



DAGSTUHL REPORTS

Volume 3, Issue 2, February 2013

Fault Prediction, Localization, and Repair (Dagstuhl Seminar 13061) <i>Mary Jean Harrold, Friedrich Steimann, Frank Tip, and Andreas Zeller</i>	1
Decentralized Systems for Privacy Preservation (Dagstuhl Seminar 13062) <i>Sonja Buchegger, Jon Crowcroft, Balachander Krishnamurthy, and Thorsten Strufe</i>	22
Dependence Logic: Theory and Applications (Dagstuhl Seminar 13071) <i>Samson Abramsky, Juha Kontinen, Jouko Väänänen, and Heribert Vollmer</i>	45
Mechanisms of Ongoing Development in Cognitive Robotics (Dagstuhl Seminar 13072) <i>Jacqueline Fagard, Roderic A. Grupen, Frank Guerin, and Norbert Krüger</i>	55
Consistency in Distributed Systems (Dagstuhl Seminar 13081) <i>Bettina Kemme, Ganesan Ramalingam, André Schiper, Marc Shapiro, and Kapil Vaswani</i>	92
Communication Complexity, Linear Optimization, and lower bounds for the nonnegative rank of matrices (Dagstuhl Seminar 13082) <i>LeRoy B. Beasley, Hartmut Klauck, Troy Lee, and Dirk Oliver Theis</i>	127
Analysis, Test and Verification in The Presence of Variability (Dagstuhl Seminar 13091) <i>Paulo Borba, Myra B. Cohen, Axel Legay, and Andrzej Wasowski</i>	144

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/dagrep>

Publication date

June, 2013

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>.

License

This work is licensed under a Creative Commons Attribution 3.0 Unported license: CC-BY.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Aims and Scope

The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.

In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:

- an executive summary of the seminar program and the fundamental results,
 - an overview of the talks given during the seminar (summarized as talk abstracts), and
 - summaries from working groups (if applicable).
- This basic framework can be extended by suitable contributions that are related to the program of the seminar, e.g. summaries from panel discussions or open problem sessions.

Editorial Board

- Susanne Albers
- Bernd Becker
- Karsten Berns
- Stephan Diehl
- Hannes Hartenstein
- Stephan Merz
- Bernhard Mitschang
- Bernhard Nebel
- Han La Poutré
- Bernt Schiele
- Nicole Schweikardt
- Raimund Seidel
- Michael Waidner
- Reinhard Wilhelm (*Editor-in-Chief*)

Editorial Office

Marc Herbstritt (*Managing Editor*)

Jutka Gasirowski (*Editorial Assistance*)

Thomas Schillo (*Technical Assistance*)

Contact

Schloss Dagstuhl – Leibniz-Zentrum für Informatik
Dagstuhl Reports, Editorial Office
Oktavie-Allee, 66687 Wadern, Germany
reports@dagstuhl.de

Digital Object Identifier: 10.4230/DagRep.3.2.i

www.dagstuhl.de/dagrep

Fault Prediction, Localization, and Repair

Edited by

Mary Jean Harrold¹, Friedrich Steimann², Frank Tip³, and
Andreas Zeller⁴

1 Georgia Tech, US, harrold@cc.gatech.edu

2 FernUniversität in Hagen, DE, steimann@acm.org

3 University of Waterloo, CA, ftip@uwaterloo.ca

4 Universität des Saarlandes, DE, zeller@cs.uni-saarland.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13061 “Fault Prediction, Localization, and Repair”.

Software debugging, which involves localizing, understanding, and removing the cause of a failure, is a notoriously difficult, extremely time consuming, and human-intensive activity. For this reason, researchers have invested a great deal of effort in developing automated techniques and tools for supporting various debugging tasks. In this seminar, we discussed several different tools and techniques that aid in the task of Fault Prediction, Localization and Repair.

The talks encompassed a wide variety of methodologies for fault prediction and localizing, such as

- statistical fault localization,
- core dump analysis,
- taint analysis,
- program slicing techniques,
- dynamic fault-comprehension techniques,
- visualization techniques,
- combining hardware and software instrumentation for fault detection and failure prediction,
- and verification techniques for checking safety properties of programs.

For automatically (or semi-automatically) repairing faulty programs, the talks covered approaches such as

- automated repair based on symbolic execution, constraint solving and program synthesis,
- combining past fix patterns, machine learning and semantic patch generation techniques,
- a technique that exploits the intrinsic redundancy of reusable components,
- a technique based on memory-access patterns and a coverage matrix,
- a technique that determines a combination of mutual-exclusion and order relationships that, once enforced, can prevent buggy interleaving.

In addition, this seminar also explored some unusual topics such as Teaching Debugging, using Online Courses. Another interesting topic covered was the low representation of females in computing, and how programming and debugging tools interact with gender differences.

Seminar 03.–08. February, 2013 – www.dagstuhl.de/13061

1998 ACM Subject Classification D.3 Programming Languages, D.2 Software Engineering, D.2.5 Testing and Debugging, D.2.4 Software/Program Verification, F.3 Logics and Meanings of Programs, F.3.1 Specifying and Verifying and Reasoning about Programs

Keywords and phrases Program analysis, Automated debugging, Fault prediction, Fault repair, Fault localization, Statistical debugging, Change impact analysis

Digital Object Identifier 10.4230/DagRep.3.2.1

Edited in cooperation with Mangala Gowri Nanda



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Fault Prediction, Localization, and Repair, *Dagstuhl Reports*, Vol. 3, Issue 2, pp. 1–21

Editors: Mary Jean Harrold, Friedrich Steimann, Frank Tip, and Andreas Zeller



DAGSTUHL
REPORTS

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

1 Executive Summary

Mary Jean Harrold

Friedrich Steimann

Frank Tip

Andreas Zeller

License © Creative Commons BY 3.0 Unported license
© Mary Jean Harrold, Friedrich Steimann, Frank Tip, and Andreas Zeller

Even today, an unpredictable part of the total effort devoted to software development is spent on debugging, i.e., on finding and fixing bugs. This is despite the fact that powerful static checkers are routinely employed, finding many bugs before a program is first executed, and also despite the fact that modern software is often assembled from pieces (libraries, frameworks, etc.) that have already stood the test of time. In fact, while experienced developers are usually quick at finding and fixing their own bugs, they too spend too much time with fixing the interplay of components that have never been used in combination before, or just debugging the code of others. Better automated support for predicting, locating, and repairing bugs is therefore still required.

Due to the omnipresence of bugs on the one side and the vastly varying nature of bugs on the other, the problems of fault prediction, localization, and repair have attracted research from many different communities, each relying on their individual strengths. However, often enough localizing a bug resembles the solution of a criminal case in that no single procedure or evidence is sufficient to identify the culprit unambiguously. It is therefore reasonable to expect that the best result can only be obtained from the combination of (insufficient) evidence obtained by different, and ideally independent, procedures. One main goal of this seminar is therefore to connect the many different strands of research on fault prediction, localization, and repair.

For researchers it is not always obvious how debugging is embedded in the software production process. For instance, while ranking suspicious program statements according to the likelihood of their faultiness may seem like a sensible thing to do from a research perspective, programmers may not be willing to look at more than a handful of such locations when they have their own inkling of where a bug might be located. On the other hand, commercial programmers may not be aware of the inefficiency of their own approaches to debugging, for which promising alternatives have been developed by academics. Bringing together these two different perspectives is another goal of this seminar.

Last but not least, the growing body of open source software, and with it the public availability of large regression test suites, provide unprecedented possibilities for researchers to evaluate their approaches on industrial-quality benchmarks. In fact, while standard benchmarks such as the so-called Siemens test suite still pervade the scientific literature on debugging, generalization of experimental results obtained on such a small basis is more than questionable. Other disciplines, such as the model checking or the theorem proving communities, have long established competitions based on open benchmarks to which anyone can submit their problems. Based on such benchmarks, progress would be objectively measurable, and advances in research would be better visible. It is another goal of this seminar to establish a common understanding for the need of such benchmarks, and also to initiate the standards necessary for installing them.

2 Table of Contents

Executive Summary

<i>Mary Jean Harrold, Friedrich Steimann, Frank Tip, and Andreas Zeller</i>	2
---	---

Overview of Talks

Automated Debugging: Are We There Yet? <i>Alessandro Orso</i>	5
Automatic Recovery from Runtime Failures <i>Alessandra Gorla</i>	5
Reconstructing Core Dumps <i>Jeremias Röbler</i>	6
Combining Machine Learning with Combinatorial Search in Program Repair <i>Satish Chandra</i>	6
The Design of Bug Fixes—ICSE 2013 <i>Thomas Zimmermann</i>	6
Lower and upper bounds of coverage-based fault localization accuracy <i>Friedrich Steimann</i>	7
Avoiding Confoundings in Delta Debugging type of Causality Inference <i>Xiangyu Zhang</i>	7
Combining Hardware and Software Instrumentation for Fault Detection and Failure Prediction <i>Cemal Yilmaz</i>	8
Males and Females Debugging: Are the Tools Getting in the Way? <i>Margaret M. Burnett</i>	9
GZoltar: An Eclipse plug-in for Testing and Debugging <i>Rui Abreu</i>	9
Programming with Rosette: Code Checking, Fault Localization, Angelic Execution, and Synthesis in 20 Minutes <i>Emina Torlak</i>	10
An Overview of EzUnit <i>Marcus Frenkel</i>	10
Basics of Causal Fault Localization from Observational Data <i>Andy Podgurski</i>	11
Understanding the Relationship Between Recent Fault-Localization Techniques and Causality <i>George K. Baah</i>	11
SEMFIX: Automated Program Repair using Semantic Analysis <i>Abhik Roychoudhury</i>	12
Teaching Debugging <i>Andreas Zeller</i>	12
SAT Solvers for Software Reliability, Security and Repair <i>Vijay Ganesh</i>	13

Fault Localization using Maximum Satisfiability <i>Rupak Majumdar</i>	13
On the biodiversity of source code: rigid or plastic repair? <i>Benoit Baudry, Martin Monperrus</i>	14
Leveraging Software Text Analytics To Detect, Diagnose, and Fix Bugs <i>Lin Tan</i>	14
Genetic Mutation Conditioned Amorphous Parametric Hybrid “Dual” Slicing <i>Jens Krinke</i>	15
Information Flow Analysis for JavaScript <i>Christian Hammer</i>	15
Gaining inSight into Programs that Analyze Programs – By Visualizing the Analyzed Program <i>Mangala Gowri Nanda</i>	16
Blended Taint Analysis for JavaScript <i>Barbara G. Ryder</i>	16
Accurate and Scalable Security Analysis of Web Applications <i>Marco Pistoia</i>	17
An Overview of the Apollo Project: Test Generation and Fault Localization for Web Applications <i>Shay Artzi, Frank Tip, and Julian Dolby</i>	17
Automated Repair of HTML Generation Errors in PHP Applications Using String Constraint Solving <i>Hesam Samimi</i>	18
Demo of the Tarantula Fault-Localization Tool <i>Jake Cobb</i>	18
Three Projects Related to Fault Prediction, Localization, and Repair <i>Milos Gligoric</i>	19
Automated Concurrency-Bug Fixing <i>Ben Liblit</i>	19
Localizing Non-deadlock Concurrency Bugs <i>Mary Jean Harrold</i>	20
Discussion and Open Problems	20
Conclusion	20
Participants	21

3 Overview of Talks

[Day 1] Session: Introduction [Mon, Feb 04, 09:30-10:30]

3.1 Automated Debugging: Are We There Yet?

Alessandro Orso (Georgia Tech, US)

License © Creative Commons BY 3.0 Unported license
© Alessandro Orso

Joint work of Orso, Alessandro; Parnin, Chris

Main reference C. Parnin, A. Orso, “Are Automated Debugging Techniques Actually Helping Programmers?” in Proc. of the 2011 Int’l Symp. on Software Testing and Analysis (ISSTA 2011), pp. 199–209, ACM, 2011.

URL <http://dx.doi.org/10.1145/2001420.2001445>

Software debugging, which involves localizing, understanding, and removing the cause of a failure, is a notoriously difficult, extremely time consuming, and human-intensive activity. For this reason, researchers have invested a great deal of effort in developing automated techniques and tools for supporting various debugging tasks. Although potentially useful, most of these techniques have yet to fully demonstrate their practical effectiveness. Moreover, many current debugging approaches suffer from some common limitations and rely on several strong assumptions on both the characteristics of the code being debugged and how developers behave when debugging such code. This talk will provide an overview of the state of the art in the broader area of software debugging, discuss strengths and weaknesses of the main existing debugging techniques, present a set of open challenges in this area, and sketch future research directions that may help address these challenges.

[Day 1] Session: Recovery and Reconstruction [Mon, Feb 04, 11:00-12:15]

3.2 Automatic Recovery from Runtime Failures

Alessandra Gorla (Universität des Saarlandes, DE)

License © Creative Commons BY 3.0 Unported license
© Alessandra Gorla

Joint work of Gorla, Alessandra; Carzaniga, Antonio; Mattavelli, Andrea; Perino, Nicolò; Pezzè, Mauro

We present a technique to make applications resilient to failures. This technique is intended to maintain a faulty application functional in the field while the developers work on permanent and radical fixes. We target field failures in applications built on reusable components. In particular, the technique exploits the intrinsic redundancy of those components by identifying workarounds consisting of alternative uses of a faulty component that avoid the failure. The technique is currently implemented for Java applications but makes little or no assumptions about the nature of the application and works without interrupting the execution flow of the application and without restarting its components. We demonstrate and evaluate this technique on four mid-size applications and two popular libraries of reusable components affected by real and/or seeded faults. In these cases the technique is effective, maintaining the application fully functional, with between 19% and 48% of the failure-causing faults, depending on the application. The experiments also show that the technique incurs an acceptable runtime overhead in all cases.

3.3 Reconstructing Core Dumps

Jeremias Rößler (Universität des Saarlandes, DE)

License © Creative Commons BY 3.0 Unported license
© Jeremias Rößler

Main reference J. Rößler, A. Zeller, G. Fraser, C. Zamfir, G. Candea, “Reconstructing Core Dumps,” in Proc. of the 6th IEEE Int’l Conf. on Software Testing, Verification and Validation (ICST’13), March 2013, to appear.

URL <http://www.st.cs.uni-saarland.de/publications/files/roessler-icst-2013.pdf>

When a software failure occurs in the field, it is often difficult to reproduce. Guided by a memory dump at the moment of failure (a “core dump”), our RECORE test case generator searches for a series of events that precisely reconstruct the failure from primitive data. Applied on seven non-trivial Java bugs, RECORE reconstructs the exact failure in five cases without any runtime overhead in production code.

[Day 1] Session: Repair 1 [Mon, Feb 04, 13:30-15:00]

3.4 Combining Machine Learning with Combinatorial Search in Program Repair

Satish Chandra (IBM India – Bangalore, IN)

License © Creative Commons BY 3.0 Unported license
© Satish Chandra

We consider the problem of automatically generating repair suggestions for a defective database program, one that behaves incorrectly due to an error in WHERE condition of a SELECT statement. A common setting in database programs is that the output is incorrect only for part of the data, e.g. for certain key values. In this paper, we use techniques from machine learning to take advantage of the information revealed by the defect-free data. Our basic approach is to learn a decision tree from correct behavior—including correct behavior on the defect-inducing data—of the SELECT statement. This decision tree can give valuable hints, if not directly the correct WHERE condition. Our novelty is in the crucial step of determining the correct behavior of the defect-inducing data. We do this using a combination of SAT-based search and prediction generated by support vector machines (SVMs). Our insight is that SVMs can learn from the behavior of the defect-free data to predict the behavior of defect-inducing data with high accuracy, with SAT-based search bridging over any deficit in the accuracy efficiently. We have implemented this approach and have done preliminary experiments on suite of programs and data sets obtained from real- world setting.

3.5 The Design of Bug Fixes—ICSE 2013

Thomas Zimmermann (Microsoft Research – Redmond, US)

License © Creative Commons BY 3.0 Unported license
© Thomas Zimmermann

Joint work of Murphy-Hill, Emerson; Zimmermann, Thomas; Bird, Christian; Nagappan, Nachiappan
Main reference E. Murphy-Hill, T. Zimmermann, C. Bird, N. Nagappan, “The Design of Bug Fixes,” in Proc. of the 35th Int’l Conf. on Software Engineering (ICSE’13), pp. 332–341, IEEE/ACM, 2013.

URL <http://dl.acm.org/citation.cfm?id=2486833>

When software engineers fix bugs, they may have several options as to how to fix those bugs. Which fix is chosen has many implications, both for practitioners and researchers: What

is the risk of introducing other bugs during the fix? Is the bug fix in the same code that caused the bug? Is the change fixing the cause or just covering a symptom? In this paper, we investigate the issue of alternative fixes to bugs and present an empirical study of how engineers make design choices about how to fix bugs. Based on qualitative interviews with 40 engineers working on a variety of products, 6 bug triage meetings, and a survey filled out by 326 engineers, we found that there are a number of factors, many of them non-technical, that influence how bugs are fixed, such as how close to release the software is. We also discuss several implications for research and practice, including ways to make bug prediction and localization more accurate.

3.6 Lower and upper bounds of coverage-based fault localization accuracy

Friedrich Steimann (Fernuniversität in Hagen, DE)

License © Creative Commons BY 3.0 Unported license
© Friedrich Steimann

Ever since the first publications on coverage-based fault localization, an empirical evaluation of the diagnostic accuracy, or of the (theoretical) effort of localizing a fault, has been a condition sine qua non. Although accuracy measures, which are usually based on rankings of program elements to be inspected in search for a fault, may be questioned as indicators of the usefulness of fault locators, they are certainly good for evaluating their performance, in a theoretical setting at least. However, in absence of absolute bounds, evaluations of the accuracies of fault locators are usually relative, that is, by comparison with other fault locators. With his talk, the speaker wishes to share and discuss his thoughts about establishing absolute lower and upper bounds of the accuracy of coverage-based fault localization, which allow one to assess the performance of any individual fault locator independently of all others.

[Day 1] Session: Fault Localization 1 [Mon, Feb 04, 15:30-16:30]

3.7 Avoiding Confoundings in Delta Debugging type of Causality Inference

Xiangyu Zhang (Purdue University, US)

License © Creative Commons BY 3.0 Unported license
© Xiangyu Zhang

Joint work of William Nick Sumner, Zhang, Xiangyu;

Main reference W.N. Sumner, X. Zhang, “Comparative causality: explaining the differences between executions,” in Proc. of the 35th Int’l Conf. on Software Engineering (ICSE’13), pp. 272–281, IEEE/ACM, 2013.

URL <http://dl.acm.org/citation.cfm?id=2486825>

In this talk, I will briefly introduce a number of challenges we have over-come in making delta debugging more practical. In particular, I will focus on improving the causality inference engine. Delta debugging type of techniques reason about causality through state replacement, that is, replacing part of the program state at an earlier point to observe whether the failure can be induced. However, such replacement often causes undesirable entangling of the replaced state and the original state and does not properly handle errors caused by

execution omission. I will introduce our new causality inference engine that leverages a recently developed program slicing technique to perform better state replacement.

3.8 Combining Hardware and Software Instrumentation for Fault Detection and Failure Prediction

Cemal Yilmaz (Sabanci University – Istanbul, TR)

License © Creative Commons BY 3.0 Unported license
© Cemal Yilmaz

Joint work of Cemal, Yilmaz; Porter, Adam

Main reference C. Yilmaz, A. Porter, “Combining hardware and software instrumentation to classify program executions,” in Proc. of the 18th ACM SIGSOFT Int’l Symp. on Foundations of Software Engineering (FSE ’10), pp. 67–76, ACM, 2010.

URL <http://dx.doi.org/10.1145/1882291.1882304>

Many data-driven program analysis approaches have studied ways to infer properties of software systems by using execution data gathered from the running systems, usually with software-level instrumentation. These approaches instrument the source code and/or binaries of programs, collect execution data from program executions every time the instrumentation code is exercised, and analyze the collected data to help shape future software development efforts. Two specific applications of this general approach, which are the focus of this paper, is fault detection, i.e., distinguishing failed executions from successful executions, and runtime failure prediction, i.e., predicting the manifestation of failures at runtime before they actually occur. A fundamental assumption of these and similar approaches is that there are identifiable and repeatable patterns in program executions and that similarities and deviations from these patterns can be used to perform many quality assurance tasks. While existing efforts appear to produce promising results, one less well-understood issue is how best to limit the runtime overhead introduced by these approaches and whether and how tradeoffs can be made between overhead and analysis accuracy. This issue is important because these approaches have been targeted at deployed software systems; excessive runtime overhead is generally undesirable. Therefore, it is important to limit instrumentation overhead as much as possible while still supporting the highest levels of analysis accuracy.

In this work we conjecture that large overhead reductions may derive from reducing the cost of the measurement instruments themselves. To evaluate this conjecture, we have designed and evaluated improved approaches in which most of the data collection work is pushed onto the hardware via the use of hardware performance counters. The data is augmented with further data collected by a minimal amount of software instrumentation that is added to the system’s software. We contrast this approach with other approaches implemented purely in hardware or purely in software. Our empirical evaluations, conducted on widely-used open source projects, strongly suggest that 1) there are identifiable and repeatable patterns in program executions, 2) our hybrid hardware and software instrumentation approach is as good or better than other approaches in capturing these patterns; and they do so at a fraction of the cost of using purely software-based instrumentation, and 3) identifying similarities to these patterns and/or deviations from them can reliably detect faults and predict failures at runtime.

[Day 2] Session: Tools 1 [Tue, Feb 05, 09:00-10:30]**3.9 Males and Females Debugging: Are the Tools Getting in the Way?***Margaret M. Burnett (Oregon State University, US)***License** © Creative Commons BY 3.0 Unported license
© Margaret M. Burnett**Joint work of** Burnett, Margaret M.; Beckwith, L.; Wiedenbeck, S.; Fleming, Scott D.; Cao, Jill; Park, Thomas H.; Grigoreanu, Valentina; Rector, Kyle**Main reference** M.M. Burnett, L. Beckwith, S. Wiedenbeck, S.D. Fleming, J. Cao, T.H. Park, V. Grigoreanu, K. Rector, "Gender Pluralism in Problem-Solving Software," *Interacting with Computers*, 23(5), Elsevier, September 2011, pp. 450–460.**URL** <http://dx.doi.org/10.1016/j.intcom.2011.06.004>

Although there has been recent investigation into how to understand and ameliorate the low representation of females in computing, there has been little research into how programming and debugging tools interact with gender differences. This talk reports the investigations my collaborators and I have conducted into whether and how software tools' features affect males' and females' debugging performance differently, and describes the beginnings of work on promising tool changes that help both male and female software developers across populations, ranging from end-user programmers to software professionals.

[Day 2] Session: Tools 2 [Tue, Feb 05, 11:00-12:15]**3.10 GZoltar: An Eclipse plug-in for Testing and Debugging***Rui Abreu (University of Porto, PT)***License** © Creative Commons BY 3.0 Unported license
© Rui Abreu**Joint work of** Abreu, Rui; Ribeiro, André; Campos, José; Perez, Alexandre**Main reference** J. Campos, A. Ribeiro, A. Perez, R. Abreu, "GZoltar: An Eclipse Plug-In for Testing and Debugging," in *Int'l Conf. on Automated Software Engineering (ASE'12)*, pp. 378–381, ACM, 2012.**URL** <http://dx.doi.org/10.1145/2351676.2351752>

Testing and debugging is the most expensive, error-prone phase in the software development life cycle. Automated testing and diagnosis of software faults can therefore drastically improve the efficiency of this phase, this way improving the overall quality of the software.

In this talk, we present a toolset for automatic testing and fault localization, dubbed GZoltar, which hosts techniques for (regression) test suite minimization and automatic fault diagnosis (namely, spectrum-based fault localization). The toolset provides the infrastructure to automatically instrument the source code of software programs to collect runtime data needed by the underlying techniques. Subsequently, the data is analyzed to both minimize the test suite just executed and compute a visual diagnostic report of the source code entities (methods, statements, etc). The toolset is a plug-and-play plug-in for the Eclipse IDE to ease world-wide adoption.

The toolset can be downloaded at www.gzoltar.com. If interested in the slides of the presentation, visit <http://www.gzoltar.com/dagstuhl/>.

3.11 Programming with Rosette: Code Checking, Fault Localization, Angelic Execution, and Synthesis in 20 Minutes

Emina Torlak (University of California – Berkeley, US)

License © Creative Commons BY 3.0 Unported license
© Emina Torlak

Joint work of Torlak, Emina; Bodik, Rastislav

Decision procedures have helped automate key aspects of programming: coming up with a code fragment that implements a desired behavior (synthesis); establishing that an implementation satisfies a desired property (code checking); locating code fragments that cause an undesirable behavior (fault localization); and running a partially implemented program to test its existing behaviors (angelic execution). Each of these aspects is supported, at least in part, by a family of formal tools. Most such tools are built on infrastructures that are tailored for a particular purpose, e.g., Boogie for verification and Sketch for synthesis. But so far, no single infrastructure provides a platform for automating the full spectrum of programming activities, making it hard to share advances (in encodings, abstractions, and domain-specific optimizations) across different families of tools.

This talk introduces Rosette, a new shared infrastructure for computer-aided programming. Rosette is a high-level functional language with symbolic reasoning capabilities, designed to enable rapid prototyping of domain-specific programming tools. The Rosette language is itself a small EDSL (embedded domain-specific language) that inherits and exposes extensive support for meta programming from its host language, Racket. To prototype a tool in Rosette, we first define the target programming model or EDSL by writing an interpreter for it. Depending on the purpose of the tool, the next step is to allow some constructs in the target language to produce symbolic, rather than just concrete, values. In the last step, we define the tool’s behavior by formulating a suitable satisfiability query about programs in the target language (e.g., a code checking tool searches for an input that satisfies the program’s pre-condition and a negation of its post-condition). Rosette’s symbolic engine then does the rest: executing a program in the target EDSL yields a symbolic encoding of the program’s semantics, which is used to instantiate and discharge the tool’s satisfiability query. We show how to use Rosette to prototype a program checker, a fault localizer, an angelic oracle, and an inductive synthesizer for a tiny DSL.

3.12 An Overview of EzUnit

Marcus Frenkel (FernUniversität in Hagen, DE)

License © Creative Commons BY 3.0 Unported license
© Marcus Frenkel

Joint work of Frenkel, Marcus; Steimann, Friedrich

URL <http://www.fernuni-hagen.de/ps/prjs/EzUnit5/>

Unit testing is one of the important tools for ensuring the quality of software. Unit tests are able to indicate the presence of errors introduced into program code, but normally they don’t provide much evidence where the fault actually might be. This task is fulfilled by coverage-based fault locators, which can use the coverage information provided by the unit tests to compute more likely and less likely locations for faults. EzUnit is an Eclipse plug-in which supports programmers in their fault localization task by applying various implemented fault locators to the test results provided by JUnit, as well as allows the implementation and

evaluation of new fault locators. The key feature of EzUnit is its modular structure which allows for an easy extension or replacement of its parts; these parts include interfaces e.g. to the tracer and test runner, fault injectors and locators, or a repository to draw test probands or coverage matrices from for the purpose of evaluation. In this talk, the EzUnit framework shall be introduced, along with its possibilities to implement and evaluate novel approaches for fault localization, and the benefits it has to use EzUnit as an easily alterable platform for fault location evaluations.

[Day 2] Session: Fault Localization and Repair 2 [Tue, Feb 05, 13:30-15:00]

3.13 Basics of Causal Fault Localization from Observational Data

Andy Podgurski (Case Western Reserve University – Cleveland, US)

License  Creative Commons BY 3.0 Unported license
© Andy Podgurski

Statistical Fault Localization (SFL) techniques use statistical measures of association between occurrences of program failures and occurrences of certain runtime events, such as coverage of individual program statements, to identify “suspicious” program locations that may contain faults. However, most proposed SFL metrics are subject to confounding bias that can seriously distort suspiciousness scores so they are not helpful for fault localization. Fortunately, important techniques (developed in recent decades by researchers from several disciplines) for making principled causal inferences from observational data can be used to reduce confounding bias and markedly improve the performance of SFL. In this talk, we explain the basic ideas behind these techniques, including causal effect measures, confounding, potential outcome random variables, conditional exchangeability, causal graphs, Pearl’s Back-Door Criterion, and the construction of causal graphs from program dependence graphs.

3.14 Understanding the Relationship Between Recent Fault-Localization Techniques and Causality

George K. Baah (Georgia Tech, US)

License  Creative Commons BY 3.0 Unported license
© George K. Baah

Several dynamic analysis techniques have been developed to find the location of faults in programs. The techniques include experimental approaches such as Delta debugging and statistical approaches such as Tarantula. In this talk, I will show analytically the limitations of the techniques (e.g., Tarantula, Ochiai) that rely on statistical metrics for solving the fault-localization problem. I will then present empirical results supporting the analytical results and the challenges that must be overcome to accurately find the causes of software failures.

3.15 SEMFIX: Automated Program Repair using Semantic Analysis

Abhik Roychoudhury (National University of Singapore, SG)

License © Creative Commons BY 3.0 Unported license
© Abhik Roychoudhury

Joint work of Nguyen, H; Qi, Dawei; Roychoudhury, Abhik; Chandra, Satish
Main reference H. Nguyen, D. Qi, A. Roychoudhury, S. Chandra, “SEMFIX: Program Repair via Semantic Analysis,” in Proc. of the 35th Int’l Conf. on Software Engineering (ICSE’13), pp. 772–781, IEEE/ACM, 2013.

URL <http://dl.acm.org/citation.cfm?id=2486890>

Debugging consumes significant time and effort in any major software development project. Moreover, even after the root cause of a bug is identified, fixing the bug is non-trivial. Given this situation, automated program repair methods are of value. In this work, we present an automated repair method based on symbolic execution, constraint solving and program synthesis. In our approach, the requirement on the repaired code to pass a given set of tests is formulated as a constraint. Such a constraint is then solved by iterating over a layered space of repair expressions, layered by the complexity of the repair code. We compare our method with recently proposed genetic programming based repair on SIR programs with seeded bugs, as well as fragments of GNU Coreutils with real bugs. On these subjects, our approach reports a higher success-rate than genetic programming based repair, and produces a repair faster.

