and 50 microns thick, which allows them to closely conform to the skin and its fine wrinkles.

Considering the complexity of designing the circuitry of an Interactive Skin device, it is of central importance to develop design tools for interface designers. These tools shall abstract from the low-level circuitry by allowing the designer to design an interface at a high level and then automatically generating the circuitry. We have presented a first design tool that automatically generates a multi-touch sensor for a desired size and shape that is specified by the designer [3]. This is only a first instance – considerably more work is required to investigate how to best support interface designers, to unleash the full power of personalization.

Multi-modal Input and Output: Our interactive skin devices contain various types of printed sensors. These include capacitive sensing of single touch and of high-resolution multi-touch input [3]. Further sensors capture squeezing interactions on the skin and flexion of joints using resistive sensing schemes. In addition, we demonstrated the fabrication of flexible light-emitting displays that are integrated inside temporary tattoos [2]. Arguably most demanding is to integrate tactile output inside the thin form factor of interactive skin. In our most recent work, we have integrated a high-density matrix for electro-tactile stimulation in a temporary tattoo. The tattoo [4] is thin enough to retain most of bare skin's tactile perception. This contributes a new type of tactile interface that allows the user to feel real-world tactile cues through the interface, while augmenting them with computer-generated stimuli.

New Interaction Techniques for Skin: Skin has unique features that present new opportunities for interaction. Skin covers complex geometries and offers numerous tactile cues. These form body landmarks, which can be used during on-skin interaction to provide eyes-free guidance. We have identified a set of landmarks that comprise skeletal landmarks, skin microstructures such as wrinkles, elastic landmarks, visual landmarks, and body-worn accessories [2]. We enable novel interaction techniques by augmenting the filigree geometry of those landmarks with very slim interfaces. For instance, this turns knuckles on the back of the hand into buttons. It can turn fine wrinkles on the fingers into an easy-to-locate slider. Interfaces on elastic flesh can support continuous pressure-based or squeezing-based interactions, etc. All these interactions suggest that future skin-based interfaces should make use of skin's geometry and stretchability, in addition to offering more conventional touch-based gestures.

With this line of research, we aim to contribute a "toolbox" for interaction designers and domain experts, allowing them to start investigating applications of Interactive Skin in various domains. While we expect the first practical applications to emerge in the medical field, we foresee beneficial use in many other areas, including industrial production, mobile computing, sports and fitness, games, and entertainment.

References

1 Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. iSkin. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems – CHI '15*, pages 2991–3000, New York, New York, USA, 2015. ACM Press.

2 Martin Weigel, Aditya Shekhar Nittala, Alex Olwal, and Jürgen Steimle. SkinMarks: Enabling Interactions on Body Landmarks Using Conformal Skin Electronics On-body interaction; on-skin sensing; on-skin display.

3 Aditya Shekhar Nittala, Anusha Withana, Narjes Pourjafarian, and Jürgen Steimle. Multi-Touch Skin: A Thin and Flexible Multi-Touch Sensor for On-Skin Input.

4 Anusha Withana, Daniel Groeger, and Jürgen Steimle. Tacttoo : A Thin and Feel-Through Tattoo for On-Skin Tactile Output. *Proceedings of the 31st Annual ACM Symposium on User Interface Software & Technology – UIST '18*, 2018.

3.18 Designing for and leveraging Active Perception

Paul Strohmeier (University of Copenhagen, DK)

Material properties of the world around us, are revealed to us by our interactions with them. It is tempting to think of the world around us as having fixed properties which we perceive through passive sensors, but various studies suggest that the pre-conscious mircointeractions between our body and the world around us in fact create our subjective experience of the world. This becomes particularly apparent when studying haptic perception. Let us analyze lifting up an object. When holding the object, the fingertips are distorted due to shear stress. This distortion of the fingertips while the object is being lifted leads to a perception of weight [1]. When holding it, there is an interaction between the compression of the fingertip and the corresponding displacement of the fingers through the object. This interaction leads to a perception of compliance [2]. When moving our fingertip over the texture of the object, the interaction between our fingerprints and the materials surface structure causes vibrations. These vibrations are perceived as texture [3]. We experience the world around us through our interactions with the world. This is relevant for HCI as it allows us to provide users with material impressions without recreating the entire material. Rather through studying the sensory modality one wishes to target, one can create the target material, by creating tightly coupled feedback loops, simulating the interaction rather than the material properties [4, 5]. This allows us to create perceptions of virtual worlds without needing to recreate the entire world, it also provides us with guidance of how to design completely new senses and experiences.