[Day 2] Session: Teaching debugging [Tue, Feb 05, 15:30-16:30]

3.16 Teaching Debugging

Andreas Zeller (Universität des Saarlandes, DE)

License © Creative Commons BY 3.0 Unported license
© Andreas Zeller

URL <http://www.whyprogramsfail.com/>

I present a course on debugging, based on an online course I developed for the online education pioneer Udacity. The course comes in six units:

1. How Debuggers Work (How failures come to be / The scientific method / Interactive debugging)
2. Asserting Expectations (Preconditions, postconditions, invariants / Inferring invariants)
3. Simplifying Failures (Delta debugging / Simplifying fuzz inputs / Causes and causality)
4. Tracking Origins (The art of deduction / Dependencies / Slices)
5. Reproducing Failures (Capturing inputs / Capturing coverage / Statistical debugging)
6. Learning from Mistakes (Bug reports / Bug distributions / Mining software repositories)

I combined this online course with six-week student projects at my university. Students were working on topics as diverse as delta debugging on LaTeX, novel combinations of debugging techniques, or fuzzers for SQL and Python. The key to having students do such large tasks in only six weeks was to use a language like Python, which allows easy access to the interpreter.

The main take away points are:

- Online courses are effective
 - Videos for individual pace; Quizzes for continuous self-assessment
 - But: Presenting or discussing scientific literature does not fit well (reading groups!)

- Online courses are efficient
 - Time spent in classroom is better spent on working with students directly
- Python is a great language for tracing
 - Assertions, invariants, delta debugging, statistical debugging, etc. are easy
 - But: Little support for dynamic dependencies and symbolic reasoning
- Python is great for experimenting
 - Delta debugging on states: 90 lines, 3 hours. (Read this again and again.)
- Projects unleash student creativity
 - Attract students to your research topics

[Day 3] Session: Solvers and Satisfiability [Wed, Feb 06, 09:00-10:30]

3.17 SAT Solvers for Software Reliability, Security and Repair

Vijay Ganesh (University of Waterloo, CA)

License © Creative Commons BY 3.0 Unported license
© Vijay Ganesh

In recent years SAT solvers have had a huge impact on software engineering research and practice. The reason for this is the impressive improvement in the efficiency of SAT solvers and their use as part of SMT solvers (e.g., DPLL(T) and as a backend for solvers for theories of bit-vectors and arrays). In this talk, I will highlight the 4 heuristics that have played a key role in this dramatic improvement in SAT solver performance, namely, 1) conflict-driven clause-learning with backjumping, 2) branching heuristics, 3) restarts and 4) efficient implementation of Boolean constant propagation.

I will also introduce my work on programmatic SAT solvers, where a user can specialize the SAT solver to their specific domain using an API that allows them to add code to influence the SAT solver's search and branching heuristics. This idea is a variation on the idea of DPLL(T), and I will highlight the differences between the two.

3.18 Fault Localization using Maximum Satisfiability

Rupak Majumdar (MPI for Software Systems – Kaiserslautern, DE)

License © Creative Commons BY 3.0 Unported license
© Rupak Majumdar

Joint work of Majumdar, Rupak; Jose, Manu

Main reference M. Jose. R. Majumdar, "Cause clue clauses: error localization using maximum satisfiability," in Proc. of the 32nd ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI'11), pp. 437–446, ACM, 2011.

URL <http://dx.doi.org/10.1145/1993498.1993550>

Several verification tools exist for checking safety properties of programs and reporting errors. However, a large part of the program development cycle is spent in analyzing the error trace to isolate locations in the code that are potential causes of the bug. Currently, this is usually performed manually, by stepping through the error trace in a debugger. We present Bug-Assist, a tool that assists programmers localize error causes to a few lines of code. Bug-Assist takes as input an ANSI-C program annotated with assertions, performs model checking internally to find potential assertion violations, and for each error trace returned by

the model checker, returns a set of lines of code which can be changed to eliminate the error trace. Bug-Assist’s algorithm formulates error localization as a MAX-SAT problem and uses scalable MAX-SAT solvers.

We discuss a set of open problems related to this idea, for example, the use of similar techniques for compiler bugs, for regression testing, for concurrency testing, as well as possible techniques for scaling up to large programs.

[Day 3] Session: Repair 3 [Wed, Feb 06, 11:00-12:15]

3.19 On the biodiversity of source code: rigid or plastic repair?

Benoit Baudry (INRIA Bretagne Atlantique – Rennes, FR), Martin Monperrus (INRIA Nord Europe – Lille, FR)

License © Creative Commons BY 3.0 Unported license

© Benoit Baudry, Martin Monperrus

Joint work of Baudry, Benoit; Monperrus, Martin

Main reference M. Monperrus, M. Mezini, “Detecting Missing Method Calls as Violations of the Majority Rule”, ACM Transactions on Software Engineering and Methodology (TOSEM), 22(1), pp. 7:1–7:25, 2013.

URL <http://dx.doi.org/10.1145/2430536.2430541>

In this talk, we first present a static code analysis that finds a specific kind of bug (missing method calls when using third-party frameworks) based on a radical abstraction over the code (type usages: a list of method calls on a variable of a given type). Furthermore, the analysis provides a partial diagnostic on how to repair the bug. Those diagnostics enabled us to provide valid patches on a million LOC unknown code base – valid in the sense that they were incorporated in the code base by the lead developers. Further experiments were done to understand the dynamics of type-usages at the code ecosystem level. Those experiments revealed that some Java library classes give birth to many different type usages that we call “usage diversity”. A high “usage diversity” indicates a great “plasticity” of the class itself, and we have empirical evidence of a relation between the usage diversity of a class and its success (the class is used by many client classes). Those novel observations open new research questions: how does the code plasticity relate to the kind of repair techniques? how to step from code diversity to automated code diversification?

3.20 Leveraging Software Text Analytics To Detect, Diagnose, and Fix Bugs

Lin Tan (University of Waterloo, CA)

License © Creative Commons BY 3.0 Unported license

© Lin Tan

Joint work of Liu, Chen; Yang, Jinqiu; Tan, Lin; Hafiz, Munawar

Main reference C. Liu, J. Yang, L. Tan, M. Hafiz, “R2Fix: Automatically Generating Bug Fixes from Bug Reports,” in Proc. of the 6th IEEE Int’l Conf. on Software Testing, Verification and Validation (ICST’13), March, 2013, to appear.

URL <https://ece.uwaterloo.ca/~lintan/publications/r2fix-icst13.pdf>

Software bugs seriously hurt software reliability. In this talk, I will present our recent research on leveraging software textual information in program comments, source code, and bug reports to detect, diagnose, and fix software bugs. R2Fix automatically generates bug-fixing patches from bug reports written free-form in a natural language. R2Fix combines past

fix patterns, machine learning techniques, and semantic patch generation techniques to fix bugs automatically. We evaluate R2Fix on three projects, i.e., the Linux kernel, Mozilla, and Apache, for three important types of bugs: buffer overflows, null pointer bugs, and memory leaks. R2Fix generates 57 patches correctly, 5 of which are new patches for bugs that have not been fixed by developers yet. We reported all new patches to the developers; 4 have already been accepted and committed to the code repositories. The 57 correct patches generated by R2Fix could have shortened the bug fixing time by up to 63 days on average. In addition, they could save developers' time and effort in diagnosing and fixing bugs. R2Fix is safe as patches are not applied until developers have confirmed the correctness of the patches. iComment, aComment, and @tComment automatically extract specifications from source code and code comments written in a natural language, and use these specifications to detect comment-code inconsistencies, i.e., software bugs and bad comments.

[Day 4] Session: Program Analysis 1 [Thu, Feb 07, 09:00-10:45]

3.21 Genetic Mutation Conditioned Amorphous Parametric Hybrid “Dual” Slicing

Jens Krinke (University College London, GB)

License  Creative Commons BY 3.0 Unported license
© Jens Krinke

This talk highlights the most important ideas, problems, and solutions in the history of program slicing. It turns out that most of the technical problems have been solved in the past 10 years while there have not been a lot of advances in the academic research on the long standing challenges.

In addition, the talk presents current trends in the usage of dependence analysis and program slicing. It shows how dependence form large clusters of statements that are indistinguishable in terms of impact. The talk also shows how slicing can be used in information flow control to check security policies including declassification.

3.22 Information Flow Analysis for JavaScript

Christian Hammer (Universität des Saarlandes, DE)

License  Creative Commons BY 3.0 Unported license
© Christian Hammer

Joint work of Just, Seth; Cleary, Alan; Shirley, Brandon; Hammer, Christian

Main reference S. Just, A. Cleary, B. Shirley, C. Hammer, “Information flow analysis for JavaScript,” in Proc. of the 1st ACM SIGPLAN Int'l Workshop on Programming Language and Systems Technologies for Internet Clients (PLASTIC'11), pp. 9–18, ACM, 2011.

URL <http://dx.doi.org/10.1145/2093328.2093331>

Modern Web 2.0 pages combine scripts from several sources into a single client-side JavaScript program with almost no isolation. In order to prevent attacks from an untrusted third-party script or cross-site scripting, tracking provenance of data is imperative. However, currently no browser offers this security mechanism. This work presents the first information flow control mechanism for full JavaScript. We track information flow dynamically as much as possible but rely on intra-procedural static analysis to capture implicit flow. Our analysis

handles even the dreaded eval function soundly and incorporates flow based on JavaScript’s prototype inheritance. We implemented our analysis in a production JavaScript engine and report both qualitative as well as quantitative evaluation results.

3.23 Gaining inSight into Programs that Analyze Programs – By Visualizing the Analyzed Program

Mangala Gowri Nanda (IBM India Research Lab. – New Delhi, IN)

License © Creative Commons BY 3.0 Unported license
© Mangala Gowri Nanda

Joint work of Nanda, Agastya; Nanda, Mangala Gowri

Main reference A. Nanda, M. Gowri Nanda, “Gaining insight into programs that analyze programs: by visualizing the analyzed program,” in Proc. of the 24th ACM SIGPLAN Conf. Companion on Object Oriented Programming Systems Languages and Applications, pp. 1023–1030, ACM, 2009.

URL <http://dx.doi.org/10.1145/1639950.1640074>

Visualization of a program typically entails low level views of the program execution state showing, for example, method invocations or relations amongst heap objects. In most cases, this would imply visualization of the executable program. However there is a certain genre of programs that analyze or transform other programs. These programs could be compilers, static bug detectors, test suite analyzers, model to model transformers etc. In such cases, very often, it helps to visualize what is happening to the input program rather than the analyzer program. It is for such programs that we describe a configurable, analysis framework. For ease of exposition, we call the analyzer program the “manipulate” program, and the input program the “puppet” program. To facilitate the visualization, we instrument the manipulate program to generate a dump as it analyzes the puppet program. Using the “dump”, we reconstruct the interprocedural control flow graph of the puppet program and then visualize the flow of the manipulate program over the puppet program. In particular, our visualization consists of rendering the control flow graph of the puppet program, and then tracking the behavior of the manipulate program by watching its interaction with the puppet program. We use colors to highlight different events in the manipulate program. Using this scheme, we are able to (1) gain insight into the manipulate program; (2) collect useful information / statistics about the puppet program. We have implemented the visualizer in a tool called “Insight”. We ran Insight on a static debugging tool (the manipulate program) called Xylem. Xylem applies static analysis to find potential null pointer exceptions in a puppet program, as for example, the Apache Ant program. We report the insights gained by running Xylem through Insight on Ant and other puppet programs.

[Day 4] Session: Program Analysis 2 [Thu, Feb 07, 11:15-12:15]

3.24 Blended Taint Analysis for JavaScript

Barbara G. Ryder (Virginia Polytechnic Institute – Blacksburg, US)

License © Creative Commons BY 3.0 Unported license
© Barbara G. Ryder

Joint work of Ryder, Barbara G.; Wei, Shiyi

JavaScript is ubiquitous in website code, introducing challenges to static analyses by its heavily used dynamic features such as dynamic code generation and variadic functions. Taint analysis is an important tool for finding data integrity problems in programs. The blended analysis paradigm combines a dynamic representation of program calling structure , with

static analysis applied to that calling structure. Recently, we developed a blended taint analysis for JavaScript website code that is more scalable and more accurate than a pure static analysis. Because taint analysis is posed as reachability on a program representation between tainted sources and sensitive sinks, there is the possibility of the analysis gathering a witness path for each reported vulnerability.

3.25 Accurate and Scalable Security Analysis of Web Applications

Marco Pistoia (IBM TJ Watson Research Center – Hawthorne, US)

License © Creative Commons BY 3.0 Unported license
© Marco Pistoia

Joint work of Pistoia, Marco; Omer Tripp; Patrick Cousot; Radhia Cousot; Salvatore Guarnieri

Security auditing of industry-scale software systems mandates automation. Static taint analysis enables deep and exhaustive tracking of suspicious data flows for detection of potential leakage and integrity violations, such as cross-site scripting (XSS), SQL injection (SQLi) and log forging. Research in this area has taken two directions: program slicing and type systems. Both of these approaches suffer from a high rate of false findings, which limits the usability of analysis tools based on these techniques. Attempts to reduce the number of false findings have resulted in analyses that are either (i) unsound, suffering from the dual problem of false negatives, or (ii) too expensive due to their high precision, thereby failing to scale to real-world applications. In this paper, we investigate a novel approach for enabling precise yet scalable static taint analysis. The key observation informing our approach is that taint analysis is a demand-driven problem, which enables lazy computation of vulnerable information flows, instead of eagerly computing a complete data-flow solution, which is the reason for the traditional dichotomy between scalability and precision. We have implemented our approach in ANDROMEDA, an analysis tool that computes data-flow propagations on demand, in an efficient and accurate manner, and additionally features incremental analysis capabilities. ANDROMEDA is currently in use in a commercial product. It supports applications written in Java, .NET and JavaScript. Our extensive evaluation of ANDROMEDA on a suite of 16 production-level benchmarks shows ANDROMEDA to achieve high accuracy and compare favorably to a state-of-the-art tool that trades soundness for precision.

[Day 4] Session: Mending the Web [Thu, Feb 07, 13:30-15:00]

3.26 An Overview of the Apollo Project: Test Generation and Fault Localization for Web Applications

Shay Artzi (IBM – Littleton, US), Frank Tip (University of Waterloo, CA), and Julian Dolby (IBM TJ Watson Research Center – Hawthorne, US)

License © Creative Commons BY 3.0 Unported license
© Shay Artzi, Frank Tip, and Julian Dolby

Joint work of Tip, Frank; Dolby, Julian; Artzi, Shay; Pistoia, Marco

The Apollo project at IBM Research was aimed at developing practical automated techniques for finding, localizing, and fixing bugs in web applications. We adapted an existing dynamic test generation technique that applies concrete and symbolic execution to the domain of web applications written in PHP, and used it to find dozens of failures in open-source PHP

applications. To help programmers with localizing the faults that cause these failures, we adapted existing fault localization techniques that predict in which statements a fault is located by applying a statistical analysis to execution data gathered from multiple tests. Our results indicate that, using our best technique, nearly 90% of faults are localized to within 1% of all executed statements. We also address the question of how to localize a fault when the programmer is confronted with a failure but no test suite is available that can be used for fault localization. In such cases, our new directed test generation technique is capable of generating small test suites with high fault-localization effectiveness.

This presentation is based on papers published at ISSTA'08, ICSE'10, ISSTA'10, IEEE TSE'10 and IEEE TSE'12.

3.27 Automated Repair of HTML Generation Errors in PHP Applications Using String Constraint Solving

Hesam Samimi (Univ of California – Los Angeles, US)

License © Creative Commons BY 3.0 Unported license
© Hesam Samimi

Joint work of Schäfer, Max; Artzi, Shay; Millstein, Todd; Tip, Frank; Hendren, Laurie

Main reference H. Samimi, M. Schäfer, S. Artzi, T. Millstein, F. Tip, L.e Hendren, “Automated repair of HTML generation errors in PHP applications using string constraint solving,” in Proc. of the 2012 Int'l Conf. on Software Engineering (ICSE'12), pp. 277–287, IEEE Press, 2012.

URL <http://dl.acm.org/citation.cfm?id=2337223.2337257>

PHP web applications routinely generate invalid HTML. Modern browsers silently correct HTML errors, but sometimes malformed pages render inconsistently, cause browser crashes, or expose security vulnerabilities. Fixing errors in generated pages is usually straightforward, but repairing the generating PHP program can be much harder. We observe that malformed HTML is often produced by incorrect “string literal prints”, i.e., statements that print string literals, and present two tools for automatically repairing such HTML generation errors. PHPQuickFix repairs simple bugs by statically analyzing individual prints. PHPRepair handles more general repairs using a dynamic approach. Based on a test suite, the property that all tests should produce their expected output is encoded as a string constraint over variables representing string literal prints. Solving this constraint describes how string literal prints must be modified to make all tests pass. Both tools were implemented as an Eclipse plugin and evaluated on PHP programs containing hundreds of HTML generation errors, most of which our tools were able to repair automatically.

[Day 4] Session: Miscellaneous [Thu, Feb 07, 15:30-16:30]

3.28 Demo of the Tarantula Fault-Localization Tool

Jake Cobb (Georgia Tech, US)

License © Creative Commons BY 3.0 Unported license
© Jake Cobb

Joint work of Cobb, Jake; Harrold, Mary Jean; Jones, James A.

URL <http://aristotleresearch.com/>

I will give a demo of an implementation of the Tarantula statistical fault-localization technique. It includes fault-localization calculations using a modification of the Ochiai formula, a SeeSoft visualization of results, navigation based on the fault-localization results, and a fault-localization-based test-case clustering algorithm. The tool currently supports Java programs

using the popular JUnit testing framework. The data import step supports integration with both Ant and Maven, two of the dominant build tools for Java projects. Finally, coverage information can be obtained from either the proprietary Clover or open-source Cobertura coverage tools.

3.29 Three Projects Related to Fault Prediction, Localization, and Repair

Milos Gligoric (University of Illinois – Urbana, US)

License © Creative Commons BY 3.0 Unported license
© Milos Gligoric

This talk briefly presents three projects from our research group that are closely related to the three topics of the seminar: instead of “Fault Prediction”, “Fault Localization”, and “Fault Repair”, I will present “Failure Prediction”, “Dependency-Aware Fault Localization”, and “Test Repair”. In “Failure Prediction” we estimate the likelihood of triggering a failure in a specific part of a system by running the system with a large number of real-world inputs. The results of applying this technique on the Eclipse refactoring engine estimate the number of failures for each refactoring and show that the Eclipse refactoring engine is less buggy than we expected. In “Dependency-Aware Fault Localization” we use knowledge about changes of the system under test to improve fault localization by identifying unchanged statements that cannot affect the results of any test. The results on a number of programs demonstrate that dependency-aware fault localization can reduce the number of statements to be inspected before the fault is identified. In “Test Repair” we present a solution for repairing broken unit tests after the system under test is changed by analyzing dynamic execution of the tests. The results show that this approach can repair many common test failures and that its suggested repairs match developers’ expectations.

[Day 5] Session: Concurrency [Fri, Feb 08, 09:30-10:30]

3.30 Automated Concurrency-Bug Fixing

Ben Liblit (University of Wisconsin–Madison, US)

License © Creative Commons BY 3.0 Unported license
© Ben Liblit

Joint work of Jin, Guoliang; Zhang, Wei; Deng, Dongdong; Liblit, Ben; Lu, Shan

Main reference G. Jin, W. Zhang, D. Deng, B. Liblit, S. Lu, “Automated Concurrency-Bug Fixing,” in Proc. of the 10th USENIX Symp. on Operating Systems Design and Implementation (OSDI’12), pp. 221–236, USENIX Assoc., 2012.

URL <https://dl.acm.org/citation.cfm?id=2387880.2387902>

Concurrency bugs are widespread in multithreaded programs. Fixing them is time-consuming and error-prone. We present CFix, a system that automates the repair of concurrency bugs. CFix works with a wide variety of concurrency-bug detectors. For each failure-inducing interleaving reported by a bug detector, CFix first determines a combination of mutual-exclusion and order relationships that, once enforced, can prevent the buggy interleaving. CFix then uses static analysis and testing to determine where to insert what synchronization operations to force the desired mutual-exclusion and order relationships, with a best effort to avoid deadlocks and excessive performance losses. CFix also simplifies its own patches by merging fixes for related bugs.

Evaluation using four different types of bug detectors and thirteen real-world concurrency-bug cases shows that CFix can successfully patch these cases without causing deadlocks or excessive performance degradation. Patches automatically generated by CFix are of similar quality to those manually written by developers.

3.31 Localizing Non-deadlock Concurrency Bugs

Mary Jean Harrold (Georgia Tech, US)

License © Creative Commons BY 3.0 Unported license
© Mary Jean Harrold

Joint work of Park, Sangmin; Vuduc, Richard; Harrold, Mary Jean

Main reference S.Park, R. Vuduc, M.J. Harrold, “A Unified Approach for Localizing Non-deadlock Concurrency Bugs,” in Proc. of the 5th IEEE Int’l Conf. on Software Testing, Verification and Validation (ICST’12), pp. 51–60, IEEE, 2012.

URL <http://dx.doi.org/10.1109/ICST.2012.85>

In this talk, I will present our automated dynamic fault-comprehension technique, that provides a way to explain concurrency bugs with additional information over existing fault-localization techniques, and thus, bridges the gap between fault-localization and fault-fixing techniques. The technique inputs a list of memory-access patterns and a coverage matrix, groups those patterns responsible for the same concurrency bug, and outputs the grouped patterns along with suspicious methods and bug graphs. Griffin is the first technique that handles multiple concurrency bugs. I will also describe the implementation of our technique in Java and C++, and show the empirical evaluation which shows results show that our technique clusters failing executions and memory-access patterns for the same bug with few false positives, provides suspicious methods that contain the locations to be fixed, and runs efficiently.

4 Discussion and Open Problems

The main theme of the discussion was that techniques for fault prediction, localization, and repair as presented at this Seminar were already quite advanced, but their adoption, in industry especially, was lagging behind. One usual suspect for this is the lack of large-scale empirical studies demonstrating the usefulness of the various approaches, but this would require controlled studies, which are inherently difficult to conduct with the limited (especially monetary) resources of a computer science department. Another probable cause is a lack of (continued) education or, rather, the general lag time between the discovery and publication of a new insight, and its widespread adoption in industry.

5 Conclusion

The workshop provided a productive venue for an open exchange of ideas about various topics related to fault prediction, localization, and repair. Overall, we felt that there was an excellent balance of topics and mix of participants. The format of the workshop went well. We deliberately avoided over-scheduling by keeping the presentations relatively short (with the exception of a few well-chosen tutorial-style presentations), and long breaks between sessions. These breaks provided plenty of time for fruitful interactions between the meeting participants. Overall, we consider the workshop to have been very successful, and one of the most enjoyable and productive Dagstuhl workshops we have attended.

Participants

- Rui Abreu
University of Porto, PT
- Shay Artzi
BM – Littleton, US
- George K. Baah
Georgia Inst. of Technology, US
- Benoit Baudry
INRIA Bretagne Atlantique –
Rennes, FR
- Margaret M. Burnett
Oregon State University, US
- Satish Chandra
IBM India – Bangalore, IN
- Jake Cobb
Georgia Inst. of Technology, US
- Julian Dolby
IBM TJ Watson Research Center
– Hawthorne, US
- Marcus Frenkel
FernUniversität in Hagen, DE
- Vijay Ganesh
University of Waterloo, CA
- Milos Gligoric
Univ. of Illinois – Urbana, US
- Alessandra Gorla
Universität des Saarlandes, DE
- Mangala Gowri Nanda
IBM India Research Lab. –
New Delhi, IN
- Christian Hammer
Universität des Saarlandes, DE
- Mary Jean Harrold
Georgia Inst. of Technology, US
- Jens Krinke
University College London, GB
- Ben Liblit
University of Wisconsin –
Madison, US
- Rupak Majumdar
MPI for Software Systems –
Kaiserslautern, DE
- Martin Monperrus
INRIA Nord Europe – Lille, FR
- Alessandro Orso
Georgia Inst. of Technology, US
- Marco Pistoia
IBM TJ Watson Research Center
– Hawthorne, US
- Andy H. Podgurski
Case Western Reserve University
– Cleveland, US
- Jeremias Rößler
Universität des Saarlandes, DE
- Abhik Roychoudhury
National Univ. of Singapore, SG
- Barbara G. Ryder
Virginia Polytechnic Institute –
Blacksburg, US
- Hesam Samimi
Univ. California –
Los Angeles, US
- Friedrich Steimann
Fernuniversität in Hagen, DE
- Lin Tan
University of Waterloo, CA
- Frank Tip
University of Waterloo, CA
- Emina Torlak
University of California –
Berkeley, US
- Cemal Yilmaz
Sabanci Univ. – Istanbul, TR
- Andreas Zeller
Universität des Saarlandes, DE
- Xiangyu Zhang
Purdue University, US
- Thomas Zimmermann
Microsoft Res. – Redmond, US



First row: Marcus, Mary Jean, Jake, Mangala, Barbara, Lin, Margaret, Alessandra.
Second row: Jeremias, Milos, Vijay, Shay, Abhik, Christian, Cemal, Xiangyu.
Third row: Alex, Emina, Frank, Satish, Rui, Ben, Julian, Benoit.
Last row: Jens, Andreas, Thomas, Marco, Andy, Friedrich, Hesam, George, Rupak, Martin.

Decentralized Systems for Privacy Preservation

Edited by

Sonja Buchegger¹, Jon Crowcroft², Balachander Krishnamurthy³,
and Thorsten Strufe⁴

- 1 KTH Royal Institute of Technology - Stockholm, SE, buc@csc.kth.se
- 2 University of Cambridge, GB, Jon.Crowcroft@cl.cam.ac.uk
- 3 AT&T Labs–Research – Florham Park, US, bala@research.att.com
- 4 TU Darmstadt, DE, strufe@cs.tu-darmstadt.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13062 “Decentralized Systems for Privacy Preservation”. In recent years, a number of concerns have risen about the existence of large, organizationally centralized online services (cloud services, online social networks, repositories, etc). The concerns include risks to users’ data from organizational failures and threats to user privacy. In this seminar, the organizers brought together a somewhat more diverse collection of theoreticians and practitioners from industry and academia including social scientists and economists. In keeping with the nature of the interdisciplinary attendees, the organizers also attempted a seminar organization structure intended to promote innovative, cross-discipline working. The results were mixed: some clear agenda setting outputs emerged with some less clear ones.

Seminar 03.–08. February, 2013 – www.dagstuhl.de/13062

1998 ACM Subject Classification K.4.1 Public Policy Issues: Privacy, K.6.4 System Management: Centralization/decentralization, D.4.6 Security and Protection.

Keywords and phrases Privacy, Decentralized Systems, Economics, Usability, Mobility

Digital Object Identifier 10.4230/DagRep.3.2.22


1 Executive Summary

Sonja Buchegger

Jon Crowcroft

Balachander Krishnamurthy

Thorsten Strufe

License  Creative Commons BY 3.0 Unported license
© Sonja Buchegger, Jon Crowcroft, Balachander Krishnamurthy, and Thorsten Strufe

Distributed and decentralized systems offer more potential resilience to various failures, and, on paper, higher aggregate availability than centralized systems. Centralized management repositories lead to potential risks to users’ privacy and the temptation to monetize processing of large aggregates of such data, as seen in systems such as webmail, search and online social networks. Recent years have seen the emergence of projects building prototypes with varying levels of decentralization to reduce these risks. Such systems have not seen great success in contrast to large cloud services. This seminar brought together diverse groups to tackle a series of questions to attempt to answer what may be the root causes of the logjam preventing success of these alternative approaches. There appears to be some consensus amongst at least some groups that there are good reasons for these alternatives. We present here the



Except where otherwise noted, content of this report is licensed
under a Creative Commons BY 3.0 Unported license

Decentralized Systems for Privacy Preservation, *Dagstuhl Reports*, Vol. 3, Issue 2, pp. 22–44

Editors: Sonja Buchegger, Jon Crowcroft, Balachander Krishnamurthy, and Thorsten Strufe



Dagstuhl Reports

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

output of our group working sessions on these questions. We also provide the reasoning and outcomes of the discussions along with an evaluation of the effectiveness of our mode of working in this seminar.

2 Table of Contents

Executive Summary

Sonja Buchegger, Jon Crowcroft, Balachander Krishnamurthy, and Thorsten Strufe 22

Seminar Plan

Subject Introduction 25

Position Papers 25

Work Sessions 26

Reflections on the Seminar Setup 26

Challenge: Users and Usability

Background 27

Discussion 28

Challenge: Economics

Background 31

Discussion 32

Challenge: Technology

Background 35

Discussion 36

General Observations, Questions Raised, Open Problems 42

Participants 44

3 Seminar Plan

3.1 Subject Introduction

Centralized collections of user data have threatened privacy due to data mining and intentional or accidental data leakage to third parties. Online social networks and social media sites are prominent examples, as they attract the lion's share of the Internet users' time today. These recent, Web-based services frequently provide comprehensive personalization, aiming at a precise identification of the individuals using them. While offering valuable services to the individuals on the Web, they collect large amounts of information about the users, including the content willingly uploaded by the users themselves, but more importantly patterns in their preferences and behavior as well as relations to others. All this personally identifiable information is concentrated at a small set of companies that are logically centralized service providers. Large collections of extensive, detailed personal information are needed by these providers, since their exploitation, primarily for targeted advertisement, represents their main business model.

Numerous attempts have been undertaken to counter threats regarding the centralized collection of information. One promising approach is to create services such as online social networks, private data storage and backup, or anonymous content dissemination in a distributed fashion, thus removing the centralized provider with all its knowledge and power. Typically, the gatekeeper functionality of the centralized service provider is replaced by using cryptographic means for access control, metadata-minimizing system design, and other privacy-enhancing technologies to prevent unauthorized data leakages.

While there have been advances on the technical front for decentralized social networks, usability and user acceptance are a challenge. Economics remains a key issue to address head on for any decentralized approach to work. A decentralized approach to privacy-preserving systems inherently means a paradigm shift from today's, mostly Web-based, services. This shift opens a range of research questions in terms of Computer Science (feasibility, scalability, security, new privacy challenges, robustness, resource allocation, resource heterogeneity, efficiency, mobility, etc.) and other disciplines such as Economics, Law, Policy Research, and Sociology. Considering the vast acceptance and ubiquity of these services and their impact on the daily life of individuals, decentralization for privacy is not limited to academic research but needs contributions from other parts of society, such as industry, activists, communities, and policy makers.

We see a number of challenges to be overcome when pursuing the idea of decentralization as a means to increasing the control and privacy of the users. Serving as nuclei for discussions at the seminar, we divided them into three grand challenges User Challenge, Economic Challenge, and Technological Challenge.

3.2 Position Papers

Before the seminar, we sent out a call for one-page position papers to all the participants. The call included the following prompt for the challenges identified for privacy-preserving decentralized systems.

The users. Acceptance and adoption of decentralized services requires a critical mass of users who expect a constant flow of recent, rich content, which conflicts with privacy preservation. How can awareness be raised and to what extent will users actually care about their privacy is one challenge.

Economics of decentralized systems. Centralized systems are highly profitable and monetizing personal data subsidizes such services. Sustainable decentralized systems without monetization potential remains an unresolved challenge. Privacy-preserving queries or introducing differential privacy may help but there are no functional systems yet.

Technological feasibility. Systems requirements like availability, scalability, and robustness or mobility have been discussed but the security of such designs is not well understood. How can functional extensions, such as recommendation schemes, be implemented without access to the entirety of user data and behavior?

The position papers were collected and made available to all participants, with a request to take them into account when introducing themselves and their research briefly at the beginning of the seminar. We asked the participants to state which position paper they resonated, agreed, and disagreed with the most. Additionally, the position papers served to set the scene before the seminar and to extend the challenges and questions already present in the seminar proposal.

With input from the position papers written by the invited participants, the three aspects were further detailed, thus extending the User Challenge to *Users and Usability*, and the Economic Challenge to additionally address *social bootstrapping* (or: roll-out) and incentives.

3.3 Work Sessions

The seminar was organized in sessions of team work, rather than a sequence of presentations with subsequent discussion. Relevant questions for the seminar were identified in an initial expert session and then used for dividing the participants into groups on the three aspects. The seminar addressed each of the aspects separately, discussing economical aspects and users/usability aspects in one separate session, each, and the technological aspects in two sessions.

Mixed sets of experts and other participants were grouped by interest to discuss one of the identified questions from the respective topic and to report their results in a plenary session. The experts on each aspect finally collected these outcomes and established conclusions for their field from the seminar, which are reported in the following sections.

The reasoning behind the division into two types of sessions, called expert and mixed sessions, was to strike a balance between interdisciplinary exchange and deeper discussions between those of similar expertise. This allowed the participants to both learn from perspectives from other fields and to discuss without having to explain a lot of background. The two expert sessions bracketed the set of mixed sessions. The first session of experts had the task to come up with questions to be discussed in the subsequent mixed sessions. The second expert session took place after the mixed sessions had come up with some answers to the posed questions. The task of this expert session was to synthesize what had come out of the mixed sessions and draw conclusions. The mixed sessions in between had participants who joined the groups according to their interest and discussed specific questions.

3.4 Reflections on the Seminar Setup

The structure of the seminar deviated from the canonical Dagstuhl seminar in that there were very few talks, a 5-min no-slide introduction by everyone and few individual presentations by participants. The program instead mostly consisted of interactive group sessions with

discussions on specific topics followed by short presentations of the outcome of these sessions in the plenary. The goal was to encourage knowledge exchange, engaging and coming up with ideas and questions for new research.

While this was certainly more interactive and resulted in some new ideas, it might, on reflection, have been excessive in the emphasis on interaction. What we observed and was mentioned in some of the comments in the seminar evaluation survey pointed to people getting saturated with interaction all day almost daily. It might have been good to switch the mode to presentations even if not as frequently as at a typical Dagstuhl seminar. One effect of the interactive day seemed to be that, at least in some cases, energy that would have been left for informal research discussions over dinner or after, had already been channeled into the work sessions.

To kill two birds with one stone, a set of (reasonably short) tutorials by domain experts for example in economics or other non-Computer-Science fields could provide more background information for group discussions to start at a more advanced level while allowing for some less interactive time.

The position papers elicited before the start of the seminar proved to be very helpful to get an overview of the participants' interests and research. They were appreciated, both by the organizers to distill questions and by the participants themselves to engage with each others' work and perspectives.

Grouping participants by expertise and interest, both self-organized with some slight load-balancing by the organizers when warranted, worked for the most part yet also failed to break up some cliques of people that tended to choose the same groups in several sessions. One clique discussed items not part of the agenda reducing their overall contribution to the seminar. To avoid this, group member allocation might be more strongly enforced by the organizers when needed.

4 Challenge: Users and Usability

Topics expected from proposal: Acceptance, adoption, and usability

Additional topics raised in position papers: transparency, trustworthiness, understandability and take-it-or-leave it EULAs with unnoticed changes, tracking of data disclosure

4.1 Background

Centralized Social Networking Services enable communication between a group of users, sharing and browsing of content. Since they can collect data about user's actions, they can generate detailed digital dossiers and aggregated profiles of their users, which can be used for targeted advertisement.

The survival and growth of online social networks requires a minimum amount of appealing and fresh content to insure their attractiveness and to guaranteeing frequent return visits of users. They rely on the contribution of participants, publishing their details, opinions, and other user-generated content, and on the visibility of this content to others. Active sharing is commonly limited to a comparably small number of active contributors, whereas the lion's share are primarily passive users who only occasionally, motivated by other posts, share some

content, or comment on discussions. It is essential for the services to reach a critical mass of contributors, and contributions visible to the common user to remain attractive.

User acceptance consequently partly depends on a wide adoption, winning enough active contributors of attractive content, and partly on the availability, and visibility of their contributions. This conflicts with the objective to protect the participants' privacy and to restrict sharing of personally identifiable information to a limited group of actually trusted individuals, only. With smaller user bases, better privacy protection, and lower volumes of attractive content for passive users, a challenge for decentralized, privacy preserving approaches is going to be to attract sufficiently large user bases and available content.

Decentralized approaches may have further shortcomings as compared to commercial services, which exacerbate this situation. Lacking funds to employ professional manpower, they may be characterized by lower usability, fewer extensions, such as games and social apps, and infrequent maintenance to improve their appearance, and absence of marketing campaigns.

Additional protection of each individual's privacy may increase the appeal of such approaches. The question to which extent the average participant of an online social network actually cares about their privacy, and hence, how much the protection of privacy would actually make up for a less content and lower usability remains open.

Beyond the protection from observation by a provider, eliminating centralized storage and control additionally fosters freedom of speech. This freedom, however, raises further challenges, since a lack of control not only prevents censorship, but also deprives the system of the possibility to deal with unwanted, or illegitimate content.

These challenges have to be further explored, to evaluate their impact and aim at potentially coming up with ideas for meeting them.

4.2 Discussion

The expert group on "Users and Usability" identified four questions for further discussions.

4.2.1 Question 1: How to reduce the gap between the complexity of the system and users' mental models?

The intent behind this question is to examine the mismatch between what users expect a system to do and what the system actually does. A good example is persistence of user data: drawing on from real life, users sometimes expect a conversation thread to be ephemeral, but are surprised to find out later that the data related to the conversation is long-lived. Many systems provide fine-grained access control for user data, but the level of expertise needed to properly configure and manage such access control is beyond most ordinary users.

The mixed group discussion recognized that the problem is exacerbated by the fact that different users have different levels of expertise, and a given user's expertise and hence their mental models have evolved over time. This led to the insight that the first step in solving the problem is to develop a reliable way to measure the gap between a user's mental model and the actual system functionality. This may be done implicitly by drawing inferences from user actions, or explicitly, by posing questions to the user. Both of these approaches are not straight-forward and would need to be designed carefully in order to pass the user-acceptance test. The end goal would be to have the system provide guidance to the user in an adaptive manner, depending on the user's expertise and what he/she expects the system to do.

4.2.2 Question 2: How to reduce of the risks of the user's data being misappropriated or used unintended?

The second question was to address the problem of user data being used in ways the user did not intend. This can happen unintentionally, as was the case when a private photo of a Zuckerberg family gathering intended to be shared just with friends became visible to a journalist because of the way Facebook access control works (a photo is visible to the friends of anyone who is tagged in addition to those intended by the sharer). The journalist assumed the photo to be public and shared it further. This example illustrates the consequences of the gap between the (journalist's) mental model and actual system functionality (Facebook's access control for photos with people tags).

Misappropriation can also happen intentionally, when a data collector who goes out of business decides to sell the data to third parties.

The mixed group discussion distinguished between two types of misappropriation.

The first, unintentional misappropriation by "honest but clueless" users can be addressed by improving usability of the sharing process. First, better tools that visualize the extent of sharing to users would alert them if they were about to share with a larger audience than they intend to. This is closely related to Question 1: the type of visualization needed will certainly depend on the user's expertise and mental model, and therefore needs to be adaptive. Second, better techniques for easily selecting the audience of a sharing action can help, too. For example, the data (who is in it) and metadata (where was it taken, who was nearby when it was taken) in a photo can help infer potential sharing targets.

The second, intentional misappropriation by malicious users or data collectors will require stronger protection mechanisms. Currently, the only mechanisms widely in use are legal and regulatory mechanisms which seek to ensure that data collected for one purpose cannot be used for another. There are two other potential approaches one can envisage. The first is the use of trusted hardware. Trusted Platform Modules are widely available for personal computers and server platforms. Hardware security mechanisms for smartphone, like ARM TrustZone, are even more widely deployed and in fact used to ensure features like secure boot. We can build on these features to allow remote platform attestation and configuration verification. Thus, before handing out sensitive data to a remote system, software running on behalf of a user can verify that the hardware and software configuration on the remote system is trustworthy and will not allow the misuse of data. Similarly, advanced cryptographic protocols like private information retrieval are now practical enough to be implemented even on smartphone platforms. Again, neither the use of trusted hardware nor systems that incorporate advanced cryptography is straight-forward. For example, both may limit the ability to run data analytics on user data which is currently used both for monetization (e.g., advertisements shown by Facebook) and for performance improvements (e.g., system health monitoring in systems like Tor).

4.2.3 Question 3: How can decentralized systems reduce the impact of power imbalance while not compromising usability?

Since the issue of decentralization did not crop up in the formulation of the other questions, the expert group discussed what effect decentralization has from the point of view of users and usability. They concluded that the primary difference that concerns the user is the imbalance in power which is manifest in all existing centralized systems: an online social network could arbitrarily change its privacy policies or usage of user data and the users have

no means of protest or debate other than by leaving the system. The network effects make even the possibility of leaving the system less of a free choice (by leaving the system, users run the risk of disrupting social contacts with people whom they care about as long as they remain in the system). This was the rationale for Question 3.

The mixed group concluded that decentralization per se is neither necessary nor sufficient to redress the power imbalance. The pre-requisites for reaching parity of power is (a) standardization of interfaces and (b) governance of the system by a neutral body. They did conclude that a decentralized system is more likely than a centralized system to reach power balance.

4.2.4 Question 4: How to mitigate collateral damage?

The last question was motivated by the fact that social networks significantly extend the speed and scope of rumors. The mixed group was asked to think of ways of limiting the damage from such rumors.

The mixed group divided the problem into three aspects. The first is detection. A system that has the ability to detect that information about a user is spreading can more effectively respond in case the information is (unfounded) rumor. The second is limiting the damage. There are several possible approaches: rate-limiting the spread of information, providing anti-rumor mechanisms like requiring quora and moderation by trusted third parties or representatives of peers, and educating honest-but-clueless users about the impact of spreading rumors. The last point relates to the issue of visualizing the effects of an action as in Question 2. The third aspect is post-damage control. The ability to trace the provenance of data by having an audit trail will help the victim identify how a rumor was spread and respond effectively. Having the possibility of rebuttal by the victim will also help.

4.2.5 Conclusion

Overall, the discussions identified two novel research issues. The first, which arose in Question 1 (and is relevant to Questions 2 and 4), is “How to reliably measure the gap between a user’s mental model and system functionality so that the guidance provided by the system can adapt accordingly?”. This is an open and difficult research question.

The second is the entire area of Question 4: “How can users mitigate the damage arising from rumors in social networks about themselves?”, which opens up many interesting questions as to how to design anti-rumor and rebuttal mechanisms and the pros and cons of throttling the spread of information.

5 Challenge: Economics

Topics expected from proposal: Sustainable operation, privacy preserving analytics, quality of additional services (recommenders), successful roll-out and incentives

Additional topics raised in position papers: Value of commercialization (targeted ads), value of detailed PII, value of privacy

5.1 Background

Current centralized online social networks are highly valuable businesses. Similar to many web services, they have turned around the model of how charging is done. It has historically evolved from static, embedded advertisement, through search engine's effort to increase click-through rates by profiling and personalization, to the accepted paradigm shift from "The user as a customer", to "The user is the product". These business models are termed "two-sided" markets, since the intermediary provides a service to customers on one side, usually for free in terms of money (but does impact their resources in other way, including eyeballcongnition tie, and network and screen and possibly battery life on mobile devices), and charges advertisers on the other side, money for delivering targeted commercial information to the customer side.

Centralized approaches are based on quite simple economic foundations: The revenue stream flows from advertiser to content/service providers, and the reasons for its success are the two possibilities the digital media offers over traditional mass media.

1. The service provider may know when someone acts on an advertisement (click through) and is capable to collect and analyze quite detailed information about this individual.
2. The service provider knows the demographics of its user base, and hence has the opportunity to perform detailed market research.

Such an environment creates a power imbalance (much like early online banking did in security assurance). The customers get a service for free in return for risking their privacy. Because the service provider wants to do a land grab on all services (healthcare, online shopping, travel, as well as education), they are incentivized to gather more and more details about their users.

This paradigm has evolved from conventional loyalty card services. Individuals in the past had a loyalty card for each different service. Health records were separate from banking. Work records and income were separate from insurance information. Unifying and centralizing all this information introduces serious risks. They introduce an immense attack surface in terms of technical, human and economic weakness, as well as in terms of smallness of the gene pool.

Depriving the systems of their economic bases, by removing the possibility to profile and target advertisement, may introduce economic pressure on decentralized approaches. Economic pressure, while certainly an issue, may not be a fatal obstacle to the provision of the service, as the success of free and open source software has shown in the past. The lack of funding as an incentive, however, may pose to be critical for the allocation of resources, cooperation, and even the provision of content.

Furthermore, decentralizing and encrypting PII may help privacy and reintroduce consumers' rights. However, they may come at the cost of losing oversight of the complete data, thus removing the control to index and locate resources, and to fight unwanted content as well.

Some of these problems may be resolvable by providing privacy preserving queries or trying to apply differential privacy. While numerous efforts have been undertaken, no one has been able to build such a system yet.

Some known methods may help to overcome these challenges, like game theory and mechanism design, experience from networking sciences and distributed systems as well. These issues require discussion, and approaches as well as road maps to aid solving them have to be identified.

5.2 Discussion

The expert group on Economical Aspects identified two complex fields of questions, which then were addressed in parallel by two mixed groups.

5.2.1 Question 1: Are personal data markets viable?

The first set of questions explores personal data markets. The attempt was to try and estimate the value service providers can extract from personal data in the current (ad-sponsored) model with the intention to bound the amount of compensation needed if privacy-friendly systems deprive service providers of this source of income. Likewise, it was attempted to estimate the willingness of users to pay to protect their personal data. Comparing these estimates should yield, *ceteris paribus*, if a market clearing could be expected or not. Although some price information may be observable in practice, externalities and context-dependence make it hard to interpret these indicators as reliable proxies for the value of personal data.

Pondering the viability of data markets, both groups discussed the actual value of personal data, when exploited for behavioral advertisement. Targeted ads currently make up for a small share of the overall advertisement market only. Even the large players like Google and Facebook can earn only two to three US dollars for behavioral advertisement per user and year. This, however, would potentially already match the cost of a decentralized social networking system. Then again, it certainly is too low to create and maintain a data market between users and advertisers. It is much lower than the revenues cell phone providers and manufacturers can realize. Taking a closer look, the groups detailed that (a) demographic information literally has only negligible value (given the prices charged by commercial services like Spokeo, Rapleaf, and Equifax for such kind of information), (b) information about creditworthiness, income, health, or consumption patterns may be slightly more valuable, and finally (c) the information value decreases over time.

Trying to assess the cost users could be willing to pay for the protection of their data yields a different picture. Protecting their personal information today may be compared to the cost of an insurance that will pay off in future after potential events of data loss. The discussion then led to the conjecture that events of data loss could become so frequent and ordinary that they may not even yield any consequences, nor reputation loss for the culprits any more, as can already be observed at recent examples (companies losing large sets of login/password pairs).

Estimating how much users may be willing to pay for the protection of their data, the groups found instances where users are willing to pay on the order of hundreds of \$US per anno for services that promise to protect the online image of a user (ReputationDefender, DeleteMe). This can be supported by the fact that individuals are willing to pay for curtains or to opt out of listings in phone books. However, there is currently not enough data to support claims that enough users would be willing to pay substantial amounts of money for multiple large businesses to form in this area.

A noteworthy observation in the discussion was the fact that users are actually willingly give their data away, if they are convinced it was for a good cause. Individuals participate in focus groups, surveys, loyalty programs (Groupon, Payback, Frequent Traveler programmes) and even disclose their entire browsing history to researchers, if asked. The main concern hence is not the fact that the users lose data, but only for whose benefit and to whom they voluntarily give it.

5.2.2 Question 2: Does a fully decentralized, privacy friendly SNS break even?

Acknowledging this difficulty and recognizing that exploitation of personal data by service providers is a necessary but inefficient way to refinance a centralized infrastructure (and make profits), we came up with a second set of questions that asks if and under which conditions a decentralized infrastructure is viable without advertising. We start with a set of simplifying assumptions, which we relax, one by one, to approach reality. First we look at the steady state (i.e., no transition from or competition with centralized systems) and assume user homogeneity. Most likely, the benefit of the network exceeds its operating cost. The cost may even be small enough to go under the radar of rounding errors in over-investment (this is considered to be controversial). When users are heterogeneous, the cost of some users may exceed their benefit, suggesting that they may exit the network and thereby impose negative externalities on all others by the loss of positive network effects. Such frictions arising from heterogeneity may destabilize the system and so it is crucial to solve the mechanism design problem to align incentives and internalize these externalities. The last step towards reality is to drop the steady-state assumption because not every system that is viable in steady-state might get there if path dependencies lock society into centralized systems. A key problem in establishing a new system is to reach a critical mass, and the most plausible strategy is to leverage existing decentralized systems to share fixed costs and enjoy the network effects of the existing user base. This leads to the question of identifying the right existing infrastructure that could be leveraged for this purpose.

Judging the potential to break even requires understanding of the involved costs and benefits. Both mixed groups identified the core costs to be monetary costs (payments for participation, donations, fees), inconvenience cost (potentially restricted or inferior functionality, viability), development cost, and infrastructure cost (storage, bandwidth, computing). Benefits include increased privacy, psychological effects (satisfaction to participate in or support such a system), infrastructure utilization (availability of data to others, traffic consumed for service), utilization (using the OSN, potential ease and convenience of use), utilization of additional services (recommendations, reputation).

Some of the benefits clearly gain from network effects: beyond a minimum size of the system, each additional user superlinearly increases the benefits (happiness to have built or to have supported the system, utility of using it). Analyzing the system, three different stakeholders with costs and benefits can be identified: (1) the users (who pay monetarily or in degraded service and functional quality and who gain privacy, potential functionality, and the access to resources), (2) the developers and maintainers (who invest time and possibly money to create the system), and (3) infrastructure providers (paying with money or resources, mainly gaining as philanthropists or by increasing the number of participants for their own benefit as users). A group made the noteworthy observation that even users who only use the SNS without providing money or any kind of infrastructure resources are actually providing benefit to others, by sharing content, opinions, and votes.

Both mixed groups dismissed the homogeneous case as unrealistic and directly addressed the heterogeneous case of a strong imbalance in utilization (shared and retrieved content vs. shared resources).

Addressing the potential to adopt a decentralized system, a gradual transition, piggybacking on existing systems, seem to be the only possibility. The majority of users is assessed to be unlikely to pay, set up and maintain partial infrastructure, and to migrate to an unpopulated replacement of any existing system. Strong support may either come from subversion (the satisfaction of participating in something subversive) or regulation (legal acts

upon serious incidents when a better solution is readily available).

Several hosts for piggybacking have been identified, and there seem to be three viable ways: (1) leveraging existing offline communities with an interest in privacy (schools, universities, unions) to support the deployment, (2) bridging with existing services (extract data and social graphs from Facebook), and (3) federation by integrating several services over a common interface as an abstraction layer (implicitly choose the most privacy preserving storage substrate through the selected audience).

5.2.3 Conclusion

The experts concluded that data markets could be viable if there were efficient ways for monetization other than advertising. One example would be to switch from users paying with eyeball time to a direct subscription system. Estimates of the cost of a payment system were mad, and it was a relatively modest fraction of current fixed and mobile broadband data access subscription prices. Although seen as certainly viable a discussion on the economic and organizational barriers to bundling cloud and network services concluded that such an approach was problematic in business and economic terms

A key discussion then turned on the need for efficient mechanisms for micro-payment, such as BitCoin. Some such mechanisms are decentralized themselves reducing the risks of merely moving ownership of data from the storage and processing provider to the digital cash mint.

However, considering that data seems not to have much worth placed on it by the users, the viability of such markets remains questionable. It is clear that more research is needed to understand the difference in perception of value of users' content amongst the users, and between users and providers. Studies on risk perception show that it is wildly fluctuating depending on recent positive or negative experiences.

Summarizing the discussions on the chance to break even and retain sustainability, the experts concluded that the choice of the initial set of users is crucial to accelerate acceptance and to achieve information mobility to ease migration. Since the majority of users is expected to be unwilling to pay, a tightening approach is suggested, in which the service could be created at a surplus to achieve a critical mass, and only slowly be burdened by requiring contributions.

It was noted that online services for music and video have moved through several evolutionary stages from piracy, to pay-by-advertisements, to subscription-based on demand systems, which are now highly successful. Indeed, the middle phase did not appear to impact piracy, but the existence of efficient media subscription services appears to cause rapid reduction in content piracy. This suggests that such an economic approach to OSNs, for example, might be equally successful in reducing abuse: abuse in the sense of loss of privacy of one user by another they don't know perhaps due to incomprehensible privacy settings; and the abuse in the sense of loss of privacy due to OSN operator's monetizing of users' PII.

Privacy preserving analytics were partially covered by a presentation by one current project startup, which, fully centralized, performs analytics on unencrypted data. Privacy is achieved by processing the data under access protection that is guaranteed by a trusted environment (anchored in a TPM), protecting the results to some extent by an approach similar to differential privacy.

There are several places that one can put the users' content in a decentralized architecture. These are not all necessarily victims of the problems discussed in the well-known results that were bought up by several participants, which pointed out the availability and latency

problems of a full p2p approach to decentralization¹. Various techniques were discussed where payment for storage for cached copies, or payment for encryption for homomorphic crypto-based interest matching for advertisement delivery and for differential privacy of semi-centralized data stores, to offset these availability problems without resorting to centralizing everything again. Such caches are controlled by users' edge devices' (home machine, home router/hub or smart phone as master copy), and could employ multiple cloud service providers, switching dynamically as desired and costed. A full spectrum of solutions in between is feasible, including some non crypted or non-decentralized data, with the corresponding range of business models.

Again, the current situation is not necessarily a good predictor of the future - performance and availability of home user systems in a future with fibre to the home may be very much better than centralized systems one day.

The cognitive overload of paying should also be properly quantitatively compared with the perceived stress of receiving advertisements.

6 Challenge: Technology

Topics expected from proposal: *Which functionality is necessary, at which quality to meet the user requirements? What crypto primitives are available/needed? Feasibility, Scalability, Mobility and Location Privacy, Opportunities.*

Additional topics raised in position papers: *Device limitations/restrictions (lost control over mobile devices, mobile app permission systems), problems of decentralization (observability, traffic analysis, DoS), novel challenges/threats from decentralization (integrity, end-point correlation, availability), metrics to measure privacy, primitives for privacy preserving recommendation, to establish trustworthiness, decentralized trust (evidence-based trust, reputation with privacy)*

6.1 Background

Privacy threats have risen along with the increase of large-scale data collections not only of user-provided content but also of automatically collected personal, relational, and behavioral data, as exemplified by online social networks, credit and loyalty cards, or tracking on the Web. These data collections are concentrated at (logically) centralized service providers, thus intentional and accidental leaks of private data to third parties can have significant impact.

There has been a lot of research activity on privacy over the last decade, especially in the Computer Science community. A main outcome there has been the development of several privacy-enhancing technologies (PETs) such as onion routing, mix-nets, or anonymous credentials. More recently, there have been approaches that propose to break up provider-dependent centralized data collections and return the control over their data to the users themselves by decentralizing systems and replacing the gatekeeper functionality of a centralized provider by technical means such as cryptographically enforced access control and metadata-minimizing system design.

¹ See for example "High availability, scalable storage, dynamic peer networks: pick two" by Blake, Charles and Rodrigues, Rodrigo, in HotOS 2003.

The main focus of these efforts has been on decentralizing social networks. Regular, centralized online social networks are a particularly good example for this problem as they have an extremely large user base and collect information in addition to what the users upload about themselves. They have data on what users say about other users, whom they interact with, and other behavior also on third-party sites thanks to tracking and functionality such as liking content on the web.

Different models for decentralization have been proposed. One common approach is to store user content in a distributed system, such as a peer-to-peer network with replicas or at least a collection of independent servers instead of a single-provider and control. There are design challenges at several levels in terms of security, availability, scalability, robustness, new privacy issues, usability, etc. for any decentralized service or application, underlying storage and network topology, as well as trust relationships. One particular challenge is to compensate for the privacy-preserving and security functionality that does exist in centralized single-provider services that can, for example, hide behavior data from other users.

It is not clear to what extent decentralization can be both feasible and beneficial for keeping the system functionality and preserve privacy. For instance, there are trade-offs in terms of provider independence and resource allocation and trust management for fully decentralized systems versus centralized systems with cryptographically protected user data versus federated solutions.

The potential and the limits of decentralization as a means to enhance privacy have to be explored. After an overview of the state of the art is established, we will be able to detect gaps and thus determine what a research roadmap would be to bridge these gaps. While there will continue to be a need for diversity in research approaches, areas of synergy have to be identified.

6.2 Discussion

The expert group for the Technological Aspects identified six questions, divided into two blocks for separate mixed sessions.

6.2.1 Question 1: What are the important design goals of decentralized privacy-preserving systems?

The goals can be separated into several categories. The first category encompasses privacy requirements. At a high level, the concept of information self-determination, i.e., that users should be in control of their data, seems like a good starting point; however, there are many details that affect the design of the system. For example, it is not clear what level of resistance to traffic analysis is necessary or desirable.

The second category is the utility goals: what are users expected to accomplish through using the system? Existing OSNs provide a variety of different communication functionalities, from sharing simple status updates, to finding friends, to monitoring for malicious content.

The final category concerns the goals of the users of the system. Here, an understanding of what kind of individual goals each user might have is required; for example, some users might prefer to know when their data is shared, some users might prefer not to be tracked, and so on. Important questions include: how do users express these goals to the system? How can a system realize the goals? And how do those conflicts between the goals of users be resolved? The latter problem touches on the complexity of data ownership in OSNs.

Different systems will have different goals; a crucial task is to create some form of taxonomy of goals to understand their individual properties, as well as the relationship between them. In terms of individual properties, it's important to understand to what extent is an individual goal realizable, deployable, or commercializable. Other concerns might be whether such a goal would lead to novel research and published papers, and more fundamentally, how well would this type of goal align with the type of aspired society. In terms of relationship, it is necessary to identify which goals are mutually incompatible.

Providing a complete taxonomy within a single session was deemed to be impossible. However, different dimensions and approaches could be identified.

The dimensions along which the goals can be measured are

- whether the goal is for the individual or for a collective
- whether the goal supports freedom of action or information flow control
- whether the goal seeks to embed social norms into technology, or rather, change social norms through technology

These dimensions are not necessarily independent, and they may not always be applicable, however, they represent good guidelines.

Two different approaches to identify goals are top-down and bottom-up: Goals can be defined either with powerful adversaries, or idealistic protection objectives in mind. Another, sometimes potentially more useful and pragmatic way is to identify users and stakeholders as well as their actions and needs, to collect specific requirements and derive goals from them. The second approach partially is motivated from the insight that current practice seems to be to design systems and protocols with certain properties and derive the goals later, instead of developing systems towards actual needs. A simple exemplary approach is to define personae and the actions they take and to analyze how they'd be affected by design decisions.

6.2.2 Question 2: What are the threat models that need to be considered?

There are various types of threats to the security and privacy of a decentralized system; given a set of goals and a potential set of threats, attempts can be undertaken to realize a system that satisfies those goals in the presence of an adversary.

To understand the space of threats, some sort of taxonomy is required. In particular, it is necessary to understand what kind of threats we have to worry about – some might be realistic today, some might become significant in the future, and others might be artificial. In addition to well understood threats, there may be research problems to identify in exploring new types of threats that are specific to OSNs.

Defining privacy threats in the scope of social applications is not straightforward, since it is hard to differentiate between the intent of the user to share some information, as opposed to the dangers of others actually retrieving it. Arriving at a taxonomy of threats necessitates the definition of a hierarchy of categories to avoid simply filling a very large matrix of categories. Both ways, though, do not seem feasible within a single discussion session, and only first steps towards this goal are taken.

Three dimensions have been identified for a taxonomy:

1. The system architecture,
2. Stakeholders
3. Assets.

Considering the categories, the type of system has to be defined and basic differences identified (e.g., centralized vs. decentralized). Further categories are the stakeholders, specifically the actors and adversaries, as well as assets that need to be protected (from directly identifying over pseudonymous to location information).

The stakeholders have to be discriminated and defined, and it is necessary to define who can actually be trusted, and how they are incentivized. The choice of adversaries to address has several sides. The honest but curious adversary may be the most realistic and hence should be taken into account at first. Considering the most extreme cases (like nation states, malicious providers, or even organized crime) yields understanding and lessons, and it may make sense to aim for the strongest but usable protection. Cases with weaker adversaries are valuable, too, though, since they may reflect reality better.