References

1 Roland S Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345, 2009.
2 Wouter M Bergmann Tiest and Astrid ML Kappers. Cues for haptic perception of compliance. *IEEE Transactions on Haptics*, 2(4):189–199, 2009.
3 Sliman Bensmaïa and Mark Hollins. Pacinian representations of fine surface texture. *Perception & psychophysics*, 67(5):842–854, 2005.
4 Paul Strohmeier and Kasper Hornbæk. Generating haptic textures with a vibrotactile actuator. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4994–5005. ACM, 2017.
5 Paul Strohmeier, Sebastian Boring, and Kasper Hornbæk. From pulse trains to coloring with vibrations: Motion mappings for mid-air haptic textures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 65. ACM, 2018.

3.19 Beauty Technology and Biosensor Tattoos for Interfacing on and inside the Skin

Katia Vega (University of California – Davis, US)

Can the skin become an interactive platform? This talk describes the possibilities to embed technology on the skin and inside the skin by the use of cosmetics and body modification techniques. In order to move forward traditional cosmetics to interactive ones, Beauty Technology extends the functionality of cosmetics by exploring them as skin interfaces, hair interfaces and nail interfaces. Conductive Makeup, Tech Nails and Hairware are some examples of Beauty Technologies. On the other hand, humans also embraced body modification as a deliberate procedure for altering the appearance and form of the body. The Dermal Abyss explores the possibilities of replacing traditional tattoo ink with biosensors that changes colors in response to changes in our metabolism. In this way, the skin is a bio-display that reveals information that is inside the body such as pH, sodium and glucose levels.

4 Working groups

4.1 Visions in human computer integration

Liwei Chan (National Chiao-Tung University – Hsinchu, TW), David Kirsh (University of California – San Diego, US), Pedro Lopes (Hasso-Plattner-Institut – Potsdam, DE), and Paul Strohmeier (University of Copenhagen, DK)

We discuss three scenarios in which a very deep & tight human computer integration results in new senses: (1) new modes of immersion (empathy) with other entities, other beings, scales; new methods of sensualizing complex entities (make sense of new things); and, new multimodal encounters can take place with new senses opening up opportunities to design whole new experiences (art). We finish with three necessary steps to achieve some of these ideas.

4.2 Interactivity: the problem of reading off control intentions

David Kirsh (University of California – San Diego, US), Liwei Chan (National Chiao-Tung University – Hsinchu, TW), Clément Duhart (MIT – Cambridge, US), Aditya Shekhar Nittala (Universität des Saarlandes, DE), and Chris Schmandt (MIT – Cambridge, US)

In our group we discussed the foundational question: how can people interact with devices that might be in, half way in, on or off our body. As befits a question as basic as this we started with some assumptions about devices. They have a set of control parameters (C1

... Cn) that a user can in principle change; they have a set of actions (A1 .. An) that they can in principle perform some of which at the direction of the agent, and they have a set of information display factors (I1 .. In) that they can manifest to reveal things about their own state, the agent's state, states of the world and so forth. How does the agent know how to manipulate those control parameters? The more devices we have the more we forget how to work with them. And people like to have their own personalized ways of controlling, though again they may forget what these are. Further, if we are augmented with a sensor we need to couple very tightly so that we can dynamically control it to pick up invariants that only show up through moving it in certain ways. Sensor control is thick. The solution we struck on may work for the control problem associated with thinner control though not for sensor control. Control of sensors is more like playing a musical instrument; it must be learned through practice. For thinner control problems we can think of the problem like this. First let's discuss it for a simple problem like remotely controlling a light that has two orthogonal parameters (intensity, color) without using voice control and without touching a switch. The problem is to read off our intentions without asking us. This is a fundamental problem. How can a system, whether on our body, or some contextual sensor system in the environment, read intentions? We can assume that we signal them in some way. But the signals may be implicit or explicit. We might signal them implicitly through implicit body language or by proceeding in a manner that assumes the system will adapt to our needs given the context. == or we may signal them explicitly by gesture using our hands, body or face or by eye movement. How do humans do this? It is not reasonable to hope that a system might do better than a human unless it has access to non-behavioral or non-contextual parameters such as brain states, or other inner bodily states. How then might a human proceed? They interpret the context, they see where we are gazing, and then interpret our gestures if we make any. Since the person may not read our intentions correctly each moment we can treat this problem as a type of iterative coordination game. If the person controls the lights they respond to our action by changing the lights in some way. If they get it right they must interpret our next actions as indicating we are satisfied. If we are dissatisfied we respond by acting to get them to improve their response. They guess, we react and if this game has an equilibrium then everyone is happy and the reader has done the right thing and knows it. We believe the future will involve our interacting with an AI middleware system that can read our intentions. This is a fundamental problem that will apply whether we are trying to control remote devices or devices on or in our bodies. And the more control states there are the more it is a serious problem.