To judge the threats and order them by importance for the sake of prioritizing them, their potential harm, their likelihood of being realized, and finally the effort or ease to fix them have to be considered. Privacy Impact Assessment is a methodology that could potentially be applied for this purpose.

6.2.3 Question 3: How do we ensure that a system has a good chance of seeing the light of day?

A clear prerequisite to the adoption of a system is the filling of an unfilled need. It is not necessarily clear that privacy alone is compelling enough as a need to accumulate a large user base, especially when trade-offs are introduced. One potential strategy may be to appeal to a niche community, at least initially, rather than try to be everything for everyone.

Even so, adoption can be slow and has a bootstrapping issue. The important question here is whether we can leverage some existing infrastructure to ease this bootstrapping burden.

Finally, there are many platforms that people use for accessing OSNs, including desktops, mobile phones, thin clients (web browsers). What are the constraints that these platforms create for the design of the system?

Privacy is not in itself an unmet need by common understanding. It may attract a few, but other reasons to change are needed for the majority of users to actually change their service. Factors of scale, like usability, which can only be guaranteed with a large number of highly qualified developers, as well as network effects actually are strong antagonists of change. Achieving the same functionality, usability and reaching the network effects, privacy as a matter of fact can be considered a compelling property. A successful deployment, however, is much more likely with additional, complementary or innovative functionality.

Piggybacking on other systems to kickstart the acceptance is conceivable in four different ways. A new system can be bootstrapped out of an existing or several existing systems. One possible way is to either harvest the content out of existing systems and replicate it in the novel platform to provide a set of interesting content and make the transition easy for the users. This, however, is impossible for institutional approaches, since the licenses of existing systems prohibit it and the providers are very keen on protecting the content they collected from the users from competition. Another approach could be federation: allowing for seamless integration, system could run behind a single interface in parallel, and for each act of sharing, the most secure medium that reaches all destinations could be chosen automatically. Another

possibility is to bootstrap the system from existing infrastructures, like eduroam² (with several hundreds of thousands of users), shibboleth³, or even SIM card infrastructures of mobile network providers. The third possibility is to leverage existing organizations with an inherent interest in communicating at a minimum level of confidentiality. Data about minors or schools is a good example. But organizations with interest in confidentiality and large numbers of members, such as Unions, are likely to act as amplifiers, as well. The fourth opportunity is to leverage fields of applications that already have large user bases, like CSCW, for instance. Rationale behind such attempts needs to be to reach critical number of users as core by such a subversive ways, to spawn and achieve network effects, to make transition more attractive to the large majority of users.

Regulation is an entirely different factor. It seems useful to develop a secure and viable solution as an alternative for the case when more critical incidents of data loss happen, in order to be able to offer it to regulators as an example, if even only to define the properties that can be required by regulators consequently.

The final sub-question raises the issue of future devices and their restrictions: Mobile devices with locked operating systems are quickly gaining market share, and users on PCs and notebooks are decreasingly willing to actually install applications and background daemons. Future deployment of TPM could exacerbate this. This results in the demand to develop purely browser-based systems, and to address loss of control over devices. It doesn't seem sensible to protect user data on the application layer of (D)OSN from players like Google, or Apple, if the protection is implemented on a device that runs Android/iOS and hence is under control of these entities on a lower layer, anyway.

6.2.4 Question 4: What are the building blocks for decentralized privacy systems?

A sub-question is whether decentralization itself is an essential building block for privacy, given the research into privacy-preserving protocols on top of, for example, cloud computing. For building blocks that we identify, we need to understand the costs and benefits of these tools, in terms of privacy guarantees, performance, the ability to generate revenue, the underlying feature set, and so on. An important question is how such blocks may be securely composed, since in general, the security of composition of functionalities has been difficult to achieve.

A related question is how can we encourage our immediate community to build reusable building blocks so that we can build on each other's work, rather than starting from ground zero each time. Are there barriers to such sharing that we can identify and perhaps address?

The question was slightly adapted to address the potential architectures of privacy preserving social applications, their respective costs and benefits, and how they could be separated into building blocks that compose well.

Decentralization in this context is defined as distribution of storage and control over several authorities, or providers. Scopes of decentralization range from centralized over hybrid to entirely decentralized systems (Facebook, Diaspora, Peerson or Safebook). Federation may be used in a way to avoid putting all eggs into the same basket, and in this case can represent decentralization, too.

To achieve a better understanding, it is necessary to understand reasons to decentralize or

² <https://www.eduroam.org/>

³ <http://shibboleth.net/>

the effect of decentralization first. Stronger control may be a motivation for decentralization, as well as for increased performance or reliability. Taking a closer look it becomes apparent that these points are actually not achieved in currently developed and proposed solutions, and they may actually be quite hard to achieve. Further reasons are to avoid being sued (the difference between Napster and BitTorrent and their greatly different history being the lack of an institution in control of the service), and avoiding a single point of failure with respect to collaboration for censorship, or “lawful interception” as defined by regimes and governments of vastly different natures. A final property of decentralization is that they be future proof: while centralized services can run into financial problems or be sold and hence quite drastically change their licenses, this threat does not affect decentralized systems.

The advantages come with drawbacks, and being both censorship proof and confidential service, the threat of abuse is high, and may actually have disincentivizing effects for large fractions of users.

Defining building blocks is actually more difficult than expected. It is neither clear if the developed systems should be general purpose OSN or if its better to create targeted solutions in order to understand features, concepts, and resulting properties. Nor is it straightforward to decide to which extent the building blocks need to be adaptable, keeping in mind the understandability and usability of the respective API.

Some building blocks have been identified nonetheless. The following list may not be comprehensive but serves as a good starting point. The group enumerated them to be: storage (not necessarily distributed and highly configurable), registration (account creation, identification, key-escrow), profile management (content publication, audience selection spanning authorization and access control, potentially by crypto and key-distribution), secure user discovery, a crypto library (integration of useful primitives, usable and comprehensive), notification mechanisms, connection establishment and NAT traversal, technical bootstrapping. Further useful functionality was identified to contain a voting mechanism, partial message ordering, and mechanisms for anonymous communication, even though it wasn't quite clear how general those were.

A final discussion revolved around the question of how reuse could be encouraged and made possible. The main obstacles to this have been identified to be of psychological nature, but good and well documented code, exact and understandable APIs, possibly even the provision of simulation or numerical models could help, to aid the people integrating building blocks understand and assess the properties of the composed system.

6.2.5 Question 5: How do we evaluate system designs?

Systems can be evaluated from a variety of different metrics and using a variety of approaches, from formal analysis to experiments to field deployments. It may not be productive to identify the space of all possible evaluation metrics and strategies, since many of them are not specific to decentralized privacy systems. Instead, we should focus on identifying what types of evaluation is especially important for decentralized privacy-preserving systems, and in particular, what types of privacy metrics might be relevant. Are there existing ones or do we need to define new ones? It would also be fruitful to identify any challenges and barriers to evaluation that privacy-preserving systems create, such as the difficulty of obtaining realistic data sets while preserving individual privacy.

The initial, motivating observation of this discussion was that no consensus nor accepted approach exists, and many previous and current studies lack rigor, realistic assumptions, and reproducibility. This is exacerbated by the fact that comparing two systems by itself

is complicated, and only convincingly possible if one system is a superset of the other, or one dominates the other entirely. There is no agreement on a set of metrics that measures privacy in a convincing manner, let alone in cross-domain scenarios, and thus evaluation is difficult and the results questionable. Even just cultural differences have been pointed out to be a cause for complication, since they define what is acceptable in different cultures, and hence may have an effect on how the protection of different systems is perceived.

Another major obstacle for evaluations is the lack of good data sets that define assumptions and environment. In an attempt to sketch a solution a possibility could be to send academic interns to companies in possession of the data. The interns then could analyze the data, specify exact datasets and their descriptions, implement and perform analyses and evaluations on site and publish only the results. Other could then reproduce the results knowing the specifications by sending their evaluation code to the companies, who would run it and again provide only the results.

6.2.6 Question 6: What is the state of the field?

We should identify which questions might have already been answered well in the research literature (or in practice), and which questions are deserving of future research. It would be useful, though perhaps controversial, to perform a retrospective analysis of existing research and identify directions that we feel are particularly compelling, as well as directions that seem less scientifically interesting or that are based on hard-to-justify assumptions.

Acknowledging that the entirety of the state of the art was too broad to be fully covered in a single session, the team discussed the general situation of the research fields that have to come together to provide privacy preserving services and their integration. It came to the conclusion that while an impressive body of work on crypto primitives, algorithms, and protocols exist, the lack of understanding for the actual requirements and functions led to only partial applicability. This problem has additionally been identified to affect the systems solutions as well, which have seen a large variety of proposals for technological solutions, yet no notable adoption by the users, so far. This conclusion was encountered with the observation that no solution has ever been successful at first shot, and that it's usually followers, who succeed, not the innovators.

6.2.7 Conclusion

The six questions of the technological challenge turned out to be too broad for conclusive discussions.

Taxonomies for goals, threats, and concepts including their properties are needed and should be provided by and to the community. Initial dimensions for classifications have been sketched and discussed, which seem to be (at least) a good starting point. *Goals* could be classified by the cardinality of their subjects (individual vs. collective), by their data control (freedom of action vs. information flow control), and their social impact (implementing or changing social norms). Potential classification dimensions for *threats* are the system architecture (centralized vs. decentralized, type of access control scheme), capabilities and goals of stakeholders (who are actors, who/how powerful are adversaries), and which assets are to be protected (explicitly/implicitly shared data; directly identifying, pseudonymous, location information). Classifying the functional concepts, finally, will greatly depend on assumptions and requirements, and the taxonomies hence will naturally address a combination of the dimensions above.

It has to be noted that decentralization by itself does not directly lead to increased privacy. Even implementing systems that enforce access control over explicitly shared data, decentralized systems can be highly vulnerable to traffic analysis attacks, partially due to the lack the mixing property of centralized systems.

It is not quite clear if a general purpose social networking service should be the aim of decentralization efforts, or if it is more promising to start with specific, more targeted services. An initial, preliminary list of general building blocks for both cases seems to include functionalities for: registration, profile management, secure user discovery, notification, general purpose storage, comprehensive crypto, bootstrapping, and connection establishment and NAT traversal.

7 General Observations, Questions Raised, Open Problems

Privacy by itself is not an unmet need, and providing privacy in an otherwise identical system will not yield widespread adoption. Privacy can, however, be a compelling property; and offering privacy protection in a system with nearly-identical usability and an attractive set of features most probably can be a strong success factor.

Successful propagation of a novel, privacy preserving, either partially or completely decentralized service will thus depend on additional factors. Reaching a critical mass requires the participation of a significant number of initial users, before network effects and word of mouth will help increase general interest. Leveraging existing infrastructures and social structures could help attract this initial user base. Existing infrastructures may be current services, integrated by federating user interfaces, or existing authentication services of large institutions and their conglomerates. Existing social structures may be institutions with the need to communicate at high levels of privacy protection, like schools, or unions.

General observations regarding the assumptions were that systems need to address the heterogeneity of users with respect to their activity and sharing behavior, as well as their need to persist beyond initial startup phases, in which the resource provision potentially can be hidden in the margins of other services, to a sustainable existence. The latter may only be possible if the systems either allow for monetization (beyond the almost negligible advertisement market), or implement an internal incentive scheme to attract sufficient resource providers.

Recurring themes from the technological perspective were the different promises and drawbacks of trusted platform modules (TPM) and both usage and control restrictions of future devices. TPM promise guarantees of trustworthiness: users may be able to restrict and verify the absence of chances for downstream abuse, which may help deciding which information to share with whom. Analyzing changing habits of the users it can be assumed that soon the vast majority will access services over mobile devices or browsers. Both cases prevent installing client software that would constantly run in the background (and in case of laptops and PCs this starts to be frowned upon) to provide services to others. The mobile devices raise the even more important question of how the users can regain control without having to jailbreak or root them, to avoid running a seemingly trusted system on a platform that is controlled by one of the potential adversaries.

Several novel research questions, which are addressed partially at the computer science communities and partially beyond, have been phrased. Questions for our own community covered HCI, economics, and technological fields.

- Oversharing frequently happens due to mistakes and misunderstood controls. *How can*

*we measure the gap between the mental model of the users and the reality? How can we adapt and encounter it, to achieve the **Principle of least astonishment**?*

- More specifically: *How can the de-facto audience of a post be visualized, how can audience selection be simplified?*
- Upon incidents: *How can users mitigate the damage arising from rumors or libeling content?*
- The community is developing large numbers of systems, algorithms and protocols: *How can re-use be fostered; How can sensible and useful crypto-libraries be provided; Which scope should systems (and hence the shared building blocks) target?*
- Considering the introduction of novel, more secure services: *How can existing social structures and infrastructures be leveraged to bootstrap a novel system?*
- The discussions yielded the insight that the behavioral advertisement market is rather negligible, but within the light of this fact: *Are monetary or other incentive markets viable, and how is the sustained operation of a novel service possible?*

Questions beyond the field of expertise of the participants address mainly legal matters:

- *Considering a fully decentralized system, who is responsible (read: can be sued) for its operation, content, and offences committed using it?*
- *In the global context, who owns the data shared on social networks, who has copyright, right of use, right of deletion?*
- *Is plausible deniability realistic protection for institutions running TOR- or Freenet nodes?*
- *To which extent is integration with existing services (federation, use as storage substrate, retrieval of data stored within) legally acceptable?*
- *Could eduroam/shibboleth be extended to global identification services; at which complexity?*

Participants

- Jonathan Anderson
University of Cambridge, GB
- N. Asokan
University of Helsinki, FI
- Rainer Böhme
Universität Münster, DE
- Nikita Borisov
Univ. of Illinois – Urbana, US
- Sonja Buchegger
KTH – Stockholm, SE
- Ramon Caceres
AT&T Labs Research –
Florham Park, US
- Jan Camenisch
IBM Research – Zürich, CH
- Jon Crowcroft
University of Cambridge, GB
- George Danezis
Microsoft Research UK –
Cambridge, GB
- Claudia Diaz
K.U. Leuven, BE
- Vijay Erramilli
Telefonica Res., Barcelona, ES
- Simone Fischer-Hübner
Karlstad University, SE
- Paul Francis
MPI for Software Systems –
Kaiserslautern, DE
- Ian Goldberg
University of Waterloo, CA
- Artur Hecker
Huawei Technologies, DE
- Urs Hengartner
University of Waterloo, CA
- Jaeyeon Jung
Microsoft Res. – Redmond, US
- Mohamed Ali Kaafar
INRIA Rhône-Alpes, FR
- Gunnar Kreitz
KTH Stockholm, SE
- Balachander Krishnamurthy
AT&T Labs Research –
Florham Park, US
- Leonardo A. Martucci
Karlstad University, SE
- Bart Preneel
KU Leuven, BE
- Stefanie Roos
TU Darmstadt, DE
- Krzysztof Rzadca
University of Warsaw, PL
- Hervais-Clemence Simo Fhom
Fraunhofer SIT – Darmstadt, DE
- Thorsten Strufe
TU Darmstadt, DE
- Paul Syverson
NRL – Washington, US
- Claire Vishik
Intel – London, GB
- Marcel Waldvogel
Universität Konstanz, DE



Dependence Logic: Theory and Applications

Edited by

Samson Abramsky¹, Juha Kontinen², Jouko Väänänen³, and Heribert Vollmer⁴

1 University of Oxford, GB, samson.abramsky@comlab.ox.ac.uk

2 University of Helsinki, FI, juha.kontinen@helsinki.fi

3 University of Helsinki, FI, jouko.vaananen@helsinki.fi

4 Leibniz Universität Hannover, DE, vollmer@thi.uni-hannover.de

Abstract

This report documents the programme and outcomes of Dagstuhl Seminar 13071 "Dependence Logic: Theory and Applications". The seminar brought together researchers from different areas such as mathematical logic, quantum mechanics, statistics, social choice theory, and theoretical computer science. A key objective of the seminar was to bring together, for the first time, researchers working in dependence logic and in the application areas so that they can communicate state-of-the-art advances and embark on a systematic interaction.

Seminar 10.–15. February, 2013 – www.dagstuhl.de/13071

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Data structures, Algorithms, Complexity, Verification, Logic

Digital Object Identifier 10.4230/DagRep.3.2.45

Edited in cooperation with Miika Hannula

1 Executive Summary

Samson Abramsky

Juha Kontinen

Jouko Väänänen

Heribert Vollmer

License  Creative Commons BY 3.0 Unported license
© Samson Abramsky, Juha Kontinen, Jouko Väänänen, and Heribert Vollmer

Brief Introduction to the Topic

Dependence Logic is a new tool for modeling dependencies and interaction in dynamical scenarios. Reflecting this, it has higher expressive power and complexity than classical logics used for these purposes previously. Algorithmically, first-order dependence logic corresponds exactly to the complexity class NP and to the so-called existential fragment of second-order logic.

Since the introduction of dependence logic in 2007, the framework has been generalized, e. g., to the contexts of modal, intuitionistic and probabilistic logic. Moreover, interesting connections have been found to complexity theory and database theory, and dependence logic has been applied in areas such as linguistics, social choice theory, and physics. Although significant progress has been made in understanding the computational side of these formalisms, still many central questions remain unsolved so far.



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Dependence Logic: Theory and Applications, *Dagstuhl Reports*, Vol. 3, Issue 2, pp. 45–54

Editors: Samson Abramsky, Juha Kontinen, Jouko Vaananen, and Heribert Vollmer



Dagstuhl Reports

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

The notions of logical dependence and independence are pervasive, and occur in many areas of science. The development of logical and semantical structures for these notions provides an opportunity for a systematic approach, which can expose surprising connections between different areas (e. g., quantum mechanics, social choice theory, and many more), and may lead to useful general results.

One of the main aims of this Dagstuhl Seminar was to bring together, for the first time, researchers working in this area so that they can communicate state-of-the-art advances and embark on a systematic interaction. In particular, bringing together researchers from areas of theoretical studies with the application areas will enhance the synergy between the different communities working on dependence logic.

Organization of the Seminar and Activities

The workshop brought together 35 researchers from mathematics, theoretical physics, statistics, social choice theory, and theoretical computer science. The participants consisted of both senior and junior researchers, including a number of postdocs and a few advanced graduate students.

Participants were invited to present their work and to communicate state-of-the-art advances. Seventeen talks of various lengths took place over the five days of the workshop. Introductory and tutorial talks of 90-60 minutes were scheduled prior to workshop. Most of the remaining slots were filled, mostly with shorter talks, as the workshop commenced. The organizers considered it important to leave ample free time for discussion.

The tutorial talks were scheduled during the beginning of the week in order to establish a common background for the different communities that came together for the workshop. The presenters and topics were:

- Jouko Väänänen, Dependence Logic
- Erich Grädel, Logics with team semantics and second-order reachability games
- Philip Dawid, Conditional Independence and Irrelevance
- Pietro Galliani, Definability Issues in Team Semantics
- Phokion Kolaitis, Foundations and Applications of Schema Mappings
- Samson Abramsky, From Quantum Mechanics to Logic, Databases, Constraints, Complexity and Beyond
- Sebastian Link, Dependence, Independence, Logic
- Wilfrid Hodges, Compositionality: Its history and formalism
- Eric Pacuit, Dependence and Independence in Social Choice Theory

There were additionally 8 other talks with a more focused and technical topic.

1. Georg Gottlob, From Local Hidden Variables in Quantum Mechanics to Robust Colorability and Satisfiability
2. Panayiota Constantinou, Extended Conditional Independence
3. Fan Yang, Uniform definability in propositional dependence logic
4. Pierfrancesco La Mura, A double-slit experiment for non-classical interference effects in decision-making
5. Julian Bradfield, Concurrency, causality and dependency
6. Miika Hannula, Axiomatizing first-order consequences in independence logic
7. Andreas R. Blass, Introduction to Secret Sharing
8. Arnaud Durand, Complexity issues in dependence logic

The workshop achieved its aim of bringing together researchers from various related communities to share state-of-the-art research. The organizers left ample time outside of this schedule of talks and many fruitful discussions between participants took place throughout the afternoons and evenings.

Concluding Remarks and Future Plans

The organizers regard the workshop as a great success. Bringing together researchers from different areas fostered valuable interactions and led to fruitful discussions. Feedback from the participants was very positive as well. Many attendants expressed their wish for a continuation and stated that this seminar was among the most fruitful Dagstuhl seminars they attended.

Finally, the organizers wish to express their gratitude toward the Scientific Directorate of the Center for its support of this workshop, and hope to establish a series of workshops on *Dependence Logic: Theory and Applications* in the future.

2 Table of Contents

Executive Summary

Samson Abramsky, Juha Kontinen, Jouko Väänänen, and Heribert Vollmer 45

Overview of Talks

Concurrency and (in)dependence

Julian Bradfield 49

Extended Conditional Independence

Panayiota Constantinou 49

Conditional Independence and Irrelevance

A. Philip Dawid 49

Definability Issues in Team Semantics

Pietro Galliani 50

Robust Constraint Satisfaction and hidden variable detection in quantum mechanics

Georg Gottlob 50

Axiomatizing first-order consequences in independence logic

Miika Hannula 50

Compositionality: its history and formalism

Wilfrid Hodges 51

A double-slit experiment for non-classical dependencies in decision-making

Pierfrancesco La Mura 51

Dependence, Independence, Logic

Sebastian Link 51

Dependence and Independence in Social Choice

Eric Pacuit 52

Dependence logic

Jouko Väänänen 52

Uniform definability of connectives in propositional dependence logic

Fan Yang 52

Participants 54

3 Overview of Talks

3.1 Concurrency and (in)dependence

Julian Bradfield (University of Edinburgh, GB)

License © Creative Commons BY 3.0 Unported license
© Julian Bradfield

Joint work of Bradfield, Julian; Fröschle, Sibylle; Gutierrez, Julian; Kreutzer, Stephan

I review work by myself and colleagues over the last 20 years, which considers logics for concurrent systems, and how such logics relate to independence-friendly logic, and thus to dependence logic.

3.2 Extended Conditional Independence

Panayiota Constantinou (University of Cambridge, GB)

License © Creative Commons BY 3.0 Unported license
© Panayiota Constantinou

The notion of Conditional Independence can be extended to encompass stochastic and nonstochastic variables simultaneously. This extended language can express various notions in statistics, like sufficiency, causal concepts etc. Formalizing the extended language we study conditions that allow us to deduce the axioms of conditional independence (classical properties of stochastic conditional independence).

3.3 Conditional Independence and Irrelevance

A. Philip Dawid (University of Cambridge, GB)

License © Creative Commons BY 3.0 Unported license
© A. Philip Dawid

Main reference A.P. Dawid, "Separoids: A mathematical framework for conditional independence and irrelevance," *Ann. Math. Artificial Intelligence*, Vol. 32, Issue 1–4, pp. 335–372, 2001.

URL <http://dx.doi.org/10.1023/A:1016734104787>

Probabilistic independence and conditional independence play a major role in statistical theory. Probabilistic conditional independence can be shown to enjoy certain fundamental general properties, which can then be used as an independent axiomatic system. A further step towards abstraction produces a mathematical object, the "separoid", which can be interpreted as embodying the informal concept of "irrelevance", and has many applications beyond, or totally removed from, the initial probabilistic framework. In particular, the logical relation of "variation independence" defines a separoid. Furthermore, by building connexions with other mathematical models of separoids, in particular undirected and directed acyclic graphs, we can streamline the analysis of a given separoid structure.


References

- 1 Dawid, A. P. (1979). Conditional independence in statistical theory (with Discussion). *J. Roy. Statist. Soc. B* **41**, 1–31.
- 2 Dawid, A. P. (1979). Some misleading arguments involving conditional independence. *J. Roy. Statist. Soc. B* **41**, 249–252.

- 3 Dawid, A. P. (1980). Conditional independence for statistical operations. *Ann. Statist.* **8**, 598–617.
- 4 Lauritzen, S. L., Dawid, A. P., Larsen, B. N. and Leimer, H. G. (1990). Independence properties of directed Markov fields. *Networks* **20**, 491–505.
- 5 Dawid, A. P. (1998). Conditional independence. *Encyclopedia of Statistical Sciences*, Update Volume 2, edited by S. Kotz, C. B. Read and D. L. Banks. Wiley-Interscience, 146–155.
- 6 Dawid, A. P. (2001). Some variations on variation independence. In *Artificial Intelligence and Statistics 2001*, edited by T. Jaakkola and T. Richardson. Morgan Kaufmann, 187–191.
- 7 Dawid, A. P. (2001). Separoids: A mathematical framework for conditional independence and irrelevance. *Ann. Math. Artificial Intelligence* **32**, 335–372.

3.4 Definability Issues in Team Semantics

Pietro Galliani (University of Helsinki, FI)


License  Creative Commons BY 3.0 Unported license
© Pietro Galliani

I will present a number of extensions and variants of Dependence Logic and discuss a number of known results (plus a couple of new ones) concerning interdefinability and expressivity.

Furthermore, I will discuss how these results can be integrated into a general theory of definability in team semantics.

3.5 Robust Constraint Satisfaction and hidden variable detection in quantum mechanics

Georg Gottlob (University of Oxford, GB)

License  Creative Commons BY 3.0 Unported license
© Georg Gottlob
Joint work of Abramsky, Samson; Gottlob, Georg; Kolaitis, Phokion

Motivated by considerations in quantum mechanics, we introduce the class of robust constraint satisfaction problems in which the question is whether every partial assignment of a certain length can be extended to a solution, provided the partial assignment does not violate any of the constraints of the given instance. We explore the complexity of specific robust colorability and robust satisfiability problems, and show that they are NP complete. We then use these results to establish the computational intractability of detecting local hidden-variable models in quantum mechanics.

3.6 Axiomatizing first-order consequences in independence logic

Miika Hannula (University of Helsinki, FI)

License  Creative Commons BY 3.0 Unported license
© Miika Hannula

Independence logic cannot be effectively axiomatized. However, first-order consequences of independence logic sentences can be axiomatized. Here we give an explicit axiomatization and sketch a proof of it being complete in this sense.

3.7 Compositionality: its history and formalism

Wilfrid Hodges (Okehampton, Devon)

License © Creative Commons BY 3.0 Unported license
© Wilfrid Hodges

Main reference W. Hodges, “Formalizing the relationship between meaning and syntax,” in *The Oxford Handbook of Compositionality*, ed. M. Werning, W. Hinzen and E. Machery, Oxford University Press, pp. 245–261, 2012.

URL <http://wilfridhodges.co.uk/semantics13.pdf>

A tutorial talk, covering the history of the idea of compositionality from its origins to its role in the discovery of the team semantics.

3.8 A double-slit experiment for non-classical dependencies in decision-making

Pierfrancesco La Mura (HHL Leipzig, DE)

License © Creative Commons BY 3.0 Unported license
© Pierfrancesco La Mura

Main reference To appear in *Topics in Cognitive Science*, special issue on Quantum Cognition, 2013.

We discuss the possible nature and role of non-physical entanglement, and the classical vs. non-classical interface, in models of human decision-making. We also introduce an experimental setting designed after the double-slit experiment in physics, and discuss how it could be used to discriminate between classical and non-classical interference effects in human decisions.

3.9 Dependence, Independence, Logic

Sebastian Link (University of Auckland, NZ)

License © Creative Commons BY 3.0 Unported license
© Sebastian Link

Joint work of Hartmann, Sven; Link, Sebastian


Main reference S. Hartmann, S. Link, “The implication problem of data dependencies over SQL table definitions: Axiomatic, algorithmic and logic characterizations,” *ACM Trans. Datab. Syst.* 37(2), Article 13, May 2012.

URL <http://dx.doi.org/10.1145/2188349.2188355>

Data dependencies enforce meaningful properties of a given application domain within a database system. Dependencies are essential for the design of databases, and facilitate many data processing tasks. Conditional independencies capture structural aspects of probability distributions, deal with knowledge in artificial intelligence, and help with learning and reasoning in intelligent systems. Reasoning about data dependencies or about conditional independencies is infeasible in general. However, expressive yet efficient subclasses have been identified in both cases, for examples, multivalued dependencies and saturated conditional independencies. These findings are based on the classic assumption that the underlying data are complete. In practice, data are missing or unknown, and structural or sampling zeros occur. In this seminar expressive and efficient notions of multivalued dependencies and saturated conditional independencies are presented in the presence of incomplete data. It is demonstrated that the implication problem for multivalued dependencies, for saturated conditional independencies, and for a propositional fragment of S-3 logic coincide. The results show that reasoning in the presence of incomplete data soundly approximates reasoning in the presence of complete data; and that reasoning can be done in almost linear time in the input.

3.10 Dependence and Independence in Social Choice

Eric Pacuit (Tilburg University, NL)

License  Creative Commons BY 3.0 Unported license
© Eric Pacuit

I surveyed a number of key results in social choice theory (e.g., Arrow’s Impossibility Theorem, May’s Characterization of Majority Rule). My goal was to highlight the notions of independence and dependence found in this literature.

3.11 Dependence logic

Jouko Väänänen (University of Helsinki, FI)

License  Creative Commons BY 3.0 Unported license
© Jouko Väänänen

Main reference J. Väänänen, “Dependence Logic – A New Approach to Independence Friendly Logic,” London Mathematical Society Student Texts, No. 70, Cambridge University Press, ISBN 9780521876599, 2007.

URL <http://www.cambridge.org/gb/knowledge/isbn/item1174541>

This is an opening introductory tutorial on the basic ideas of dependence and independence logic. I review team semantics, the main driving force behind dependence logic. I emphasise the ubiquitousness and potential applications of dependence and independence concepts throughout science and humanities.

3.12 Uniform definability of connectives in propositional dependence logic

Fan Yang (University of Helsinki, FI)

License  Creative Commons BY 3.0 Unported license
© Fan Yang

Propositional dependence logic is the propositional variant of first-order dependence logic [4]. Intuitionistic implication and intuitionistic disjunction in the setting of team semantics were introduced in [1]. Propositional dependence logic extended with intuitionistic implication and intuitionistic disjunction, called propositional intuitionistic dependence logic, is essentially equivalent to inquisitive logic [2]. Huuskonen (2012) showed that propositional intuitionistic dependence logic is equivalent to propositional dependence logic. It follows that every formula with intuitionistic disjunction and intuitionistic implication can be translated into a formula without these two connectives. In this talk, we show that although such a non-uniform translation exists, neither of intuitionistic disjunction and intuitionistic implication is uniformly definable in propositional dependence logic. This work was inspired by [3].

References

- 1 S. Abramsky, and J. Väänänen, From IF to BI. *Synthese* 167, 2 (2009), pp 207–230.
- 2 I. Ciardelli, and F. Roelofsen, Inquisitive Logic. *Journal of Philosophical Logic*, 2011, 40(1), 55–94.
- 3 P. Galliani, Epistemic Operators in Dependence Logic. *Studia Logica*, April 2013, Volume 101, Issue 2, pp 367–39.
- 4 J. Väänänen, *Dependence Logic: A New Approach to Independence Friendly Logic*, Cambridge University Press, 2007



■ **Figure 1** Afternoon walk on Wednesday.



■ **Figure 2** The tower at Dagstuhl Castle.

Participants

- Samson Abramsky
University of Oxford, GB
- Dietmar Berwanger
ENS – Cachan, FR
- Olaf Beyersdorff
University of Leeds, GB
- Andreas R. Blass
University of Michigan – Ann Arbor, US
- Julian Bradfield
University of Edinburgh, GB
- Panayiota Constantinou
University of Cambridge, GB
- Nadia Creignou
Université de Marseille, FR
- Anuj Dawar
University of Cambridge, GB
- A. Philip Dawid
University of Cambridge, GB
- Arnaud Durand
University Paris-Diderot, FR
- Johannes Ebbing
Leibniz Univ. Hannover, DE
- Uwe Egly
TU Wien, AT
- Fredrik Engström
Göteborg University, SE
- Pietro Galliani
University of Helsinki, FI
- Georg Gottlob
University of Oxford, GB
- Erich Grädel
RWTH Aachen, DE
- Miika Hannula
University of Helsinki, FI
- Lauri Hella
University of Tampere, FI
- Asa Hirvonen
University of Helsinki, FI
- Wilfrid Hodges
Okehampton, Devon, GB
- Theo Janssen
University of Amsterdam, NL
- Phokion G. Kolaitis
University of California – Santa Cruz, US
- Juha Kontinen
University of Helsinki, FI
- Antti Kuusisto
University of Tampere, FI
- Pierfrancesco La Mura
HHL Leipzig, DE
- Sebastian Link
University of Auckland, NZ
- Allen L. Mann
Birkhäuser Science – New York, US
- Arne Meier
Leibniz Univ. Hannover, DE
- Eric Pacuit
Tilburg University, NL
- Tero Tulenheimo
University of Lille, FR
- Jouko Väänänen
University of Helsinki & University of Amsterdam
- Jonni Virtema
University of Tampere, FI
- Heribert Vollmer
Leibniz Univ. Hannover, DE
- Dag Westerstahl
University of Stockholm, SE
- Fan Yang
University of Helsinki, FI



Mechanisms of Ongoing Development in Cognitive Robotics

Edited by

Jacqueline Fagard¹, Roderic A. Grupen², Frank Guerin³, and Norbert Krüger⁴

1 Université Paris Descartes, FR, jacqueline.fagard@parisdescartes.fr

2 University of Massachusetts – Amherst, US, gruppen@cs.umass.edu

3 University of Aberdeen, GB, f.guerin@abdn.ac.uk

4 University of Southern Denmark – Odense, DK, norbert@mimi.sdu.dk

Abstract

In cognitive robotics “ongoing development” refers to the ability to continuously build on what the system already knows, in an ongoing process, which acquires new skills and knowledge, and achieves more sophisticated levels of behaviour. Human infants are possibly the best known demonstrators of this ability; developmental psychology has many results documenting what infants can and cannot do at various ages, however we know very little about the mechanisms underlying the development. On the robotics side, creating a computational system which displays ongoing development is still an unsolved problem. There are major unsolved questions regarding the mechanisms of ongoing development, in both biological and artificial systems; for example: how to transfer existing skills to a new context, how to build on existing skills, and how to represent knowledge (or skills). The primary aim of the seminar was to bring together researchers from two communities (developmental robotics and infant developmental psychology) in order to spawn new collaborative research projects which will advance our scientific understanding of the mechanisms underlying ongoing development (whether in infants or robots). We especially focused on perception, understanding and manipulation skills relating to physical objects in the world, and the skills which infants acquire in approximately the 4-24 months period. The main outcomes of the seminar were ideas about how the communities could work together to advance their respective goals. This requires psychologists to become computer scientists to some degree, and computer scientists to become psychologists. In addition each may need to be willing to help to solve some challenge problems posed by the other community in order to have their challenges tackled in turn.

Seminar 11.–15. February, 2013 – www.dagstuhl.de/13072

1998 ACM Subject Classification I.2 Artificial Intelligence, I.2.0 General: Cognitive simulation, Philosophical foundations, I.2.6 Learning, I.2.9 Robotics, Manipulators, I.2.10 Vision and Scene Understanding

Keywords and phrases Developmental psychology, Infancy, Motor skill development, Perceptual development, Origins of concepts, Developmental robotics, Affordances, Intrinsic motivation, Transfer of skills/knowledge

Digital Object Identifier 10.4230/DagRep.3.2.55



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Mechanisms of Ongoing Development in Cognitive Robotics, *Dagstuhl Reports*, Vol. 3, Issue 2, pp. 55–91

Editors: Jacqueline Fagard, Roderic A. Grupen, Frank Guerin, and Norbert Krüger



DAGSTUHL
REPORTS

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


1 Executive Summary

Jacqueline Fagard

Roderic A. Grupen

Frank Guerin

Norbert Krüger

License  Creative Commons BY 3.0 Unported license

© Jacqueline Fagard, Roderic A. Grupen, Frank Guerin, and Norbert Krüger

In cognitive robotics “ongoing development” refers to the ability to continuously build on what the system already knows, in an ongoing process, which acquires new skills and knowledge, and achieves more sophisticated levels of behaviour. Human infants are possibly the best known demonstrators of this ability; developmental psychology has many results documenting what infants can and cannot do at various ages, however we know very little about the mechanisms underlying the development. On the robotics side, creating a computational system which displays ongoing development is still an unsolved problem. There are major unsolved questions regarding the mechanisms of ongoing development, in both biological and artificial systems; for example: how to transfer existing skills to a new context, how to build on existing skills, and how to represent knowledge (or skills).

The primary aim of the seminar was to bring together researchers from two communities (developmental robotics and infant developmental psychology) in order to spawn new collaborative research projects which will advance our scientific understanding of the mechanisms underlying ongoing development (whether in infants or robots). We especially focused on perception, understanding and manipulation skills relating to physical objects in the world, and the skills which infants acquire in approximately the 4-24 months period.

Working groups were formed in the areas of (i) transfer of means/skills; (ii) motor skills/manipulation; (iii) concepts/representations; (iv) motivation; (v) visual perception. These discussed gaps between what infants and robots can do and what research might close the gap. In discussion groups the most significant issue that was raised (and discussed at length) was how to get psychologists and roboticists talking together and doing research together, as there seems to exist a wide gap between the communities. It was concluded that there was a need for psychologists to become computer scientists and computer scientists to become psychologists; i.e. that the meeting of the two fields would not happen simply by people getting together in a room, but that the meeting must happen inside individual heads. Furthermore challenge problems were posed by each of the two respective communities; challenges which they would like the other community to work on.

2 Table of Contents

Executive Summary

Jacqueline Fagard, Roderic A. Grupen, Frank Guerin, and Norbert Krüger 56

Desired Seminar Outcomes and Progress

Desired Tangible Outcomes 60

Social Outcomes 60

Overview of Talks

What are intrinsic motivations? A biological and computational perspective
Gianluca Baldassarre 60

What babies do that might be hard for robots to do
Emily W. Bushnell 61

Learning Language to Describe the Activities in Videos
Paul R. Cohen 62

Observational learning of tool use: Understanding the goal of the experimenter can
enhance infants' learning of a use of a novel tool
Rana Esseily 62

Infants' failure to retrieve an out-of-reach toy with a rake: what is lacking until 18
months?
Jacqueline Fagard 63

Mechanisms for Development of Sensory Abstraction
Severin Fichtl 65

Model-Based Belief Dynamics for Manipulation Planning
Roderic A. Grupen 66

What can we learn from infants' reaches to out-of-reach objects?
Beata Joanna Grzyb 67

The Structure of Knowledge in Development
Frank Guerin 68

Sensorimotor Loop Simulations for Tool-Use
Verena V. Hafner 69

Body schema in humans and animals and how to learn and model it in robots
Matej Hoffmann 70

Thinking Like A Child: The Role of Surface Similarities in Stimulating Creativity
Bipin Indurkha 70

Affordances, Verbs, Nouns and Adjectives
Sinan Kalkan 71

Robots, Skills and Symbols
George Konidaris 71

Remarks to Frank Guerin's talk
Norbert Krueger 72

Constructing the Foundations of Commonsense Knowledge <i>Benjamin Kuipers</i>	72
Building Tool Use from Object Manipulation <i>Jeffrey J. Lockman</i>	72
Constructing Space <i>J. Kevin O'Regan</i>	73
Learning from multiple motives. A reservoir computing approach <i>Mohamed Oubbati</i>	74
Developmental Mechanisms for Autonomous Life-Long Skill Learning in Robots and Humans <i>Pierre-Yves Oudeyer</i>	75
Learning Much From Little Experience <i>Justus Piater</i>	76
What do infants perceive from the spatial relations between objects? Data from 6- to 20-months-old infants <i>Lauriane Rat-Fischer</i>	76
What Robot(ic)s might learn from Children <i>Helge Ritter</i>	76
Meta-Morphogenesis theory as background to Cognitive Robotics and Developmental Cognitive Science <i>Aaron Sloman</i>	77
Think Like a Child: Creativity, Perceptual Similarity, Analogy and How to Make Adults Think like a Child <i>Georgi Stojanov</i>	78
What Infants Can Teach Us About The Way We Program Robots <i>Alexander Stoytchev</i>	79
Unsupervised Discovery of Actions and Action Possibilities <i>Emre Ugur</i>	80
Do We Need Models to Develop Robot Vision? <i>Markus Vincze</i>	80
Piaget for Robots: Implementing Accommodation and Assimilation in a Machine <i>Florentin Wörgötter</i>	81
Working Groups	
Group on transfer of means/skills	81
Group on motor skills/manipulation	81
Group on concepts/representations	84
Group on motivation (e.g. what to explore, and what is not interesting)	85
Visual Perception	86
Miscellaneous points spanning above subareas	86

Kevin's Game	
Part A	87
Part B	88
Challenge Problems	89
Participants	91

3 Desired Seminar Outcomes and Progress

3.1 Desired Tangible Outcomes

It is the aim that the following outcomes will be pursued after the conclusion of the seminar:

- A “Roadmap Paper” to understand ongoing development by creating a working model.
 - This idea is being taken forward.
 - The idea is not to state times (e.g. achieve in 5, 10 yrs), but milestones, and parallel/serial work.
 - The roadmap should tell one what to do in what order, where to focus, and where to go next.
- Journal special issue
 - TAMD special issue should have deadline coming in October 2013
 - A special section for a psychology journal is also considered (Jeff Lockman)
- An edited book along the lines of the book “Stone Knapping: The Necessary Conditions for a Uniquely Hominin Behaviour (McDonald Institute Monographs) [Hardcover] V. Roux (Author, Editor), B. Bril (Editor)”. The idea would be to present a similar volume which addresses the question of what is missing from robots and present in humans which permits ongoing development.
 - Gianluca Baldassarre said this might be difficult on the CS/robotics side, as books are not valued here, hence it is difficult to get people to contribute. In any event this is a long term plan (5-10) years and another meeting of potential contributors would take place first. It was agreed that a sequel to this seminar would be appropriate in three years’ time.

3.2 Social Outcomes

It was an aim of the seminar to get new pairs of people working to identify work they want to undertake together: possible papers, projects, psychology experiments, coordination/integration of computational work in different labs.

4 Overview of Talks

4.1 What are intrinsic motivations? A biological and computational perspective

Gianluca Baldassarre (ISTC-CNR – Rome, IT)

License © Creative Commons BY 3.0 Unported license
© Gianluca Baldassarre

Main reference G. Baldassarre, “What are intrinsic motivations? A biological perspective,” in Proc. of the 2011 IEEE Int’l Conf. on Development and Learning (ICDL’11), Vol.2, pp. 1–8, IEEE, 2011.

URL <http://dx.doi.org/10.1109/DEVLRN.2011.6037367>

The concept of “intrinsic motivation”, initially proposed and developed within psychology, is gaining an increasing attention within cognitive sciences for its potential to produce open-ended learning machines and robots. However, a clear definition of the phenomenon is not yet available. This presentation aims to clarify what intrinsic motivations are from a biological perspective and from a computational perspective. To this purpose, it first shows how intrinsic motivations can be defined contrasting them to extrinsic motivations from an evolutionary (and engineering) perspective: whereas extrinsic motivations guide

learning of behaviours that directly increase fitness (or satisfy the user/designer purposes), intrinsic motivations drive the acquisition of knowledge and skills that contribute to produce behaviours that increase fitness (or user satisfaction) only in a later stage. Given this key difference, extrinsic motivations generate learning signals on the basis of events involving body homeostatic regulations (accomplishment of user purposes), whereas intrinsic motivations generate transient learning signals mainly based on events taking place within the brain itself (or within the controller of the robot/intelligent machine). These ideas are supported by presenting (preliminary) taxonomies and examples of biological mechanisms underlying the two types of motivations, and also by linking them to some of the most commonly used mechanisms proposed by the literature to implement intrinsic motivations in robots and machines.

4.2 What babies do that might be hard for robots to do

Emily W. Bushnell (Tufts University, US)

License © Creative Commons BY 3.0 Unported license
© Emily W. Bushnell

Joint work of Bushnell, Emily W.; Brugger, Amy; Lariviere, Leslie. A.; Mumme, Donna. L.; Sidman, Jason; Yang, Dahe J.

Main reference D.J. Yang, J. Sidman, E.W. Bushnell, “Beyond the information given: Infants’ transfer of actions learned through imitation,” *Journal of Experimental Child Psychology*, Vol. 106, Issue 1, pp. 62–81, 2010.

URL <http://dx.doi.org/10.1016/j.jecp.2009.12.005>

Learning by imitation is a very efficient, prominent, and productive learning mechanism during human infancy. If robots are to learn as infants do, they will have to be built to imitate. However, infant imitation is not a simple, straight-forward process. In this presentation, I discuss some aspects of infant imitation that may be difficult to program into robots. First, infant imitation is “optional” – if babies see a sequence of actions, they may subsequently imitate only some of these behaviors and not others. Research from my lab indicates that infants are more likely to imitate an action when it is causally relevant to achieving a goal than when it is unnecessary to the goal. They are also more likely to imitate an action followed by an effect than an unadorned action, but less so if they already know another way to create that effect. Infant imitation also interacts with their mind-reading abilities; infants will imitate irrelevant and inefficient actions if they perceive social cues that the demonstrator means for them to do so, and they will perform behaviors they perceive as intended by a demonstrator in preference to the behaviors actually observed. Furthermore, the weighting of these various parameters affecting imitation – causal relevance, efficacy, efficiency, social cueing, etc. – is not fixed. A given parameter may override another in one context, whereas in a different context their influence may be reversed. Capturing this flexibility within a robotic learning system may be a challenge.

Infant imitation is also “generative”. Work in my lab shows that by 15 months of age, infants robustly transfer actions learned by imitation to new object contexts which have not been demonstrated for them. Such transfer is a developmental and also a phylogenetic achievement; very young infants and non-human primates do not likewise transfer learned responses across object contexts so readily. Transfer from imitation hinges on extracting an action from the observed action-object-effect context, so the action becomes a distinct entity (representation) that may be combined in a grammar-like way with other objects to potentially produce new effects. Thus transfer enhances the value of learning by imitation considerably, as acquired actions serve to guide infants’ subsequent exploration so that is more focused and non-random. However, the capacity to transfer also requires some

constraints to limit instances of “negative transfer” or overgeneralization. Identifying the biases that both propel and restrain infants’ transfer from imitation is a goal for further developmental research, and likewise programming such priors into intelligent machines is a task for roboticists.

References

- 1 Brugger, A., Lariviere, L. A., Mumme, D. L., and Bushnell, E. W. (2007). Doing the right thing: Infants’ selection of actions to imitate from observed event sequences. *Child Development*, 78, 806 -824.
- 2 Yang, D.J., Sidman, J., and Bushnell, E. W. (2010) Beyond the information given: Infants’ transfer of actions learned through imitation. *Journal of Experimental Child Psychology*, 106, 62 -81.
- 3 Yang, D. J., Bushnell, E.W., Buchanan, D. W., & Sobel, D. M. (in press). Infants’ use of contextual cues in the generalization of causal actions. *Journal of Experimental Child Psychology*.

4.3 Learning Language to Describe the Activities in Videos

Paul R. Cohen (University of Arizona – Tucson, US)


License  Creative Commons BY 3.0 Unported license
© Paul R. Cohen

Developmental robotics deals with learning fundamental cognitive structures and processes by interacting with the environment over long time frames. I am particularly interested in language learning and learning in service of vision. As specific examples I would describe our work in DARPA’s Mind’s Eye initiative, where the task is to generate natural language descriptions of surveillance videos; our work on learning deep semantics for spatial language; and a new project called the Bayesian Blackboard, an architecture for integrating top-down and bottom-up processes in a probabilistically sound way.

<http://w3.sista.arizona.edu/~cohen/>

4.4 Observational learning of tool use: Understanding the goal of the experimenter can enhance infants’ learning of a use of a novel tool

Rana Esseily (Université Paris Ouest Nanterre, FR)

License  Creative Commons BY 3.0 Unported license
© Rana Esseily

Joint work of Esseily, Rana; Rat-Fischer, Lauriane; O’Regan, Kevin; Fagard, Jacqueline
Main reference R. Esseily, L. Rast-Fischer, K. O’Regan, J. Fagard, “Understanding the experimenter’s intention improves 16-month-olds’ observational learning of the use of a novel tool,” *Cognitive Development*, Vol. 28, Issue 1, pp. 1–9, 2013.

URL <http://dx.doi.org/10.1016/j.cogdev.2012.10.001>

In the beginning of the second year of life, infants become highly capable at learning by observation new means end actions such as opening a box with one hand to retrieve an object with the other hand (Esseily et al., 2010). However tool use studies show that before the end of the second year, infants fail to learn by observation how to use a tool to retrieve an out of reach object (Fagard et al., 2011). The aim of our studies was to investigate why do 16-month-old infants who have already developed some observational learning capacities, fail

to learn by observation a tool use action. We claim that in order to learn by observation a new target action, infants have to understand the goal of that action. Thus, if infants do not understand the goal of using the tool, they will not be able to predict and anticipate the demonstrator's actions and thus to relate those actions (the experimenter pulling the tool) with their consequences (the toy coming within reach). We tested this hypothesis by showing 16-month-old infants an explicit demonstration of the goal of the experimenter before demonstrating the target action. We tested 65 16-month-old infants on a tool use action consisting in grasping a rake-like object to retrieve an out of reach toy. Infants were randomly assigned to one of 5 groups: spontaneous group (spontaneous manipulation of tool use), classic demonstration group (observation of a model performing directly the demonstration of the target action), intention prior to demonstration group (observation of a model showing her goal by stretching her hand toward the toy before performing the demonstration of the target action), and two additional groups to control for local and stimulus enhancement. The results show that infants in the intention prior to demonstration group performed significantly better than infants in all other groups. However the results also show that infants' performance was not perfect and even though infants made a connection between the toy and the tool, the toy was not always successfully retrieved. One of the reasons learning was not perfect can be that the experimenter's goal was not sufficiently enhanced. Thus, in another ongoing study, we aim at making the goal of the experimenter even more salient by providing infants a situation where the goal is incongruous with the action performed, thus attracting their attention to that goal (the experimenter throws away the toy as soon as she retrieves it using the tool). The preliminary results show that the incongruity makes the situation humoristic for some infants and it is precisely those infants who laugh at the demonstration, who learn perfectly the target action; whereas infants who do not laugh, do not learn by observation how to retrieve the toy using the tool. Hypotheses regarding the underlying mechanisms responsible for these results will be discussed.

4.5 Infants' failure to retrieve an out-of-reach toy with a rake: what is lacking until 18 months?

Jacqueline Fagard (Université Paris Descartes, FR)

License © Creative Commons BY 3.0 Unported license
© Jacqueline Fagard

Main reference L. Rat-Fischer, J.K. O'Regan, J. Fagard, "The emergence of tool use during the second year of life," *Journal of Experimental Child Psychology*, Vol. 113, Issue 3, pp. 440-446, 2012.

URL <http://dx.doi.org/10.1016/j.jecp.2012.06.001>

Both robotics and developmental psychology explore how an organism becomes autonomous, learns new abilities, and builds on these abilities. In other words both investigate the emergence of higher cognitive functions through learning and development, from perception and action. We choose the emergence of tool use in infants to tackle this question. Tools allow one to overcome the limits of one's body in interacting with the environment. In everyday life infants can use a toothbrush or a spoon not long after their first year. At the same age they may even be able to use a rake-like tool to retrieve a toy in an experimental situation if the toy is placed inside the tool, thus if no spatial gap lies between rake and toy and the toy may come by simple contingency as soon as the rake is moved (Bates, Carlsonluden, & Bretherton, 1980; Brown 1990; van Leeuwen, Smitsman, & van Leeuwen, 1994). It is thus amazing to observe that it is not until 18 months that, in normal conditions, infants succeed

at using a rake to retrieve a toy when the latter is placed at distance and to the side of the rake, an observation that we did in our longitudinal (6 infants) and cross-sectional studies (60 infants). In both studies infants failed spontaneously but also after demonstrations from an adult of how to use the rake (Rat-Fischer, O'Regan, & Fagard, 2012; Fagard, Rat-Fisher, & O'Regan, 2012). This late success raises the question of what does it take to an infant to learn to use a new tool. What do infants need to use the rake? We see at least five components of success.

1/ Being able to grasp and move the rake: they obviously can do that (they can grasp the rake, bang on the table, throw it away, etc. at 12 months and even earlier).

2/ Being willing to retrieve the toy: even though it looks sometimes that the toy itself is less interesting than the raking of the toy (quite often they give the toy back to the experimenter, like a dog with a ball), they always indicate that they want the toy since pointing toward the toy in a begging gesture is the most frequent behaviour before success. However we have examples of high motivation being efficient to increase the rate of success (food as a toy) and of too high motivation leading to regression (object too much desired leading to crying and fussing)

3/ Knowing that the rake would allow bringing the toy closer (functionality of the rake): if it was the only problem, they would succeed after the first demonstration from the adult, which is not the case.

4/ Knowing where should be positioned the rake (behind the toy).

5/ Being able to precisely position the rake behind the toy: if it was the only problem, they would try hard and fail. This behaviour of near-success is observed very late, and usually gives way to success within the same session.

Among these five components, some are more on the manual control side (1, which is obviously not a limitation at the age tested, and 5), some on the motivational side (2), and some more on the cognitive side (3, 4). To explore further which one of these components is the most limiting constraint, we first showed an infant repeated demonstrations of using a rake to bring an object closer between 9 and 12 months. He never had the opportunity to manipulate the rake himself (pure visual familiarization). We then followed him longitudinally from 12 to 18 months, in the same conditions as for the six infants of our longitudinal study. This was one way of testing whether understanding the functionality of the rake would help the infant succeed before 18 months, despite the lack of manual practice. Results showed that this infant was able to succeed at using the rake to retrieve a toy placed to the side of the rake much earlier than 18 months (some near-success at 12 months, a few successes at 13 and 14 months, many successes at 16 months). Most importantly, as opposed to all infants tested so far, this infant almost never rejected the rake. These results show that repeatedly observing the functionality of the rake helps succeed earlier. This indicates how component 3 is an important limiting constraint for tool use. However, from the observation of this infant, it was also clear that even when he tried to use the rake to bring the toy closer, it was extremely difficult for him to put the rake precisely behind the toy. Thus, components 5 (and may be 4) were also a limiting constraint. The most likely hypothesis, thus, is that it is the combination of all these components which are needed for success. According to Bruner, skill emerges from the addition of sub-routines which are slowly integrated into a successful behaviour (Bruner, 1970). Besides, a negative influence of the cognitive load of a task on the quality of the infant's movement and a negative influence of the motor load of a task on the infant's understanding of the task has been shown and explained by some motor-cognitive trade-off (Boudreau & Bushnell, 1996). This might be understood as a limitation of processing capacities or attentional resources when one or the other components

required for success is made more difficult. In the case of tool-use and our experiment of visual familiarization, it is conceivable that helping understand the functionality of the tool (component 3) frees the infant's mind to try hard on where to place the rake (component 4) and how to do it (component 5). To confirm this preliminary result and to compare the impact of the cognitive and manual-control components on success at tool use, we are now comparing two groups of infants, one group with only manual familiarization with the rake alone, and one group with only visual familiarization with the action of retrieving an out-of-reach object with the rake. Both groups are familiarized during five sessions before being tested at 16 months in the same conditions as in our previous studies.

4.6 Mechanisms for Development of Sensory Abstraction

Severin Fichtl (University of Aberdeen, GB)

License © Creative Commons BY 3.0 Unported license
© Severin Fichtl

Joint work of Fichtl, Severin; Alexander, John; Mustafa, Wail; Kraft, Dirk; Jorgensen, Jimmy; Krüger, Norbert; Guerin, Frank

We are currently interested in three areas, all related to ongoing developmental learning in robotics:

1) **Sensor differentiation:** Sensor differentiation: Infants extend their repertoire of behaviours from initially simple behaviours with single objects to complex behaviours dealing with spatial relationships among objects. We are interested in the mechanisms underlying this development in order to achieve similar development in artificial systems. One mechanism is sensorimotor differentiation, which allows one behaviour to become altered in order to achieve a different result; the old behaviour is not forgotten, so differentiation increases the number of available behaviours. Differentiation requires the learning of both, sensory abstractions and motor programs for the new behaviour; here[1] we focus only on one sensory aspect: learning to recognise situations in which the new behaviour succeeds. We experimented with learning these situations in a realistic physical simulation of a robotic manipulator interacting with various objects, where the sensor space includes the robot arm position data and a kinect based vision system. The mechanism for learning sensory abstractions for a new behaviour is a component in the larger enterprise of building systems which emulate the mechanisms of infant development.

2) **Intrinsic Motivation:** In order to deal with the realistic and high dimensional environments which we encounter in our Sensor differentiation research we have to apply some strategy in order to render the complex learning problems feasible. A standard approach to decrease complexity and increase convergence speed is dimensionality reduction, which transforms the state space by projecting it to a lower dimensional feature space. In our work, we have developed a variation of intrinsic motivation called Certainty Based Curiosity (CBC)[2] in order to efficiently explore the space to facilitate quick learning. The idea behind CBC is to label samples that are likely to add most information to the model. This is achieved by labelling the sample which the current model is most unsure about how to classify. To label a sample means to perform an action in a given environment and the different samples equate to different actions that are available to the agent. In contrast to other Intrinsic Motivation algorithms, like Intelligent Adaptive Curiosity, it actively reduces the amount of training needed to improve classifiers and predictors.

3) **Learning Spatial Object Relations which determine the Outcome of Actions** In order

to construct complex plans and to achieve elaborate tasks it is essential for an agent to understand the qualitative structure and spatial relations of the objects in its environment. Our agents' vision system uses kinect or stereo cameras to generate a 3D point cloud of its environment and from this extracts a texlet based representation[3] of the scene. From this texlet representation we extract relevant information about the spatial relation between objects and store this information in form of 2D relation histograms. This information is extracted by calculating certain relations between object texlets. In this work we use two different distance relations to learn spatial relations. First we calculate the absolute distance of two texlets in the X – Y plane, neglecting the difference in height. The other distance we calculate is the difference in height with respect to the texlet of object 2. From labelled histograms we train Random Forest models to recognise spatial relations. Preliminary experiments suggest that this is a valid approach to learning Spatial relations in 3D environments.

References

- 1 Fichtl, S., Alexander, J., Kraft, D., Jorgensen, J. A., Krüger, N., Guerin, F. (2012) Rapidly learning preconditions for means-ends behaviour using active learning, ICDL
- 2 Fichtl, S., Alexander, J., Kraft, D., Jorgensen, J. A., Krüger, N., Guerin, F. (2013) Learning object relationships which determine the outcome of actions, Paladyn (Special Issue on Advances in Developmental Robotics), <http://dx.doi.org/10.2478/s13230-013-0104-x>
- 3 Pugeault, N., Wörgötter, F., Krüger, N. (2010) Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics, International Journal of Humanoid Robotics (Special Issue on Cognitive Humanoid Vision)

4.7 Model-Based Belief Dynamics for Manipulation Planning

Roderic A. Grupen (University of Massachusetts – Amherst, US)

License © Creative Commons BY 3.0 Unported license
© Roderic A. Grupen

This presentation proposes a data-driven computational approach that accumulates both skills and experience. Skills and partial models of the world are the focus of an intrinsically motivated exploration driven by the difference between expectations and observations[Hart]. Examples are presented of a skill hierarchy accumulated by Dexter (the UMass bimanual humanoid) over the course of approximately four days of training using this approach. These skills include:

1. a policy (searchtrack) for searching for and then tracking visual features
2. a policy (reachgrasp) for reaching to and grasping an object
3. a policy (pick-and-place) for putting one object in contact with a second object
4. a few simple assembly policies (stacks of objects)

We introduce a Bayes filter for representing objects in terms of probabilistic models of how these actions cause effects and then formulate plans that optimize the information gain of the learning system[Sen]. The presentation concludes with new demonstrations of this framework configured to discriminate between objects by composing informative sequences of manual and visual actions.

4.8 What can we learn from infants' reaches to out-of-reach objects?

Beata Joanna Grzyb (Universitat Jaume I – Castellon de la Plana, ES)

License © Creative Commons BY 3.0 Unported license
© Beata Joanna Grzyb

The knowledge of one's own action capabilities and bodily characteristics plays a crucial role in perceptuo-motor behavior and hence needs to be incorporated, very early in life, in a bodily frame of reference for action. In general, the bodily frame of reference has to be updated throughout life to properly accommodate changes in perceptual, action or cognitive abilities. We investigated how infants' knowledge of their reachable space changes as their capabilities change over a relatively short developmental timescale. Reaching action provides a good measure of infants' body (and space) awareness, since to successfully reach for an object infants need to know not only the distance to the object, but also how far they can reach and lean without losing balance.