4.3 Body Noir

Antonio Krüger (DFKI – Saarbrücken, DE), Kayla J. Heffernan (The University of Melbourne, AU), Eric Paulos (University of California – Berkeley, US), Chris Schmandt (MIT – Cambridge, US), and Katia Vega (University of California – Davis, US)

This group discussed and presented the negative visions for on-body sensing and actuation and provided possible solutions for these. The discussion involved various negative aspects such as using the body-based interaction as a mechanism for controlling users, their mind. Other issues such as constant tracking of the body-based private data was discussed. Lastly,

body-based interfaces could devalue human dignity and can make user over-rely on technology making them not-opt-out of the technology. Some possible outcomes were to have government policies and regulations so that users can opt-out of the technology. Another possibility is to support critical design as valid respected research within the community and on the societal level, there should be technology -free parks and zones.

4.4 The Future of On-Body Interfaces

Joseph A. Paradiso (MIT – Cambridge, US), Nadia Bianchi-Berthouze (University College London, GB), Clément Duhart (MIT – Cambridge, US), Nur Hamdan (RWTH Aachen, DE), Christian Holz (Microsoft Research – Redmond, US), Kasper Hornbaek (University of Copenhagen, DK), Karon MacLean (University of British Columbia – Vancouver, CA), and Aditya Shekhar Nittala (Universität des Saarlandes, DE)

This group discussion led to the ideation about the positive visions for the future of on-body interfaces. The group presented the various avenues for the future of on-body interfaces: for e.g. how legal/democratic frameworks can bring about a positive change, similarly they discussed the various application areas for on-body interfaces which include, Graceful ageing, personalized medication also reflected on how on-body interfaces can bring about low energy consumptions. Some of the negative connotations were also discussed such as slavery that can be inflicted with on-body interfaces (though EMS), privacy issues which can give outsiders access to information about one's body and the excessive data logging that can be exploited by greedy organizations.

4.5 Challenges in Human Computer Integration

Joseph A. Paradiso (MIT – Cambridge, US), Nadia Bianchi-Berthouze (University College London, GB), Clément Duhart (MIT – Cambridge, US), Nur Hamdan (RWTH Aachen, DE), Christian Holz (Microsoft Research – Redmond, US), Kasper Hornbaek (University of Copenhagen, DK), Karon MacLean (University of British Columbia – Vancouver, CA), and Aditya Shekhar Nittala (Universität des Saarlandes, DE)

This group discussion led to the ideation about the positive visions for the future of on-body interfaces. The group presented the various avenues for the future of on-body interfaces: for e.g. how legal/democratic frameworks can bring about a positive change, similarly they discussed the various application areas for on-body interfaces which include, Graceful ageing, personalized medication also reflected on how on-body interfaces can bring about low energy consumptions. Some of the negative connotations were also discussed such as slavery that can be inflicted with on-body interfaces (though EMS), privacy issues which can give outsiders access to information about one's body and the excessive data logging that can be exploited by greedy organizations.

Participants

- Eduard Arzt
INM – Saarbrücken, DE
- Nadia Bianchi-Berthouze
University College London, GB
- Liwei Chan
National Chiao-Tung University – Hsinchu, TW
- Clément Duhart
MIT – Cambridge, US
- Michael Haller
University of Applied Sciences Upper Austria – Hagenberg, AT
- Nur Hamdan
RWTH Aachen, DE
- Kayla J. Heffernan
The University of Melbourne, AU
- Christian Holz
Microsoft Research – Redmond, US
- Kasper Hornbaek
University of Copenhagen, DK
- David Kirsh
University of California – San Diego, US
- Antonio Krüger
DFKI – Saarbrücken, DE
- Pedro Lopes
Hasso-Plattner-Institut – Potsdam, DE
- Paul Lukowicz
DFKI – Kaiserslautern, DE
- Karon MacLean
University of British Columbia – Vancouver, CA
- Aditya Shekhar Nittala
Universität des Saarlandes, DE
- Joseph A. Paradiso
MIT – Cambridge, US
- Eric Paulos
University of California – Berkeley, US
- Anastasia Pavlidou
MPI für biologische Kybernetik – Tübingen, DE
- Henning Pohl
University of Copenhagen, DK
- Ivan Poupyrev
Google Inc. – Mountain View, US
- Joan Sol Roo
Universität des Saarlandes, DE
- Chris Schmandt
MIT – Cambridge, US
- Albrecht Schmidt
LMU München, DE
- Jürgen Steimle
Universität des Saarlandes, DE
- Paul Strohmeier
University of Copenhagen, DK
- Katia Vega
University of California – Davis, US