Five experiments compared 9- and 12-month olds in reaching tasks to targets at varying distances – manipulating the salience of the objects, the novelty of the motor act via added wrist weights, and the ordering of the target distances (random, near to far, far to near). The results show that older infants, 12-month-olds do not honor in their attempted reaches a boundary between targets at reachable and not reachable distances but reach to targets at patently unreachable distances. For the infants in our empirical studies, it is likely that few of the 9-month olds were walking or “cruising” upright while holding on to a support, but it is highly likely that many of the 12 month olds were walking or spending time in some form of pre-walking activity in an upright posture. Thus, the developmental change in the alignment between attempted and successful reaching distances could be related to the transition to walking.

We extended our Experiment 1 to include infants with different walking abilities: non-walkers, walkers with help, and independent walkers. The results of our extended Experiment show that walkers (with or without help) constantly reached for the nonreachable target, whereas non-walkers reached less showing better alignment of their reaching attempts to the distances they can reach. An examination of reaches to far distances as a function of trial block reveals that all infants reached with high probability the first time the object was presented. The reaches of non-walkers, however, decreased over trial blocks showing a clear adjustment of reaching behavior at the “near boundary” distances in the task. Walkers in contrast persistently reached to far distance regardless of the trial block showing little adjustment of their behavior with failures to make contact at the far distances .

The decision whether to reach or not for an object depends on many cognitive, motivational, social, perceptual and motor factors. Developmental changes in any or several of these components could be central to the present findings. With all these possibilities in mind, we offer three hypotheses as starting points for understanding why 12-month-old infants with more walking experience reach to targets at nonreachable distances. These hypotheses are: (i) the decreased ability to learn from negative outcome while reaching makes infants fine-tune their walking skill, (ii) the processes responsible for integration of different visual depth cues reorganize themselves at the onset of walking so as to incorporate information from self-motion-based depth cues, (iii) the representation of space changes with the onset of walking; near and far space are being integrated with the reaching and walking actions to constitute a coherent space representation. These hypotheses have been modeled and their plausibility subsequently tested in a robotic setup. The results of robot experiments showed that these hypotheses are not mutually exclusive and overlap in underlying mechanisms,

providing further evidences that goal directed reaching is a complicated skill with a long and protracted developmental course.

We advocate that new impetus to robotics can be given from these studies aiming at improving the efficacy of contemporary robotic systems. From a pragmatic point of view, a robot should be able to purposefully and consistently interact with its environment, by grounding its skills on the integration of different stimuli. Such skills could be based on building a representation of its nearby environment, representation which can be exploited for more precise and complex interactions with the environment components. The representation of space should be plastic, and change with the acquisition of new motor skills to properly reflect current robot action abilities. A robotic system should be provided with the ability to autonomously build a coherent representation of the environment for purposeful exploration and actuation in both peripersonal and extrapersonal space, through the active interaction with the environment in a similar way as infants do. Such joint studies should advance robotics, and give some insights for further understanding of human cognitive development, and the nature of embodied intelligence more generally.

4.9 The Structure of Knowledge in Development

Frank Guerin (University of Aberdeen, GB)

License © Creative Commons BY 3.0 Unported license
© Frank Guerin

Joint work of Guerin, Frank; Alexander, John; Fichtl, Severin

Main reference F. Guerin, N. Krüger, D. Kraft, “A Survey of the Ontogeny of Tool Use: from Sensorimotor Experience to Planning,” *IEEE Transactions on Autonomous Mental Development*, Vol. 5, Issue 1, pp. 18–45, 2012.

URL <http://dx.doi.org/10.1109/TAMD.2012.2209879>

Finding the appropriate representation for knowledge is long-standing difficulty in Artificial Intelligence (AI). Studying infants to get some idea of the types of representations they might be using is one possible way to attack the AI problem. Presumably infant representations are simpler and fewer than adult ones, and may provide a point of entry to understand adult ones. In previous work we analysed the development of infant behaviours and the parallel development of representations; two tracks which seem to bootstrap each other. Because of the close linkage of these two tracks, all representations, or fragments of representations are associated with behaviours, so that the infant knows what can be done with each concept or fragment thereof. The present talk focuses more on the representational track, and in particular identifies the need for structure in these representations so that concepts are constructed from components which (i) allows an infant to focus on facets of a complex concept, and to know the behavioural possibilities which facilitates planning); (ii) can be re-used by other concepts; (iii) can facilitate analogical reasoning via components shared with other concepts; (iv) can facilitate the construction of advanced concepts from components.

4.10 Sensorimotor Loop Simulations for Tool-Use

Verena V. Hafner (HU Berlin, DE)

License © Creative Commons BY 3.0 Unported license
© Verena V. Hafner

Joint work of Hafner, Verena V.; Schillaci, Guido; Lara, Bruno

Main reference G. Schillaci, B. Lara, V.V. Hafner, V.V. "Internal Simulations for Behaviour Selection and Recognition," in Proc. of the 3rd Int'l Workshop on Human Behaviour Understanding (HBU'12), LNCS, Vol. 7559, pp. 148–160, Springer, 2012.

URL http://dx.doi.org/10.1007/978-3-642-34014-7_13

In order to choose and perform appropriate actions, one can internally simulate an action and its predicted outcome. We implemented internal models based on pairs of inverse and forward models on a humanoid robot. The models were learned during body babbling. In the specific experiment, two different models were learned: one for the robot reaching an object with its arm, and one for the robot reaching an object with a tool, a stick attached to the robot's arm serving as an elongated end-effector. The robot could thus internally simulate the desired action for a given reaching position, and make a decision of whether to use the tool or not [2]. The same mechanism of internal models is used to recognise actions of others and even to distinguish between self and others [1, 6]. We are currently investigating the use of internal models to recognise human behaviour, e.g. in throwing [3]. The work is related to our previous work on body maps [5] and intrinsic motivation for exploratory learning [4].

References

- 1 Schillaci, G., Lara, B. and Hafner, V.V. (2012), Internal Simulations for Behaviour Selection and Recognition, in Human Behaviour Understanding 2012, A.A. Salah et al. (Eds.), Lecture Notes in Computer Science, Volume 7559, pp. 148-160.
- 2 Schillaci, G., Hafner, V. V., Lara, B. (2012), Coupled Inverse-Forward Models for Action Execution Leading to Tool-Use in a Humanoid Robot, Proceedings of the 7th ACM/IEEE International Conference on Human- Robot Interaction (HRI 2012), pp. 231-232, Boston, USA.
- 3 Frömer, R., Hafner, V.V. and Sommer, W. (2012), Aiming for the bull's eye: throwing investigated with event related brain potentials, Psychophysiology, Volume 49, Issue 3, pages 335-344, Wiley New York.
- 4 Oudeyer, P.-Y., Kaplan, F., Hafner, V.V. (2007), Intrinsic Motivation Systems for Autonomous Mental Development, IEEE Transactions on Evolutionary Computation, Special Issue on Autonomous Mental Development, Volume: 11, Issue: 2, pp. 265-286
- 5 Hafner, V.V. and Kaplan, F. (2008), Interpersonal Maps: How to Map Affordances for Interaction Behaviour, In: E. Rome et al. (Eds.): Affordance-Based Robot Control, LNAI 4760, pp. 1-15, Springer-Verlag Berlin Heidelberg
- 6 Schillaci, G., Hafner, V.V., Lara, B. and Grosjean, M. (2013), Is That Me? Sensorimotor Learning and Self-Other Distinction in Robotics, in Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2013), Tokyo, Japan.

4.11 Body schema in humans and animals and how to learn and model it in robots

Matej Hoffmann (Universität Zürich, CH)

License © Creative Commons BY 3.0 Unported license

© Matej Hoffmann

Main reference M. Hoffmann, H. Marques, A. Hernandez Arieta, H. Sumioka, M. Lungarella, R. Pfeifer, “Body schema in robotics: a review,” *IEEE Transactions on Autonomous Mental Development*, Vol. 2, Issue 4, pp. 304–324, 2010.

URL <http://dx.doi.org/10.1109/TAMD.2010.2086454>

The mechanisms that underlie body representations are co-responsible for many of the admiring capabilities of humans: combining information from multiple sensory modalities, controlling complex bodies, adapting to growth, failures, or using tools. These features are also desirable in robots. We review the concept of body schema in robotics. First, we briefly examine application-oriented research: being able to automatically synthesize, extend, or adapt a model of its body gives more autonomy and resilience to a robot. Second, we summarize the research area in which robots are used as tools to verify hypotheses on the mechanisms underlying biological body representations.

Finally, we present a case study, in which we performed a quantitative analysis of sensorimotor flows in a running quadruped robot using tools from information theory (transfer entropy). Starting from very little prior knowledge, through systematic variation of control signals and environment, we show how the agent can discover the structure of its sensorimotor space, identify proprioceptive and exteroceptive sensory modalities, and acquire a primitive body schema.

References

- 1 M. Hoffmann, H. Marques, A. Hernandez Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, “Body schema in robotics: a review,” *IEEE Trans. Auton. Mental Develop.*, vol. 2 (4), pp. 304–324, 2010.
- 2 N. Schmidt, M. Hoffmann, K. Nakajima, and R. Pfeifer, “Bootstrapping perception using information theory: Case studies in a quadruped robot running on different grounds,” *Advances in Complex Systems J.*, vol. 16, no. 6, 2012.

4.12 Thinking Like A Child: The Role of Surface Similarities in Stimulating Creativity

Bipin Indurkha (IIIT – Hyderabad, IN)

License © Creative Commons BY 3.0 Unported license

© Bipin Indurkha

Main reference B. Indurkha, “Thinking like a child: the role of surface similarities in stimulating creativity,” in *Proc. of the AAAI-2013 Spring Symposium Series: Creativity and (Early) Cognitive Development*, Stanford University, Palo Alto, California (USA), 2013.

URL <http://www.aaai.org/ocs/index.php/SSS/SSS13/paper/view/5725/5924>

An oft-touted mantra for creativity is: think like a child. We focus on one particular aspect of child-like thinking here, namely surface similarities. Developmental psychology has convincingly demonstrated, time and again, that younger children use surface similarities for categorization and related tasks; only as they grow older they start to consider functional and structural similarities. We consider examples of puzzles, research on creative problem solving, and two of our recent empirical studies to demonstrate how surface similarities can stimulate creative thinking. We examine the implications of this approach for designing creativity-support systems.

4.13 Affordances, Verbs, Nouns and Adjectives

Sinan Kalkan (Middle East Technical University – Ankara, TR)

License © Creative Commons BY 3.0 Unported license
© Sinan Kalkan

Joint work of Kalkan, Sinan; Sahin, Erol; Yuruten, Onur; Uyanık, Kadir Fırat; Çalışkan, Yiğit; Bozcuoğlu, Asil Kaan

Main reference O. Yürüten, K.F. Uyanık, Y. Çalışkan, A. Kaan Bozcuoğlu, Erol Şahin, Sinan Kalkan, “Learning Adjectives and Nouns from Affordances on the iCub Humanoid Robot,” in Proc. of the 12th Int’l Conf. on Adaptive Behavior (SAB’12), LNCS, Vol. 7426, pp. 330–340, Springer, 2012.

URL http://dx.doi.org/10.1007/978-3-642-33093-3_33

Learning and conceptualizing word categories in language such as verbs, nouns and adjectives are very important for seamless communication with robots. Along these lines, we linked the notion of affordance proposed by Gibson to (i) conceptualize verbs, nouns and adjectives, and (ii) demonstrate how a robot can use them for several important tasks in Robotics. For verbs, we compare different conceptualization views proposed by Psychologists over the years. Moreover, we show that there is an important underlying distinction between adjectives and nouns, as supported by recent findings and theories in Psychology, Language and Neuroscience.

4.14 Robots, Skills and Symbols

George Konidaris (MIT – Cambridge, US)

License © Creative Commons BY 3.0 Unported license
© George Konidaris

Joint work of Konidaris, George; Kuindersma, Scott; Barto, Andrew; Grupen, Roderic; Kaelbling, Leslie; Lozano-Perez, Tomas

My presentation approaches the problem of designing hierarchical control structures for robots that enable high-level symbolic reasoning, while ultimately remaining grounded in low-level sensorimotor control. The central theme of my talk is that the way to build such hierarchies is around learning sensorimotor skills. I first briefly cover my existing work on autonomous robot skill acquisition, which demonstrates that we are beginning to understand how to build robots that can discover skills through solving one task, and transfer them to more effectively solve future tasks. I will then consider the problem of symbolic planning using acquired skills—in particular, the question of which symbols are required to express and evaluate plans composed of sequences of skills. My (preliminary) work in this area shows that symbolic predicates corresponding to the preconditions and effects of the agent’s skills are sufficient for task-level planning in any problem, and necessary in some. The immediate implication of this is that a robot’s skills, environment and goal directly and completely specify the symbolic representation that it should use for planning. Since this representation is grounded and amenable to learning, a robot can acquire a symbolic representation appropriate for planning from its own experience.

4.15 Remarks to Frank Guerin’s talk

Norbert Krueger (University of Southern Denmark – Odense, DK)

License © Creative Commons BY 3.0 Unported license
© Norbert Krueger

As a reply to Frank Guerin’s talk, I dwell on four problems connected to developmental robotics:

1. Suitable hardware with enough dexterity and stability;
2. Defining meaningful initial behaviours;
3. Interaction of the behavioral track and representational track;
4. The definition of required prior knowledge.

4.16 Constructing the Foundations of Commonsense Knowledge

Benjamin Kuipers (University of Michigan, US)

License © Creative Commons BY 3.0 Unported license
© Benjamin Kuipers
Joint work of Kuipers, Benjamin; Pierce, David; Modayil, Joseph; Muga, Jonathan; Xu, Changhai
URL <http://web.eecs.umich.edu/~kuipers/research/whats-new.html>

An embodied agent experiences the physical world through low-level sensory and motor interfaces (the “pixel level”). However, in order to function intelligently, it must be able to describe its world in terms of higher-level concepts such as places, paths, objects, actions, goals, plans, and so on (the “object level”). How can higher-level concepts such as these, that make up the foundation of commonsense knowledge, be learned from unguided experience at the pixel level? I will describe progress on providing a positive answer to this question.

This question is important in practical terms: As robots are developed with increasingly complex sensory and motor systems, and are expected to function over extended periods of time, it becomes impractical for human engineers to implement their high-level concepts and define how those concepts are grounded in sensorimotor interaction. The same question is also important in theory: Must the knowledge of an AI system necessarily be programmed in by a human being, or can the concepts at the foundation of commonsense knowledge be learned from unguided experience?

4.17 Building Tool Use from Object Manipulation

Jeffrey J. Lockman (Tulane University, US)

License © Creative Commons BY 3.0 Unported license
© Jeffrey J. Lockman
Main reference J.J. Lockman, “A perception-action perspective on tool use development,” *Child Development*, 71(1), pp. 137–144, 2000.
URL <http://www.ncbi.nlm.nih.gov/pubmed/10836567>

Tool use has long been considered a cognitive advance. In contrast, in our work we suggest that tool use should be considered a problem of perceptuomotor adaptation in which individuals learn over an extended period of time how a tool changes the action possibilities or affordances of the hand.

Specifically, we have been studying the development of object manipulation in infants and how the behaviors involved in object manipulation transition to tool use. Our work indicates that infants adapt to changes in the properties of their hands when holding objects – a key component of tool use. In the second half year, they combine objects and surfaces together selectively, varying the actions that they perform based on the properties of the object in hand and the type of surface that the object contacts. Equally important, they do so when they hold handled objects: infants relate objects located at the end of the handle to surfaces appropriately, even though they are holding the handle and not the object directly.

Likewise, at a motor level, there is continuity in the behaviors that support the emergence of tool use. Our work employing motion tracking technology and kinematics indicates that in the second half year, infants naturally adapt the percussive up-down movements involved in banging in ways that make these actions ideally suited for instrumental tool use. We suggest that through spontaneous and repeated performance of banging behaviors, infants become skilled in controlling these behaviors, easing the transition toward incorporating these behaviors into such instrumental forms of tool use as hammering.

More broadly, we maintain that there is considerable utility in framing the problem of the emergence of tool use as an ongoing process of perceptuomotor adaptation. Such a process-oriented approach not only offers a way of linking the manual behaviors of infants to the tool use behaviors of older children, but also provides a way of viewing tool use as a product of more general perception- action processes that characterize the functioning of all organisms. This approach, in turn, may offer clues for promoting flexibility and learning in artificial agents that are designed for tool use.

4.18 Constructing Space

J. Kevin O'Regan (Université Paris Descartes, FR)

License © Creative Commons BY 3.0 Unported license

© J. Kevin O'Regan

Joint work of O'Regan, J. Kevin; Laffaquière, Alban; Terekhov, Alexander

URL <http://www.kevin-oregan.net>

Space seems to be given to us a priori, as a container which contains “stuff” like “objects” that can “move”. Among the objects are our “bodies”, which we can use to “act upon” the objects. These actions obey certain mathematical constraints dictated by the fact that space is three-dimensional and more or less Euclidean. But for our brains such goings-on are only nerve firings, and nerve firings can occur without there being such a thing as space outside the body. So how can the nerve firings lead to space? Evolution may have built our brains to create space, but how can this have come about? What patterns of nerve firing enable this to be done?

The problem is complicated by the fact that sensory receptors do not signal spatial properties directly. For example in vision, distance is confounded with size; position is confounded with eye and body posture. In hearing, distance must be deduced from a combination of intensity and inter-aural time differences. Another problem is that in order to deduce spatial properties of the environment, the brain needs to know something about the body's own spatial structure. And this is signalled by proprioceptive receptors whose outputs are also ambiguous. Finally, some a priori knowledge of body structure would seem to be necessary. So how can space arise from such a magma of neural firings?

When we think carefully about what space really is, we realize that we cannot hope to


find space as a feature of the environment that is directly perceived. Space is a construction that allows us to describe our worlds more conveniently. It is a collection of invariants linking neural output to neural input.

Extracting such invariants must allow the brain to define concepts like “body”, “environment”, “action”, “object”, “position”, “movement”, “distance”. Underlying such concepts are further facts like Separability: What I do here is generally not affected by what I do there; Relativity: Objects can be placed in the same spatial relation here as there; Impenetrability: Generally two objects cannot simultaneously occupy the same position; Group structure: some actions done on objects obey certain combinatorial rules independently of what the objects are. All of these notions are a few of many that are aspects of what we call space, but not all may be necessary for animals to function properly. Even humans’ notion of space may not rigorously encompass all these notions.

To understand better what are the basic concepts underlying the notion of space, a way to proceed is to build artificial agents of different degrees of complexity and see what notions of space they require in order to function. In my talk, I will present different agents illustrating different aspects of space, and will speculate how the underlying invariants could be learnt. I will show a naive agent that understands space as a set of “viewpoints from which things can be observed”. I will show how this agent can determine the dimension of this space and acquire its metric properties.

4.19 Learning from multiple motives. A reservoir computing approach

Mohamed Oubbati (Universität Ulm, DE)

License  Creative Commons BY 3.0 Unported license
© Mohamed Oubbati

Joint work of Oubbati, Mohamed; Palm, Günther

URL <http://www.uni-ulm.de/in/neuroinformatik/forschung/neurobotik.html>

Intrinsic-Extrinsic motivation can be viewed as another version of the mind-body dualism, such that intrinsic motives (e.g. curiosity) are those of the mind, while extrinsic motives (e.g. Hunger) are those of the body. The pressure exerted by such motives will keep a situated agent on the track to learn how to make trade-off between them in order to maintain its internal equilibrium. We are interested in studying how several motives influences the decision making process of the agent. We propose to integrate the concept of Reservoir Computing within the frame of Adaptive Dynamic Programming so that the agent learns to act and adapt in presence of several sources of reward. A single reservoir maybe trained to estimate several value functions simultaneously. This would be possible, because recurrent networks are able to learn from heterogeneous data, i.e. memory is in the recurrent activation, not only in the synaptic weights. In this way, a single reservoir could be able to cope with the conflicting demands imposed by different rewards.

4.20 Developmental Mechanisms for Autonomous Life-Long Skill Learning in Robots and Humans

Pierre-Yves Oudeyer (INRIA – Bordeaux, FR)

License © Creative Commons BY 3.0 Unported license
© Pierre-Yves Oudeyer

Joint work of Oudeyer, Pierre-Yves; Kaplan, Frédéric; Baranes, Adrien; Hafner, V.; Nguyen, Mai; Stulp, Freek; Lopes, Manuel

Main reference P.-Y. Oudeyer, A. Baranes, F. Kaplan “Intrinsically Motivated Learning of Real-World Sensorimotor Skills with Developmental Constraints,” in G. Baldassarre, M. Mirolli, (eds.), *Intrinsically Motivated Learning in Natural and Artificial Systems*, Springer, 2013.

URL http://dx.doi.org/10.1007/978-3-642-32375-1_13

URL <http://www.pyoudeyer.com/OudeyerBaranesKaplan13.pdf>

Developmental robotics studies and experiments mechanisms for autonomous life-long learning of skills in robots and humans. One of the crucial challenges is due to the sharp contrast between the high-dimensionality of their sensorimotor space and the limited number of physical experiments they can make within their life-time. This also includes the capability to adapt skills to changing environments or to novel tasks. To achieve efficient life-long learning in such complex spaces, humans benefit from various interacting developmental mechanisms which generally structure exploration from simple learning situations to more complex ones. I will present recent research in developmental robotics that has studied several ways to transpose these developmental learning mechanisms to robots [4], and which allowed to generate original hypothesis for mechanisms of infant development [2, 5, 7]. In particular, I will present and discuss computational mechanisms of intrinsically motivated active learning, which automatically select training examples of increasing complexity [6, 5, 2], or tasks through goal babbling [1], and their interaction with imitation learning [3], as well as maturation and body growth where the number of sensori and motor degrees-of-freedom evolve through phases of freezing and freeing [4, 7]. I will discuss them both from the point of view of modeling sensorimotor and cognitive development in infants and from the point of view of technology, i.e. how to build robots capable to learn efficiently in high-dimensional sensorimotor spaces.

References

- 1 Baranes, A., Oudeyer, P.-Y. (2013) Active Learning of Inverse Models with Intrinsically Motivated Goal Exploration in Robots, *Robotics and Autonomous Systems*, 61(1), pp. 49–73. <http://dx.doi.org/10.1016/j.robot.2012.05.008>.
- 2 Kaplan F. and Oudeyer P.-Y. (2007) In search of the neural circuits of intrinsic motivation, *Frontiers in Neuroscience*, 1(1), pp. 225–236.
- 3 Nguyen M., Baranes A. and P.-Y. Oudeyer (2011) Bootstrapping intrinsically motivated learning with human demonstrations, in proceedings of the IEEE International Conference on Development and Learning, Frankfurt, Germany.
- 4 Oudeyer P.-Y., Baranes A., Kaplan F. (2013) Intrinsically Motivated Learning of Real-World Sensorimotor Skills with Developmental Constraints, in *Intrinsically Motivated Learning in Natural and Artificial Systems*, eds. Baldassarre G. and Mirolli M., Springer.
- 5 Oudeyer P.-Y. Kaplan F. and V. Hafner (2007) Intrinsic motivation systems for autonomous mental development, *IEEE Transactions on Evolutionary Computation*, 11(2), pp. 265–286.
- 6 Schmidhuber, J. (1991) Curious model-building control systems, in: *Proc. Int. Joint Conf. Neural Netw.*, volume 2, pp. 1458–1463.
- 7 Stulp F., Oudeyer P.-Y. (2012) Emergent Proximo-Distal Maturation with Adaptive Exploration, in *Proceedings of IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-Epirob)*, San Diego, USA.

4.21 Learning Much From Little Experience

Justus Piater (Universität Innsbruck, AT)

License © Creative Commons BY 3.0 Unported license
© Justus Piater

Joint work of Piater, Justus; Szedmak, Sandor

Learning about objects and actions upon them should take advantage of previously-acquired knowledge of objects and actions. We introduce a framework for propagating object-action knowledge to new objects via action-specific similarity functions and action parameter transformations that are learned simultaneously from limited experience. The framework is based on a generalized regression algorithm capable of simultaneously learning object-object, object-action and action-action relations. These relations can be quite general and can represent notions such as similarities, parameter transformations or success probabilities.

4.22 What do infants perceive from the spatial relations between objects? Data from 6- to 20-months-old infants

Lauriane Rat-Fischer (Université Paris Descartes, FR)

License © Creative Commons BY 3.0 Unported license
© Lauriane Rat-Fischer

Joint work of Rat-Fischer, Lauriane; Florean, Cecilia; O'Regan, J. Kevin; Fagard, Jacqueline

From birth, infants have to coordinate vision and action to explore their environment. Around 10 months of age, they start reaching for out-of-reach objects by pulling a string attached to them, or a cloth on which the objects stand. This type of behavior, called means-end behavior, involves a key concept: the notion of spatial connectedness. Psychologists have shown that the presence/absence of a spatial gap between objects influences the performance of infants in such means-end behaviours. Infants are able to identify composite objects as a unique object when both are contiguous and move in a similar way. However, little is known about what infants perceive and understand from the spatial relationship between unmoving objects. When do infants start to consider the spatial connection as a relevant information to identify composite objects? And then, as soon as they understand that two contiguous objects are connected to each other, how do they apply these informations to solve problems involving the retrieval of out-of-reach objects? Are these informations sufficient in situations with more complex spatial relationships? Two behavioral studies [one involving an eyetracker] on infants aged 6 to 20 months gives us more informations on infants' expectations of composite objects, and their perception of spatial connectedness.

4.23 What Robot(ic)s might learn from Children

Helge Ritter (Neuroinformatics Group Faculty of Technology and Cluster of Excellence Cognitive Interaction Technology (CITEC), Bielefeld University, DE)

License © Creative Commons BY 3.0 Unported license
© Helge Ritter

The world of current robots is very different (and very far) from the world of children: there is a strong bias to “solve tasks”, to carry out “useful activities” and to deal with artificial, mostly rigid objects. In comparison, children primarily play, or engage in social behavior.

Their actions often exhibit a low degree of precision, but high variability and the ability to deal with soft and deformable objects.

Therefore, a first strong message from these differences concerns representational biases: we might question our bias for representing actions for geometrically well-defined, mostly rigid objects, and shaping behavior with a strong emphasis on well-defined goals and constraints. One challenge would be to come closer to abilities of children manifested in coping with deformable objects, such as clothes, food, or toys and materials such as plasticine.

A second major aspect are interfaces. Despite a high appreciation for robustness and flexibility, we still deal with rather rigid interfaces in robotics that are akin to “clockworks”. This is not only a matter of fact for most mechanical parts of our robots (such as rather rigid arms and grippers), but also the way we use and combine data structures which we can only accomplish by employing a very high degree of precision for their specification. In contrast, the interfaces that we see at work in children appear extremely exible: highly tactile hands, the emergence of language instead of precise codes, the ability to match and compare objects and situations based on patterns such as “Gestalts” and the formation of capabilities such as an effectively useable “theory of mind” to efficiently approximate the complex inner states of agents.

We believe that recent developments open up fruitful directions to come better to grip at least with some of these challenges. Our own work has been focused on a better understanding of how to replicate some of the “interface flexibility” of the human hand in robots, combining compliant movement grasping strategies and approaches for realizing touch and tactile behavior in a range of contexts. We have developed algorithms for dextrous manipulation, such as bimanual unscrewing actions, or folding of paper under real-time visual feedback. Another line of research has been the study of neural network approaches for Gestalt perception with the goal of action coordination based on Gestalt principles. This work is connected with further research lines within CITEC but outside of our group that emphasizes the role of social interaction for learning.

An overarching exciting aspect of “what children can do for robotics” is that they allow us to observe how the capability of cognitive interaction emerges, and how it does so within resource limits of processing and time that appear parsimonious in comparison to current large scale systems. How this parsimony is achieved is a major open question. An overarching long term goal in CITEC is to bring together several of the above-mentioned research strands into an architecture that allows us to bootstrap cognitive interaction from resource-parsimonious, guided growth and adaptivity, with only parsimonious blueprinting of initial “scaffolds” that direct the development of the system. As it appears, children are a strong existence proof and a great encouragement for successful solutions along such an approach.

4.24 Meta-Morphogenesis theory as background to Cognitive Robotics and Developmental Cognitive Science

Aaron Sloman (University of Birmingham, GB)

License © Creative Commons BY 3.0 Unported license
© Aaron Sloman

How could our minds and the rest of life have come from a cloud of dust?

Since its beginnings, we have made a lot of progress in AI and Cognitive Science in some areas, and done abysmally in others. That’s because there are some very deep problems about animal intelligence that have not been solved, and some have not even been noticed by

most researchers. These include problems connected with human mathematical competences (e.g. in geometry) and problem solving competences in other animals. There are also some allegedly hard problems that are actually not so hard – for people who have understood advances in virtual machinery, e.g. problems about the evolution, and implementation of qualia and various ways of being self conscious. I'll suggest a (Turing-inspired) strategy for trying to clarify the problems and gain ideas about the solutions produced by evolution that so far surpass anything we have in AI/Robotics. The strategy is to attempt to identify and explain transitions in the evolutionary history of biological information-processing – since microbes – since it is possible that current animals, including humans, still make important use of solutions to old problems, in ways that would not occur to us starting with computers. I call this the Meta-Morphogenesis project – partly inspired by reading Turing's paper on Morphogenesis (1952), and partly because mechanisms that produce new mechanisms, can also produce new mechanisms for producing new mechanisms! I have to select and over-simplify because the topic is far too large for a single talk. It's a project, not results, and this is just a sample. The focus is not physical morphology, or behaviour, but information-processing. My slides expand on these points.

Revised versions of the slides uploaded will be available at
<http://www.cs.bham.ac.uk/research/projects/cogaff/talks/#talk107>

4.25 Think Like a Child: Creativity, Perceptual Similarity, Analogy and How to Make Adults Think like a Child

Georgi Stojanov (The American University of Paris, FR)

License © Creative Commons BY 3.0 Unported license

© Georgi Stojanov

Joint work of Stojanov, Georgi; Indurkha, Bipin; Roda, Claudia; Kianfar, Dana

We put forward an assumption that creative thinking and creative behavior are an integral part of typical human cognitive development. Therefore, by looking into the early stages of this development, we can learn more about creativity. Conversely: by exploring creative behavior we might be able to learn something about early cognitive development. In addition, we believe that analogy is a core part of the creativity and developmental mechanisms. During the evolution, we have acquired enough innate knowledge which is crucial for bootstrapping the cognitive development in newborns, and continuously extending it mainly via analogical reasoning and behavior.

Some researchers of creativity make a distinction between historical-creativity (H-creativity) and psychological-creativity (P-creativity), which is about small creative deeds, probably new only to the individual performing them. According to our basic assumption, we also hypothesize that they share the same basic cognitive mechanisms, and that creative perception (in viewing an artifact) involves the same mechanisms that are responsible for generating creative artifacts. Moreover, these mechanisms can also be observed during cognitive development: a constant re-conceptualization of one's understanding of their environment in the process of agent-environment interaction, maturation, and education. If this hypothesis is accepted, then it suggests that by exercising and stimulating creative perception, we can also strengthen the ability to generate creative ideas and artifacts in the individual.

We have re-casted Piaget's theory of cognitive development by describing assimilation and accommodation as progressive reasoning by analogy starting from early analogizing in

terms of bodily sensory-motor schemas, to analogies in mature cognitive agents who have developed object representations and language.

One of the consequences of the above would be that if we would be able to induce child-like behavior in adults, this would result in increased ability for creative behavior and creative problem solving.

For example it is well known that young children have much shorter attention spans, or that they tend to focus on surface similarities in categorization tasks (see Bipin Indurkha's abstract). Another property unique for human infants is the pretend play. During a typical episode of pretend play, children detach themselves from the immediate here and now, and pretend that some objects actually represent other objects, depending on their scenario. For example, they may pretend that they are parents, dolls are their children, and that the banana is a phone which they may use to call their children's school. This is an example of analogy at work: seeing given object/situation as something else.

In this context (trying to make adults think like a child), we have designed a methodology where adult subjects (18 to 22 years old) are given a standard creativity test (in our case that was the ideational fluency test); while they were performing the test we were interrupting them with simple tasks irrelevant for the primary one. Our hypothesis was that this would lead to better performance as a result of widening and defocusing their attention. Although our first results were not encouraging — participants who were being interrupted did not show an increase in their ideational fluency, we believe that this methodology (making adults to “think like a child”) is a promising one and we are currently working on different experimental scenarios. One explanation may be that interruptions lead to stress? Another may be related to motivation: the subject didn't receive any compensation for participating in the experiment. The literature on the subject shows both positive and negative results in experiments similar to ours.

4.26 What Infants Can Teach Us About The Way We Program Robots

Alexander Stoytchev (Iowa State University, US)

License © Creative Commons BY 3.0 Unported license
© Alexander Stoytchev

Joint work of Stoytchev, Alexander; Sinapov; Jivko; Sukhoy; Vladimir; Sriffith; Shane

Main reference A. Stoytchev, “Baby Gym For Robots: A New Platform For Testing Developmental Learning Algorithms,” In Proc. of the 2011 AAAI Workshop on Lifelong Learning from Sensorimotor Experience, held at the 25th National Conf. on Artificial Intelligence (AAAI), pp. 63–64, 2011.

URL <http://www.aaai.org/ocs/index.php/WS/AAAIW11/paper/view/3895>

This talk will focus on recent research results that show how a robot can solve multiple tasks based on what it learns during a developmental period similar to a child's play. During this period the robot actively tries to grasp, lift, shake, touch, scratch, tap, push, drop, and crush objects. At the end of this period the robot knows what different objects sound like when they are dropped, feel like when they are squeezed, etc. Because these properties are grounded in the robot's sensorimotor repertoire the robot can autonomously learn, test, and verify its own representations without human intervention. The talk will demonstrate how the robot can use this information to recognize objects, separate objects into functional categories, and even find the odd-one-out in a set of objects. The talk will also demonstrate how the robot can use sensorimotor interactions to bootstrap the development of its visual system in the context of a button-pressing task. Experiments on learning the properties of cup using water will also be presented.

4.27 Unsupervised Discovery of Actions and Action Possibilities

Emre Ugur (ATR – Kyoto, JP)

License © Creative Commons BY 3.0 Unported license
© Emre Ugur

Joint work of Ugur, Emre; Oztop, Erhan; Sahin Erol

Main reference E. Ugur, E. Oztop, E. Sahin, “Goal emulation and planning in perceptual space using learned affordances,” *Robotics and Autonomous Systems*, 59:7–8, pp. 580–595, 2011.

URL <http://dx.doi.org/10.1016/j.robot.2011.04.005>

In our framework that is inspired from Developmental Psychology, the robot’s discovery of action possibilities is realized in two sequential phases. In the first phase, the robot that initially possesses a basic action and reflex discovers new behavior primitives by exercising the action and by monitoring the changes created in its initially crude perception system. In the second phase, the robot explores a more complicated environment by executing the discovered behavior primitives and using more advanced perception to learn further action possibilities, aka affordances. After learning affordances through self-interaction and self-observation, the robot can make plans to achieve desired goals, emulate end states of demonstrated actions, monitor the plan execution and take corrective actions using the perceptual structures employed or discovered during learning.

This research was partially supported by a contract with the Ministry of Internal Affairs and Communications entitled, ‘Novel and innovative R&D making use of brain structures’.

4.28 Do We Need Models to Develop Robot Vision?

Markus Vincze (TU Wien, AT)

License © Creative Commons BY 3.0 Unported license
© Markus Vincze

Joint work of Wohlkinger, Walter; Aldoma, Aitor; Rusu, Radu Bogdan; Tombari, Federico; di Stefano, Luigi

Main reference W. Wohlkinger, A. Aldoma, R.B. Rusu, M. Vincze, “3DNet: Large-Scale Object Class Recognition from CAD Models,” in *Proc. of the 2012 IEEE Int’l Conf. on Robotics and Automation (ICRA’12)*, pp. 5384–5391, IEEE, 2012.

URL <http://dx.doi.org/10.1109/ICRA.2012.6225116>

Main reference A. Aldoma, F. Tombari, L. di Stefano, M. Vincze, “A Global Hypotheses Verification Method for 3D Object Recognition,” in *Proc. of the 12th European Conf. on Computer Vision (ECCV’12)*, LNCS, Vol. 7574, pp. 511–524, Springer, 2012.

URL http://dx.doi.org/10.1007/978-3-642-33712-3_37

The robots we wish to build work in a human environment. It is know (e.g., studies by M. Land) that humans strongly use models to cope with the complexity of their environment. Hence, it is argued that models play a strong role in vision. Consequently, the work presents attempts to learn models of objects and the environment, understand when models are complete, and then use models to detect target items given a robotics task. It can be shown that the use of models is highly beneficial to improve the robustness of object and object class detection as well as scene segmentation and object tracking.

4.29 Piaget for Robots: Implementing Accommodation and Assimilation in a Machine

Florentin Wörgötter (Universität Göttingen, DE)

License  Creative Commons BY 3.0 Unported license
© Florentin Wörgötter

Using generative models it is possible to implement the two piagetian mechanisms of Accommodation and Assimilation in a robot. By viewing a human the machine extracts the observed action and compares it to its (earlier acquired) action knowledge. If the action to entirely new it stores it as a whole (accommodation), if it is similar to a known action only the novel sub-aspects are memorized (assimilation). This is possible because of a new type of action representation – the Semantic Event Chain (SEC) – by which the “essence” of an action is extracted. This allows the machine to distinguish between known and unknown actions. Altogether this leads to a substantial speed-up of learning and supplements other learning mechanisms (e.g. learning by exploration) in an efficient way.

5 Working Groups

Working groups met on the Tuesday.

5.1 Group on transfer of means/skills

(Konidaris, Bushnell, Liu, Alexander, Ugur)

We were mostly in agreement that our focus should be on policy transfer, where an agent learns a policy in one task that can be redeployed in other tasks. We felt the most interesting directions involved learning policy libraries and then using the agent’s context to determine which might be applicable to new tasks. Challenges here included determining which of the many features available to the agent could be useful for this prediction, whether we could infer that features used to predict the usefulness of some actions might also be likely to predict the usefulness of others, how to include structured bias into this process (in the form of both motion primitives and prior knowledge about the way the world works) and how to avoid negative transfer. This discussion was partly motivated by Prof. Bushnell’s example of babies both under- and over-generalizing learned skills.

5.2 Group on motor skills/manipulation

(J. Lockman, B. Kuipers, A. Sloman, R. Grupen, J. Fagard, E. Ugur, L. Rat-Fischer)

Interesting time points for infant development

1. Fetal period – birth
2. Reaching – grasping (5-7 mo)
3. Manipulation (6-12 months)
4. Means-end, multiple-step actions (7-24 months) (No time to deal with this one)

5.2.1 Fetal period

Infant development: To know motor primitives, it is important to start from long before birth: fetuses start moving at 8 weeks, then show general movement, then isolated movements at 10 weeks, then hand-head touching a week later, then soon thumb sucking, and finally thumb sucking with anticipation of mouth opening, all develops within a few weeks. Thus sensorimotor contingencies detection starts several months before birth and we should try to understand the very beginning. (why thumb sucking in particular? The tip of the thumb and the mouth have a particularly high density of sensors; prepare the infant for being breastfed; the space is limited in utero; target toward body always easier than target toward the outside; etc.)

Robotic: Few fetuses studies beside Kuniyoshi's simulation.

Propose more studies with less priors in the robot?

5.2.2 Birth

Infant development: Huge change in the environment: from liquid to aerial environment (in addition to better vision of the hand and of the environment). Does the learning in utero help the neonate scaffold his motor repertoire or does he have to relearn from scratch? Would it be more difficult to learn the 'general landscape' of this gesture without having experienced this in utero?

If the effect of the environment is minimized, as in the condition of freed motor control ("motricite liberee", Grenier), with the neck supported by the adult, then the neonate is capable of pre-reaching to a bright object.

Robotic: No robot was ever programmed to go from liquid to aerial environment! One idea: have robots go from kinematic position to dynamic. There has been work that exploit kinematic-then-dynamic strategies (Rosenstein's weight lifter robot, for example), and policy iteration techniques that start from kinematic seeds (Schaal, Kuindersma).

In general, infants spend a lot of time discovering the dynamics of their body. There are numerical techniques that automatically identify the dynamics of a limb or of the limb and an object that is grasped. These techniques in robotics can acquire the forward dynamics of the limb up to a set of equivalent parameter settings and with more training get better. Moreover, having identified the dynamics of the limb, these algorithms can also identify the inertial parameters of grasped objects while undergoing generic movements. These results have not yet been generalized to "whole-body" dynamics.

5.2.3 5–7 months: emergence of grasping

Infant development: In order to reach and grasp objects, infants need to know how to compensate gravity, how to slow down their movement before touching the object, anticipate the shape, size, etc. of the object so that the hand arrives around it, ready to grasp. He does it by freezing the degrees of freedom of the arm.

There are several studies focused on corrective movements instead of integrated movements (Berthier), and reach-touch, reach-grasp skill learning (Gruppen). However, these studies do not match the human infant's broad attention to motor contingencies. One of the appropriate foci for study involves frameworks for attention and exploration in robots aimed specifically in competency and situation assessment in unstructured situations

Robotic: How can robotic help solve understand the mechanisms underlying changes? For instance robotics has solved the forward-inverse problem but we still don't understand how infants solve the problem.

Several studies have explored “freezing and thawing” degrees of freedom with inconclusive results. For example, Luc Berthouze has analyzed jolly jumpers and swinging motions using robots that first engage hip motions and then knees. The simple proximal degree of freedom followed by proximal plus distal degree of freedom has not yet proven to lead to optimal 2 DOF strategies. The proximal-distal proposition has, therefore, not yet been demonstrated conclusively.

5.2.4 6–12 months: emergence of manipulation

Infant development: as soon as the infant is able to grasp objects, he manipulates them. A lot of mouthing but also many actions adapted to the object’s characteristics (banging hard object on hard surface for instance, etc. Ruff, Lockman). They often transfer the object from hand to hand, use role-differentiated bimanual actions.

Robotic: Need for bimanual coordination? (is it difficult for a robot to transfer object from hand to hand? This is a very common behaviour in infants starting to explore objects)

Tasks involving two hands can easily be decomposed into left and right hand roles in robot control systems, however, to optimize the behaviour of the bimanual system, a subsequent optimization period is required. Several analytical approaches are applicable (dynamic motion primitives, policy iteration techniques), but I am not aware that this has been demonstrated in bimanual tasks in a compelling fashion.

Conclusion: should we treat robots as a different species?! Humans develop, grow physically whereas robots do not ? does it change the learning processes? Infants & robot learn in an opposite way: – infants start with a bunch of explorations, – robots start with “what is my task?”

5.2.5 Comments from Rod Grupen

In general, skills and abilities in infants and robots are still acquired in quite different way. Infants build layer upon layer of support skills by exploration that seems to be independent of any other purpose than to acquire comprehensive mastery of increasingly sophisticated relationships to the world. No task is required. The state of the art in robotics, however, typically starts with a target task and is reduced into pieces that are described algorithmically. Typically, a designer anticipates all the events and intermediate states and therefore, the robot is unsupported by the same breadth of contingencies that the infant spends all of its time constructing during the sensorimotor stage of development. This is an opportunity for both fields. If the artifacts of development in human infants can be transformed into theories of the processes of development, then these can be verified on robot platforms to create better robots as well as to codify theories of development in animals.

5.2.6 Comments from Aaron Sloman

A general point is that I don’t believe we have an adequate ontology for formulating theories about the information processing going on at different stages in an infant, nor the changes that can occur (possibly along different trajectories in different individuals, including changes in architectures, forms of representation, ontologies, data-structures, information actually stored, transformations of information, control strategies – e.g. selecting capabilities that exist but may or may not be used at various stages).

Compare the differences between the Aristotelian attempt to explain behaviours of physical matter (in terms of something like the goals of different kinds of matter and what

they do to achieve their goals), and Newton's explanations in terms of inertial mass, velocities, accelerations and forces.

The changes we need are probably much more complex than the transition from Aristotle to Newton.

Jackie Chappell and I wrote an invited paper for "The International Journal of Unconventional Computing" trying to bring out the different relations between genome and development at different stages of development, suggesting that some of the genomic influences can only operate on results of previous learning that are environment-dependent. This undermines many evo-devo debates. The paper is online here:

<http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0609>

One aspect that changes during development in ways that current theories in AI and psychology do not seem to address is the role of mechanisms for representation of possibilities (in the world, in the child) and the control of use of those mechanisms. The developing understanding of sets of possibilities and constraints on those possibilities appears to be the basis of at least some mathematical competences, including those that led our ancestors to the discovery/creation of Euclidean geometry.

I also wonder how many differences there are between infants in different cultures and different physical environments (e.g. cave dwellers, tent dwelling nomads, infants with and without planar surfaces in their environment...

5.3 Group on concepts/representations

(O'Regan, Woergoertter, Stoychev, Ugur, Stojanov)

The discussion in this group was on a much more general level, which was probably to be expected given the breadth of the topic. Just to give an idea: the discussion went from philosophical theories of concepts (feature, rule, prototype, or example based) to the conceptual framework of children raised in non-typical environments (raised by animals, in confinement, and so on). We started by criticizing current mainstream research in developmental psychology, agreeing that if Piaget were to submit a paper to Jean Piaget Society annual conference, it would probably be rejected.

During the discussion two opposing views emerged:

1. Concepts emerge gradually, starting with the motor babbling, which gives rise to repetitive sensory-motor experience, especially before the time infants are capable of walking. These similar sensory-motor trajectories are then clustered producing a primary categorization tool for objects that can be physically manipulated. The process continues with the development of language, when these primary categories start being labeled. The syntax of the language then allows for hypothetical (never-seen) word constructs, novel categories to be expressed. Perception of objects/situations, in this view, would involve bottom up activation of these clusters and spread activation to close ones.
2. The opposing view, articulated by a practicing robotics researcher, was that this cannot be the whole story, for the way it has been presented, the previous theory could not give a satisfactory account of the emergence of abstract concepts like 'containment'. Also, perception of an object as a member of some category happens virtually instantly in human perception. But, so far, we do not know of such fast algorithms that will start from the pixels and come out with the object category name instantly. A suggestion was made that we probably need some grammar-like structures, perhaps innate (the name of Jerry Fodor was mentioned), to account for the 'immediacy of perception'.

We certainly did not come to a point to suggest some experimental designs for psychologists, but did discuss possibilities of exploring perception in congenitally blind people, or people born with other types of sensory-motor deficiencies. We noted that none of us were aware of psychological research that would suggest what kinds of data structures/algorithms were better suited for modeling concepts in machines. Currently, one might say that the bottom-up ('clustering') approach is more popular at least among researchers in developmental robotics.

5.4 Group on motivation (e.g. what to explore, and what is not interesting)

Outcomes of the meeting of the group.

- Focus on intrinsic motivations (IMs)
- Focus on IMs in infants, but also a bit older children and adults
- Focus on psychological problems (we considered robots as models to study children, although being aware that this will be also useful for technology in the future)

Why is studying IMs important?

- They are fundamental for individual learning, when there are no EMs or social pressures. E.g., consider a child playing. A large amount of knowledge and skills are acquired based IMs (we played a game: formulating the subjective estimation of the percent of knowledge and skills acquired by a 1 year old child during 1 month based on IMs. The results was: Verena, Gianluca, Beata, 90)
- IMs are fundamental for education, but the basic mechanisms and principles behind them are not well understood: if they will be better understood we could improve the education system
- Because they are implicitly exploited in most developmental psychology experiments (where are take for granted), so if you understand them you can control such experiments better
- ...but IMs are not studied much per se: they should, given their importance.

Why are IMs not studied much in developmental psychology?

- Because the focus of whole "scientific" psychology is on cognition rather than on motivations/emotions (treated in psychology only for therapy, etc.).
- Because developmental psychology rarely studies mechanisms; e.g., it very often studies when different cognitive capacities emerge (50-70)
- So, importance of:
 - collaboration with modelers for developing experiments on mechanisms
 - finding new experimental paradigms suitable to study IMs

Methodological problems and solutions:

- Problem: How to study IMs for ongoing development in the limited time of experiments? Solution: Longitudinal studies can also help a lot to do this Solutions: developmental experiments are always exploiting IMs, so:
 - We can look at existing research to have info
 - They show we can study IMs mechanisms, e.g. creating suitable set ups (novel objects, agency).

- Other problem: adults, but also children (and even babies!) feel a lot of social pressure and implicit requests for tasks.
- Other problem: it is in general difficult to study motivation in the lab as you cannot control motivation.

IMs and other motivations:

- Very often you have multiple motivations:
 - multiple IMs
 - multiple EMs
 - multiple social motivations and they together drive behaviour
- One interesting problem is how these different sources of motivations are arbitrated

Relations between IMs and social motivations:

- Problem of the relation between IMs and social motivations, for example imitation.
- Imitation might be an innate drive, so not related to IMs.
- Or imitation might be (at least in part) the consequence of IMs as they drive the child to engage in experiences that maximise learning rate.

5.5 Visual Perception

What is missing in robotics as opposed to children?

- Segmentation: developmental studies assume objects as entities, however, this is one of the big open visual perception questions.
- Perception of gestalts: It is unclear how to group percepts and how to weigh different Gestalt principles.
- Stable object and object class recognition, similarly for navigation and localisation
- Small parts and small objects: difficult to perceive (Kinect is blind to small objects) and difficult to find reliably.
- Object permanence: knowing this is the backside of the object seen before.
- Object models: taking into account changes in objects such as cutting them or other non-trivial deformations.
- Hardware is missing: hands, tactile sensors, robustness of HW is lacking, 6-7DOF arms suitable for mobile robots, costs are far too high.

5.6 Miscellaneous points spanning above subareas

Rod Grupen made the point that some problems might be too difficult in a single modality. The group in general identified a lack of psychology research on mechanisms of development (most research is about abilities at timepoints). What new experimental paradigms could be proposed to address this deficiency?

6 Kevin's Game

This was a game proposed by Kevin O'Regan.

6.1 Part A

The Rules:

1. The psychologists pose a concrete question about a particular behaviour
2. Roboticist volunteers give 6 minute explanations suggesting why their particular theoretical constructs can explain the behaviour
3. The psychologists comment and evaluate on the responses

6.1.1 Question from Psychologist Emily Bushnell

There is a library of elementary skills. How does this battery of skills increase in number, and how do they recombine and get modified over developmental time?

6.1.2 Ben Kuipers

Using QLAP, and starting from raw sensory data, the agent builds a hierarchy of actions aimed towards making the combinations coherent for a given task. One problem is how to find meaningful qualitative states from the continuous data. Seek "contexts" in which observed contingencies between events become more reliable To attain "goals", find conditions where appropriate contexts exist that allow wanted actions to occur Use reinforcement learning to search the space of qualitative values that correspond to those contexts. Method allows simple skills to develop into higher level skills. The search space is reduced by this progressive method. "NEVER SOLVE HARD PROBLEMS". Solve easy problems first.

Comments from Psychologists: An interesting approach.

6.1.3 Gianluca Baldassarre

There are three parts of the model: (i) a skill learning part: makes a map between state of the world and motor behaviour; (ii) Goal creating part; (iii) Motivation part: (intrinsic, extrinsic, social).

Goals can be set either through motivation or through a change in the Environment.

Comments from Psychologists: The most translatable into psychology among the robotic competitors.

6.1.4 Rod Grupen

Build General Motor schemes which correspond to major "routes" to solve problems, fit them to smaller problems.

Comments from Psychologists: Good metaphors, good timing; good tripping, sometimes lost en route though.

6.1.5 Florentin Wörgötter

List the things you can do with your hand; three types: grasp, Take down, put on top. Describe the essential characteristics of the different things you can do. Learning problem: two components: 1. Acquire the "essence"; 2. Find out how to do it nicely.

Comments from Psychologists: Very Good list.

6.1.6 Alex Stoytchev

No answers, only analogies... Skills are “replicators” which evolve. Where did the skills come from? How do they evolve? Is imprinting fundamentally different from the mechanisms that make skills evolve?

Comments from Psychologists: Extra credit for being brief. Good to be clumsy at the beginning.

6.1.7 Aaron Sloman

Vive l’architecture!

Comments from Psychologists: Vive l’architecture!

Some robotics/AI researchers have noticed that human and animal competences have different “layers”, providing very different capabilities.

Reactive architectures can provide rich and versatile behavioural competences, with little understanding of how they work what their limitations are, what might have gone wrong in a successful action, Insects and perhaps the majority of successful behaving organisms have only that kind of intelligence.

There’s another architectural layer that seems to have evolved much later and in fewer species which involves not just the ability to perform successfully, but the ability to consider unrealised possibilities, to speculate about unknown structures and processes (e.g. in the distance, out of sight, in the past in the future, or what might have happened but did not) and to understand some of the constraints on those possibilities, which can be used in planning, predicting, explaining, designing new machines or buildings or changing actions to improve them, without depending on trial and error.

Yet another collection of competences which seems to be even rarer and probably develops more slowly in humans involves meta-semantic competences: being able to represent and reason about things that represent and reason (including having goals, plans, beliefs, preferences, puzzles, etc.)

I conjecture that in children the three types of architecture develop in parallel, with increasing roles for the second and third layers over time, and that the patterns of interaction are so diverse (across individuals, and across cultures) that human developmental trajectories are far more variable than in other species., a fact that can be missed by some researchers. For more on this see

<http://www.cs.bham.ac.uk/research/projects/cogaff/#overview>

To be continued.

6.2 Part B

The Rules:

Roboticists ask psychologists what precise experiments they would like psychologists to do.

6.2.1 Ben Kuipers:

Analogously to Hilbert’s list of important problems in mathematics, create a small and concrete set of puzzles that the roboticists could model, and that psychologists think would be productive. Find ways to visualize individual results with their intrinsic variation.

6.2.2 Markus Vincze & Justus Piater

Create experiments that could separate maturational learning from (adult-type) learning. Determining that something happens at Month X does not mean anything useful to roboticists. It is better to specify the necessary capacities that precede each next developing capacity. Individual trajectories are useful because they help specify sequences.

6.2.3 Gianluca Baldassarre

How do children set their goals when there is no social pressure?

6.2.4 Frank Guerin

More longitudinal observations of individual infants' day by day evolution (e.g. as Ester Thelen did analysing reaching). Make detailed video databases of individual evolution and make them available.

6.2.5 Pierre-Yves Oudeyer

Experiments to show whether development of language vocalization use the same mechanisms as skill learning?

6.2.6 Alex Stoytchev

It would be desirable if psychologists could publish unaveraged results, to show what each individual child does, and their individual trajectory of development. Averaging loses a lot of important data.

7 Challenge Problems

Ben Kuipers initiated a discussion about “challenge problems” along the lines of Hilbert’s problems. This came to be known as “Ben’s Ten” within the group.

The following is the proposal for Ben’s Ten concrete psychological problems to be solved by roboticists:

1. To model the Rovee Collier experiments showing the effect of a change of context on an infant skill. Can the reinforcement learning framework explain this progressive specialisation, and effect of context, perhaps through generalization and transfer of the RL skills. (Testing basic reinforcement-type learning on infant phenomena.)
2. To model the infant’s intrinsic motivation and the developmental progression in their spontaneous exploration of objects of different complexity (moderate discrepancy theory) (cf. Emily Bushnell). Perhaps modelled on Pierre-Yves Oudeyer’s experiment to test intrinsic motivation
3. To model the process of symbol generation/abstraction/object concept formation, e.g. Sinan Kalkan’s noun/adjective distinction; chunking.
4. To model child abilities in sequential attention/planning/sequential behaviour (perhaps modelled with hierarchical reinforcement learning policies).
6. To model convincingly the A not B task.
7. To model infant abilities in Combining elements which are related (spontaneously sorting objects).

8. To model the following phenomena described by Piaget: the child's difficulty with problems like: rose is a flower and a flower is a plant.
9. To model the problem described by Jacqueline Fagard of lifting a rod from slot, where the rod has board stuck on top; infants struggle to do this, why? It seems to be related to retrieving an object from on top of a support, where the object loses one boundary.
10. Pavlovian conditioning has 47 phenomena which have not been explained mechanistically (cf. Stoytchev) (cf behaviourism: discriminative learning; generalization.).

Aaron Sloman had given the example of Richard Young's modeling with production system of Piaget's serial sorting task, as a particularly clear example of how AI techniques could model some psychological phenomena.

Participants

- John Alexander
University of Aberdeen, GB
- Gianluca Baldassare
ISTC-CNR – Rome, IT
- Emily W. Bushnell
Tufts University, US
- Paul R. Cohen
Univ. of Arizona – Tucson, US
- Rana Esseily
Univ. Paris Ouest Nanterre, FR
- Jacqueline Fagard
Université Paris Descartes, FR
- Severin Fichtl
University of Aberdeen, GB
- Roderic A. Grupen
University of Massachusetts –
Amherst, US
- Beata Joanna Grzyb
Universitat Jaume I – Castellon
de la Plana, ES
- Frank Guerin
University of Aberdeen, GB
- Verena V. Hafner
HU Berlin, DE
- Matej Hoffmann
Universität Zürich, CH
- Bipin Indurkha
AGH University of Science &
Technology – Krakow
- Sinan Kalkan
Middle East Technical University
– Ankara, TR
- George Konidaris
MIT, US
- Norbert Krüger
University of Southern Denmark –
Odense, DK
- Benjamin Kuipers
University of Michigan, US
- Ales Leonardis
University of Birmingham, GB
- Honghai Liu
University of Portsmouth, GB
- Jeffrey J. Lockman
Tulane University, US
- Bärbel Mertsching
Universität Paderborn, DE
- J. Kevin O'Regan
Université Paris Descartes, FR
- Mohamed Oubbati
Universität Ulm, DE
- Pierre-Yves Oudeyer
INRIA – Bordeaux, FR
- Justus Piater
Universität Innsbruck, AT
- Lauriane Rat-Fischer
Université Paris Descartes, FR
- Helge Ritter
Universität Bielefeld, DE
- Aaron Sloman
University of Birmingham, GB
- Georgi Stojanov
The American University of
Paris, FR
- Alexander Stoytchev
Iowa State University, US
- Emre Ugur
ATR – Kyoto, JP
- Markus Vincze
TU Wien, AT
- Florentin Wörgötter
Universität Göttingen, DE



Consistency in Distributed Systems

Edited by

Bettina Kemme¹, Ganesan Ramalingam², André Schiper³,
Marc Shapiro⁴, and Kapil Vaswani⁵

- 1 McGill University – Montreal, CA, kemme@cs.mcgill.ca
- 2 Microsoft Research India – Bangalore, IN, grama@microsoft.com
- 3 EPFL – Lausanne, CH, Andre.Schiper@epfl.ch
- 4 INRIA & LIP6 – Paris, FR, Marc.Shapiro@acm.org
- 5 Microsoft Research India – Bangalore, IN, kapilv@microsoft.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13081 “Consistency in Distributed Systems.”

Seminar 18.–22. February, 2013 – www.dagstuhl.de/13081

1998 ACM Subject Classification C.1.4 Parallel Architectures – Distributed architectures, C.2.4 Distributed Systems – Distributed databases, D.4.2 Storage Management – Distributed memories, E.1 Data Structures – Distributed data structures, F.1.2 Modes of Computation – Parallelism and concurrency, H.2.4 Database Management Systems – Distributed databases

Keywords and phrases Replication, Consistency, Strong Consistency, Weak Consistency, Distributed Systems, Distributed Algorithms

Digital Object Identifier 10.4230/DagRep.3.2.92

1 Executive Summary

Bettina Kemme

Ganesan Ramalingam

André Schiper

Marc Shapiro

License  Creative Commons BY 3.0 Unported license
© Bettina Kemme, Ganesan Ramalingam, André Schiper, and Marc Shapiro

In distributed systems, there exists a fundamental trade-off between data consistency, availability, and the ability to tolerate failures. This trade-off has significant implications on the design of the entire distributed computing infrastructure such as storage systems, compilers and runtimes, application development frameworks and programming languages. Unfortunately, it also has significant, and poorly understood, implications for the designers and developers of end applications. As distributed computing become mainstream, we need to enable programmers who are not experts to build and understand distributed applications.

A seminar on “Consistency in Distributed Systems” was held from 18th to 22nd, February, 2013 at Dagstuhl. This seminar brought together researchers and practitioners in the areas of distributed systems, programming languages, databases and concurrent programming, to make progress towards the abovementioned goal. Specifically, the aim was to understand lessons learnt in building scalable and correct distributed systems, the design patterns that have emerged, and explore opportunities for distilling these into programming methodologies,



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Consistency in Distributed Systems, *Dagstuhl Reports*, Vol. 3, Issue 2, pp. 92–126

Editors: Bettina Kemme, Ganesan Ramalingam, André Schiper, Marc Shapiro, and Kapil Vaswani



DAGSTUHL
REPORTS

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

programming tools, and languages to help make distributed computing easier and more accessible.

We may classify current approaches to deal with the challenges of building distributed applications into the following three categories:

- **Strong Consistency and Transactions:** Strong consistency means that shared state behaves like on a centralised system, and programs (and users) cannot observe any anomalies caused by concurrent execution, distribution, or failures. From a correctness perspective, this is a most desirable property. For instance, a database management system protects the integrity of shared state with transactions, which provide the so-called ACID guarantees: atomicity (all-or-nothing), consistency (no transaction in isolation violates database integrity), isolation (intermediate states of a transaction cannot be observed by another one), and durability (a transaction's effects are visible to all later ones).
- **Weak Consistency:** Unfortunately strong consistency severely impacts performance and availability [1, 2]. As applications executing in the cloud serve larger workloads, providing the abstraction of a single shared state becomes increasingly difficult. Scaling requires idioms such as replication and partitioning, for which strongly-consistent protocols such as 2-Phase Commit are expensive and hard to scale. Thus, contemporary cloud-based storage systems, such as Amazon's Dynamo or Windows Azure Tables, provide only provide weak forms of consistency (such as eventual consistency) across replicas or partitions. Weakly consistent systems permit *anomalous* reads, which complicates reasoning about correctness. For example, application designers must now ascertain if the application can tolerate stale reads and/or delayed updates. More parallelism allows better performance at lower cost, but at the cost of high complexity for the application programmer.
- **Principled Approaches to Consistency:** A number of approaches and tools have been developed for reasoning about concurrently-accessed shared mutable data. The concept of linearizability [3] has become the central correctness notion for concurrent data structures and libraries. This has led to significant advances in verification, testing and debugging methodologies and tools. Transactional memory provides a higher-level, less error-prone programming paradigm [4].
- **Principles for weak consistency:** More recently, a number of principles have emerged for dealing with weak consistency. For example, if all operations in a program are monotonic, strong correctness guarantees can be provided without the use of expensive global synchronization. Similarly, certain data structures such as sets and sequences can be replicated in a correct way without synchronisation.

These developments illustrate the benefits of cross-fertilization of ideas between these different communities, focused on the topic of concurrency. We believe that such principled approaches will become increasingly critical to the design of scalable and correct distributed applications. The time is ripe for the development of new ideas by cross-fertilisation between the different research communities.

Goals

It is crucial for researchers from different communities working in this same space to meet and share ideas about what they believe are the right approaches to address these issues. The questions posed for the seminar include:

- Application writers are constantly having to make trade-offs between consistency and scalability. What kinds of tools and methodologies can we provide to help this decision? How does one understand the implications of a design choice?
- Weakly consistent systems are hard to design, test and debug. Do existing testing and debugging tools suffice for identifying and isolating bugs due to weak consistency?
- Can we formalize commonly desired (generic) correctness (or performance) properties?
- Can we build verification or testing tools to check that systems have these desired correctness properties?
- How do applications achieve the required properties, while ensuring adequate performance, in practice? What design patterns and idioms work well?
- To what degree can these properties be guaranteed by the platform (programming language, libraries, and runtime system)? What are the performance tradeoffs (when one moves the responsibility for correctness between the platform and application)?

In order to ensure a common understanding between the different research communities that the workshop brings together, the seminar started with a few tutorials from the perspective of each community. Other presentations presented a specific piece of research or a research question. Participants brain-stormed on a specific issue during each of the two break-out sessions.

This report

This report is the compilation of notes taken by several note-takers, rotating at each session. The majority of the participants served as scribe for some session.

It will be helpful to refer to the abstracts and slides of the different presentations, which are available at <http://www.dagstuhl.de/mat/index.en.phtml?13081>.

References

- 1 Daniel J. Abadi. Consistency tradeoffs in modern distributed database system design. *Computer*, 45(2):37–42, February 2012.
- 2 Eric Brewer. CAP twelve years later: How the “rules” have changed. *IEEE Computer*, 45(2):23–29, February 2012.
- 3 Maurice Herlihy and Jeanette Wing. Linearizability: a correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems*, 12(3):463–492, July 1990.
- 4 Maurice Herlihy and J. Eliot B. Moss. Transactional memory: Architectural support for lock-free data structures. In *Int. Conf. on Comp. Arch. (ISCA)*, pages 289–300, San Diego CA, USA, May 1993.

2 Table of Contents

Executive Summary

Bettina Kemme, Ganesan Ramalingam, André Schiper, and Marc Shapiro 92

Tutorials

Tutorial on geo-replication in data-centre applications: Marcos Aguilera
Carla Ferreira and Rodrigo Rodrigues 97

Cloud Storage Consistency Explained Through Baseball: Tutorial by Doug Terry
Kapil Vaswani and Alan Fekete 98

Database consistency tutorial: Alan Fekete
Carlos Baquero and Nicholas Rutherford 99

Technical presentations

Presentation of the workshop: Marc Shapiro
Carla Ferreira and Rodrigo Rodrigues 100

Making geo-replication fast as possible, consistent when necessary: Rodrigo Rodrigues
Kapil Vaswani and Alan Fekete 100

Cloud Types and Revision Consistency: Sebastian Burckhardt
Carlos Baquero and Nicholas Rutherford 102

Semantics of Eventually Consistent Systems: Alexey Gotsman
Petr Kustnetzov and Kaushik Rajan 102

Quantifying inconsistency: the transactional way: Bettina Kemme
Petr Kustnetzov and Kaushik Rajan 103

The Emperor's new consistency – the case against weak consistency in data centres:
Marcos Aguilera
Alexey Gotsman and Jennifer Welch 104

HAT, not CAP: Highly available transactions: Alan Fekete
Alexey Gotsman and Jennifer Welch 106

Scalable transactional consistency for your cloud data: David Lomet
Alexey Gotsman and Jennifer Welch 107

Principles for Strong Eventual Consistency: Marc Shapiro
Vivien Quéma and Amr el Abbadi 108

Composing Lattices and CRDTs: Carlos Baquero
Vivien Quéma and Amr el Abbadi 108

Swiftcloud: geo-replication right to the edge: Nuno Preguiça
Doug Terry and Sebastian Burckhardt 109

Applications for geo-replicated systems — Where are the limits?: Annette Bieniusa
Doug Terry and Sebastian Burckhardt 110


Specifying, Reasoning About, Optimizing, and Implementing Atomic Data Services
for Distributed Systems: Alexander A. Shvartsman
Doug Terry and Sebastian Burckhardt 111

Snapshot Isolation with Eventual Consistency: Douglas B. Terry <i>Ioannis Nikolakopoulos and Ricardo Jiménez-Peris</i>	112
ChainReaction: a Causal+ Consistent Datastore based on Chain Replication: Luís Rodrigues <i>Ioannis Nikolakopoulos and Ricardo Jiménez-Peris</i>	113
Consistently Spanning the Globe for Fault-tolerance: Amr el-Abbadi <i>Ioannis Nikolakopoulos and Ricardo Jiménez-Peris</i>	113
Consistency without consensus and linearizable resilient data types: Kaushik Rajan <i>Maurice Herlihy and Allen Clement</i>	114
ACID and modularity in the cloud: Liuba Shrira <i>Maurice Herlihy and Allen Clement</i>	114
Conditions for Strong Synchronization in Concurrent Data Types: Maged Michael <i>Alex Shvartsman and Luís Rodrigues</i>	115
Commutativity, Inversion, and other Stories of Consistency and Betrayal: Maurice Herlihy <i>Alex Shvartsman and Luís Rodrigues</i>	116
Concurrent Data Representation Synthesis: Mooly Sagiv <i>Mike Dodds and Marcos Aguilera</i>	117
Distributed unification an a basis for transparently managing consistency and replication in distributed systems: Peter van Roy <i>Mike Dodds and Marcos Aguilera</i>	118
Time bounds for shared objects in partially synchronous systems: Jennifer Welch <i>Pierre Sutra and Achour Mostefaoui</i>	118
Reduction theorems for proving serialisability with application to RCU-based synchronisation: Hagit Attiya <i>Pierre Sutra and Achour Mostefaoui</i>	119
Abstractions for Transactional Memory: Noam Rinetzky <i>Annette Bieniusa and Peter van Roy</i>	119
Idempotent transactional workflow: Ganesan Ramalingam <i>Annette Bieniusa and Peter van Roy</i>	120
BubbleStorm: replication, updates and consistency in rendezvous information systems: Alejandro Buchmann and Robert Rehner <i>Annette Bieniusa and Peter van Roy</i>	121
Breakout sessions	
Breakout sessions on distributed applications <i>Pawel Wojciechowski and Noam Rinetzky</i>	122
Breakout Groups and Discussion of Workshop Followup <i>Alex Shvartsman, Lués Rodrigues, Pierre Sutra and Achour Mostefaoui</i>	124
Workshop Followup	
Consensus paper <i>Alex Shvartsman and Luís Rodrigues</i>	125
Participants	126

3 Tutorials

3.1 Tutorial on geo-replication in data-centre applications: Marcos Aguilera

Carla Ferreira and Rodrigo Rodrigues

License  Creative Commons BY 3.0 Unported license

© Carla Ferreira and Rodrigo Rodrigues

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.AguileraMarcos2.Slides.pdf>

The basic scenario is a multi-data center infrastructure supporting a web application or a number crunching computation. Data is usually considered valuable, so geo-replication allows for disaster tolerance. It also provides availability and low latency access. A major challenge is that delays across replicas are significant.

The talk focuses on geo-replication mechanisms, classified along two axes: when replicas are updated (synchronous vs. asynch replication) and what type of service is supported (read/write vs. state machine vs. transactions). We suggest the reader to refer to the classification table, included in his slides.

He gave examples of protocols and systems that implement all combinations of these two features. He started with read/write synchronous replication, and gave a basic quorum construction.

Doug Terry inquired why not mention 2PC at this point. Marcos replied it is more powerful and therefore could be used to implement these simpler abstractions, but he would leave the more powerful constructs to a later point in the talk. He refined the basic construction to obtain the ABD algorithm. Then moved to synchronous state machine replication. The basic abstraction to implement this is consensus and referred to two existing consensus protocols (Paxos and PBFT).

Doug questioned the difference between consensus and 2PC locks. Marcos replied that in consensus you don't have to worry about locks. Consensus is implementing an abstract distributed lock manager. Then moved on to synchronous replication with transaction support. Basic approach is to broadcast all operations to all replicas, locking can be problematic and a separate locking service helps.

A possible technique is deferred updates: execute transactions locally and use total order broadcast to force all replicas to execute all ops in the same order. Another possibility is to use state machine replication to geo-replicate storage servers and 2PC upon commit. (Several people asked for clarifications.)

Moving on to asynch R/W replication, this raises issues of lost updates and of updates arriving at different replicas in different orders. To address the latter, you can have a primary replica that is the only one that accepts updates, you can use merge strategies like last writer wins or an application-specific merge function. He spoke about how to detect concurrent writes using vector timestamps, which raised lots of questions regarding why such detection matters when you are doing blind writes.

Bettina then gave an example where such detection mattered: if one replica adds 2 to x and another replica adds also 2 to x , you want to keep both updates to x . Marcos commented that, because operations are not grouped together (as in transactions), one might get unexpected results.

At this point Marc Shapiro commented that transactions are a proxy for invariants. Marcos stated that these techniques are not good for maintaining invariants, specially for black-box invariants. Then with state machine asynch, there is an advantage in using

commutative operations since different replicas can apply updates in any order.

Regarding asynch transaction replication, commutativity also helps and you arrive to an isolation level called parallel snapshot isolation. He concluded with an overview of examples of several systems.

Alan Fekete pointed out that the talk was about replication but not so much about geo-replication, where replicas are far apart. Marcos concurred and explain a series of challenged with geo-replication. André Schiper asked which techniques are in the table that Marcos presented (see the slides) are useful in the context of geo-replication. Marcos gave an overview of some techniques and said that his advice is to stay in the upper left corner (synch read-write) if possible, and add complexity if performance or semantics require it.

Petr Kuznetsov asked whether new consistency levels are required to meet the needs of geo-replicated applications. Marcos pointed out the need to have a negotiation between performance and invariant preservation. A strategy would be to start from what the application expects in terms of invariants and then select the techniques adequate for those.

3.2 Cloud Storage Consistency Explained Through Baseball: Tutorial by Doug Terry

Kapil Vaswani and Alan Fekete

License © Creative Commons BY 3.0 Unported license
© Kapil Vaswani and Alan Fekete

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.TerryDouglas1.Slides.pptx>

Doug Terry spoke about the different consistency choices that a cloud system could offer to applications. He mentioned several systems. (i) AWS S3 has eventual consistency. (ii) AWS SimpleDB gives a choice between eventual or strong (it started with eventual, and later added the choice by a flag in the API for a given read to be done with strong consistency). (iii) Google AppEngine offers the same choice, but it started from a strong read and then added a choice for eventual consistent read. (iv) Yahoo! PNUTS also offers a choice of eventual or strong consistency on each read. (v) Cassandra has eventual or strong, but the choice must be made system wide by setting the size of R and W parameters. (vi) Windows Azure is only strong, except for a limited period (e.g., 15 minutes) if there is a major system failure.

Many other consistency options have been proposed in research papers, but they not being used in practice. The goal of Doug's talk is to explain usefulness of some intermediate forms. Doug is using a model of operations that are either read or arbitrary writes (these can be append, deposit, etc., not just obliterate the prior value with a new one).

Doug gave six favourite consistency options; he sees these, and combinations of them, as having real use in different situations. (We refer the interested reader to his slides.) In all the consistency options, a read should never see inflight writes, nor should it return a value that was invented out of nowhere. Also conditions apply per item but they could instead be defined on the whole database.

- Strong: the read must see all previous writes.
- Eventual: the read must see the outcome of a subset of previous writes.
- Consistent prefix: the read must see the outcome of an initial sequence of the previous writes.
- Monotonic reads: each read within a session sees a subset of previous writes, and the subset seen increases (or stays equal) from one read to the next.

- Read My Writes (“RMW”): each read sees a subset of previous writes, and the subset must include all the writes issued in the reader’s session before the read itself as issued.
- Bounded staleness: each read sees a subset of previous writes, and the subset must include all the writes that are reasonably old (for example this might be defined as those writes issued up to some time interval before the read was issued, though other definitions are based on the number of writes missed rather than the clock-time).

Prefix, bounded, monotonic, RMW properties are incomparable, measuring a property by the set of allowable return results; each of these is stronger than eventual and each is weaker than strong. There is a tradeoff: the properties differ in performance (mostly latency), and availability, as well as in consistency. Generally the stronger consistency comes with worse performance and lower availability.

Doug worked through some examples, showing how different participants in a baseball game would be able to usefully ask for reads with different consistency. For example, the scorekeeper can do RMW to the score variable, since he is only writer of this location. The umpire needs to read score locations and make binding decisions, so strong consistency is needed.

All six guarantees appear in the examples, sometimes in combination. They would all be happy with strong consistency, but performance trade-offs drive weaker consistency (but it must be not too weak for the application’s need).

Peter van Roy asked whether the system could figure things out, and determine which level is needed? Doug’s answer was probably not, but annotations or other mechanisms might help.

Jennifer Welch asked is there a performance advantage in the combined guarantees, compared to just asking for strong consistency. Doug said Yes.

Allen Clement asked why not talk about the state of the system (rather than what reads return)? Doug said you can only observe state through reads, so properties talk only about reads.

Doug’s conclusions are that replication schemes involve tradeoffs; consistency choices can benefit applications. We as researchers must provide better definitions, analysis and tools.

Doug was asked what is the write consistency model? His answer was that only reads matter.

3.3 Database consistency tutorial: Alan Fekete

Carlos Baquero and Nicholas Rutherford

License © Creative Commons BY 3.0 Unported license

© Carlos Baquero and Nicholas Rutherford

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.FeketeAlan1.Slides.pdf>

Alan Fekete presented a tutorial on consistency from a database perspective, with the aim of helping people from other fields spot differences of perspective and terminology. The talk included the relationship of real-world state-changes to transactions, ACID guarantees, isolation levels, and where inconsistency might originate in a faulty system or program. The comment was made that many databases in the wild run with read-commit isolation, rather than serialisability, and may not be aware of the consequences.

Marc Shapiro asked about the registering of real world events on the database. Alan replied that the database is not limited to the registering of real world events without influencing it, and complemented with an example: When processing a transaction that

registers the paying for a real world exchange, if the transaction aborts due to lack of credit this is reported back to the real world and cancels the exchange that was taking place.

Rodrigo Rodrigues, Marc Shapiro and David Lomet commented on the issue that often there are consistency constraints that are not made explicit in the database. Some constraints are enforced in the program logic that interacts with the database.

Marcos Aguilera asked if there was a reason why the control flow is excluded from the database management system. Alan commented that to some extent this can be declared in stored procedures.

In response to a question by Bettina Kemme, Alan commented that serializability can be extended to include external consistency, leading to linearizability and encompassing session guaranties.

Marc Shapiro questioned whether considering that all operations are effective is equivalent to only considering transactions that have committed, ignoring those that have aborted. Alan replied that one can project the execution to include only the committed transactions.

4 Technical presentations

4.1 Presentation of the workshop: Marc Shapiro

Carla Ferreira and Rodrigo Rodrigues

License © Creative Commons BY 3.0 Unported license
© Carla Ferreira and Rodrigo Rodrigues
Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.SWM.Slides.pdf>

Marc Shapiro presented the workshop and gave some context regarding Schloss Dagstuhl. He proposed the main topic of the workshop could be the tension between strong versus weak consistency. He also called the attention to the fact that different communities use different terminology, focus on different angles of this problem, and sometimes use the same terminology to mean different things. He concluded with some open questions.

Liuba Shrira commented that there is an interesting question that was left out was the evolution of technology, and how it influences the work of these communities. An example was the availability of the notion of global time.

Alan Fekete commented that in cross-community gatherings it is important to pay attention that different communities use different languages, and to make few assumptions about what the audience knows about terminology or background knowledge.

4.2 Making geo-replication fast as possible, consistent when necessary: Rodrigo Rodrigues

Kapil Vaswani and Alan Fekete

License © Creative Commons BY 3.0 Unported license
© Kapil Vaswani and Alan Fekete

Rodrigo Rodrigues spoke next on “Making geo-replication fast as possible, consistent when needed.”

He pointed that delays make users unhappy, and the system achieves less revenue. So one aims to place replicas around world, thus people can find a close one (with low latency) when they want to do something.

Rodrigo pointed to system designs with levels of hierarchy (the higher ones are cheaper to synchronise but have slower latency). Replication at different levels can coexist. The 1st replication level is a central server or master/primary. The 2nd level has remote georeplicated replicas. At level 3: replicas are in CDN infrastructure. The 4th level of replication is P2P or hybrid CDNS (e.g., Akamai NetSession). A 5th level comes with replicas on mobile devices.

Rodrigo identified the challenge: to design distributed systems to be aware of this hierarchy. For example, in Facebook, or PNUTS, writes are done at a single master, and there are read-only mirror replicas.

He mentioned prior work [1, 2]. His systems goal is to balance strong consistency (coming with a total order on operations) with eventual (which is sometimes called causal) consistency, in which there is a partial order on operations. The aim can be summarised as being fast whenever possible, strongly-consistent when necessary.

Rodrigo proposes red-blue consistency, with some operations labelled red and others blue. Blue operations must commute with everything. The partial order is such that all red operations have a total order. The system can be implemented by a token (that circulates from one site to another) and only the token-holder can execute red operations.

Rodrigo worked through a bank application.

A question asked why not synchronise red with all operations (as done in previous systems)? Answer is that one can implement the proposed scheme efficiently, and create tools to decide which operations to make red, Any operation that doesn't universally commute must be red, thus slow.

So Rodrigo suggests that one can redesign the application to split the operation into a generator (which computes what changes are needed) and a shadow (which applies the computed delta). Then we label as red all resulting operations which don't commute universally OR which break invariants. The authors are working on how to automate the split of operations into generator and shadow, and how to label them properly (rather than manually, as so far).

They did an evaluation on real applications, and found that many operations can be blue, and the red ones are invoked rarely. The red-blue approach gets huge latency improvements compared to a multisite strong consistent implementation.


In question time, Rodrigo made the following points: causal consistency is done with version vectors having one entry per datacentre; the want users to indicate invariants that should be maintained. They don't have a proof whether we can always find a shadow that commutes universally, though in examples so far it has generally been easy. If the application adds a new operation, we need to reanalyse everything that had been analysed previously (as well as analysing the new code).

References

- 1 Rivka Ladin, Barbara Liskov, Liuba Shrira, and Sanjay Ghemawat. Providing high availability using lazy replication. *Trans. on Computer Systems*, 10(4):360–391, November 1992.
- 2 Yair Sovran, Russell Power, Marcos K. Aguilera, and Jinyang Li. Transactional storage for geo-replicated systems. In *Symp. on Op. Sys. Principles (SOSP)*, pages 385–400, Cascais, Portugal, October 2011. Assoc. for Computing Machinery.

4.3 Cloud Types and Revision Consistency: Sebastian Burckhardt

Carlos Baquero and Nicholas Rutherford

License  Creative Commons BY 3.0 Unported license
© Carlos Baquero and Nicholas Rutherford

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.BurckhardtSebastian.Slides.pptx>

Sebastian Burckhardt presented “concurrent revisions” for cloud programming, and the TouchDevelop programming platform where it will be given to users to evaluate its effectiveness as a simpler way to address consistency in distributed concurrent programming. The talk began with a summary of concurrent revisions, a fork-join replicated state model for multicore programming which provides parallelism and preserves determinism, with replica conflicts addressed by type-specific merging.

Maurice Herlihy made a comment on possible relations with persistent data structures [1].

Cloud Types were then presented, an application of concurrent revisions to cloud computing applications, with the example of client-side replicas for mobile applications. TouchDevelop was presented as an existing platform for non-expert programmers to develop mobile applications, to which Cloud Types will be added.


Liuba Shrira asked if they considered direct communication between devices. In reply, Sebastian added that communication with the servers is important but that extra connections, between devices, can be also useful as a complement.

References

- 1 Chris Okasaki. *Purely functional data structures*. Cambridge University Press, 1999.

4.4 Semantics of Eventually Consistent Systems: Alexey Gotsman

Petr Kustnetsov and Kaushik Rajan

License  Creative Commons BY 3.0 Unported license
© Petr Kustnetsov and Kaushik Rajan

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.GotsmanAlexey.Slides.pdf>

Paper <http://www.dagstuhl.de/mat/Files/13/13081/13081.GotsmanAlexey.Paper.pdf>

The speaker is primarily interested in verification of concurrent programs, which enables reasoning about the correctness of a program only having the specification of the used library in mind, and not the internals of its implementation. Examples of specification techniques includes strong and composable notions like linearizability, as well as weak and non-composable memory models (x86, C/C++). Given that processors and programming languages do not provide sequential consistency, a multiprocessor machine should in fact be treated as a distributed system.

In distributed systems, however, strong consistency criteria are often replaced with weak ones (to reach A and P in the CAP triad), such as eventual consistency. A popular definition of eventual consistency, given by Werner Vogels “If no new updates are made to the object, eventually all accesses will return the last updated value” [1] does not allow to reason about scenarios in which updates never stop. There is a number of fixes of this for various setting and types of implementations (ruling out anomalies, preserving causality, restrict to conflict-free replicated data types or transactions). But there does not exist a declarative definition of the semantics of eventually consistent systems.

This work proposes a framework for declarative specification of consistency model. The system model considers a replicated service with full replication (every replica stores complete

data), generic operations not limited to read/write, asynchronous replication scheme, link failures, but no replica crashes.

The framework encompasses various existing conflict-resolution techniques, such as:

- Timestamp last writer,
- High level commutativity,
- Return all conflicting values to users,
- Application-dependent resolution,

as well as invariants that should be observed in the presence of ongoing updates, such as:

- Read your own writes
- Causal or FIFO Ordering of operations

The framework addresses the conflict-resolution techniques via *data type specification* and invariants via *consistency axioms*.

Users interact with the system via requests and response events. An execution is modeled as a tuple which capture relations between events (operations, parameters, and returned values, etc.), session order (similar to program order), visibility relation (delivery of updates), and arbitration relation (for conflicting updates).

Now the data type is specified as a function $F : \text{CONTEXT} \mapsto \text{OPERATION OUTPUT}$, where context of an operation a is a projection of events that have been delivered to the replica performing a (actions visible to a).

Now consistency axioms capture the desired semantics of the system, such as "eventuality" (an operation cannot be invisible to infinitely many actions on the same object), or "causality" (all actions that happen before on the same object are visible).

Altogether, this defines semantics of eventually consistent systems, and the paper (available on the seminars' web page) validates the specifications via example abstract implementations.

Marcos Aguilera and Marc Shapiro questioned if the proposed definition of eventual consistency is not too weak actually to be useful.

References

- 1 Werner Vogels. Eventually consistent. *ACM Queue*, 6(6):14–19, October 2008.

4.5 Quantifying inconsistency: the transactional way: Bettina Kemme

Petr Kustnetzov and Kaushik Rajan

License © Creative Commons BY 3.0 Unported license

© Petr Kustnetzov and Kaushik Rajan

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.KemmeBettina.Slides.pdf>

How likely is it that a real execution gives strong consistency even though the system only guarantees weaker consistency? The talk advocates the idea of *quantifying* inconsistency with respect strong consistency (e.g., serializability). One example of such a quantification is to count the number of serialization *cycles* that are observed in executions of the system under a given workload.

The approach looks promising because even if potential inconsistencies are known, it is not clear how often inconsistency actually occur. This may help to answer questions like : when we move to cloud storage with a different consistency guarantee, how consistent my existing application would be?

The quantitative approach boils down to:

1. Deploy an application on a multitier platform

2. Choose a level of isolation provided.
3. Run the application and count serialization anomalies, their types, etc.
4. Be efficient, do not slow down execution

The approach was first applied to a traditional database system. The technique here was based on building a dependency graph with RW, WR, WW conflict edges. To make the search for cycles efficient, we can use the properties of isolation levels, which allows us to exclude some paths. Cycles are visualized, and patterns are identified that might help in rewriting the system.

The second application of the method was cloud storage where transactions are not necessarily supported. The notion of a *work unit* is introduced instead.

It is assumed that local read-write ordering is preserved and that write order is consistent with commit order (primary master with FIFO messaging). However, in this case, some edges (WW, RW) are undetectable. The solution would be to build an approximate graph which says things like either one of two ordering holds. The resulting anomaly classification would thus include notions like *maybe cycles*.

Future work will include object-based anomalies, stale reads, violations of monotonicity. Questions:

- Anomalies are often caused by some system failures: can the framework be used to detect and identify failures – yes it can.
- What is the slowdown it causes: not much, finding cycles in application engine incurs slowdown of only 2–3%
- How the does the effect depend on workloads: need testing framework around.

4.6 The Emperor’s new consistency – the case against weak consistency in data centres: Marcos Aguilera

Alexey Gotsman and Jennifer Welch

License © Creative Commons BY 3.0 Unported license

© Alexey Gotsman and Jennifer Welch

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.AguileraMarcos3.Slides.pdf>

Several data center storage systems adopt weak consistency models, in order to improve performance and availability; plus they are easier to develop. Examples include Yahoo! PNUTS, Amazon S2, Amazon SimpleDP, Microsoft Azure, and Dynamo. Users see stale data, but only sometimes, and the conventional wisdom is that users are tolerant.

Someone asked whether stale data is inherent in all weak consistency systems. Marcos answered that staleness is the most common form of lack of strong consistency, but there could be worse things like garbage. Someone pointed out that stale data may still be consistent, e.g., you get multiple data items corresponding to a consistent snapshot. For instance, serializability can return stale data in reads; the response was to combine serializability with external consistency. Another comment was that PSI [1] returns data that is stale but still pretty good. One can also have eventual consistency + causality + session guarantees, which is pretty good. One needs to qualify the statement that weak consistency returns bad data. A response was that there are degrees of badness.

Marcos argues first that *the drawbacks of weak consistency are expanding* due to technical, legal, and sociological reasons.

Someone raised the point that weak consistency is not well-defined (it is commonly construed as simply meaning not strong consistency), and thus who is to say that there isn’t

a better definition of weak consistency that avoids the drawbacks?

Users typically tolerate strange behavior in free apps. However, increasingly users pay for their apps, and paying users plus strange behavior equals angry users. This is one example of increasing drawback of weak consistency.

The question was asked why equate staleness with strange behavior? The answer gave an example that violates causality. Then it was pointed out that causality can be provided cheaply. Someone pointed out that serializability also gives stale data, but the response was that usually systems try to avoid it.

Another factor is the increasing reliance on data center apps (e.g., banking and medical records). As the application provider is subject to liability, anomalous behavior can lead to monetary loss.

Someone pointed out that the pressure of liability does not lead to systems being built better, it just leads to checklists. The response was that then the burden is on the government to set up the checkboxes.

A third factor is that there are increasing number of layers and more integration among apps. Humans can identify and tolerate inconsistency but programs have a harder time.

Marcos' second argument is that *the benefits of weak consistency are shrinking* for technological reasons.

First, network partitions will disappear, as their causes are being addressed both within and across data centers. Second, wide-area latency is shrinking. Third, the number of applications is growing relative to the number of storage systems, and so the argument that it is easier to build a weakly consistent storage system loses force because it makes it harder to develop apps.

On the other hand, some factors that would undermine his argument were given, including the possibility that people will just become inured to the problems raised by inconsistency.

As a result of this double movement, in the future, drawbacks will outweigh benefits. Weak consistency creates challenges that will become harder to overcome, and solves problems that will be solved differently in the future.

Someone commented that strong consistency is not that expensive to solve, say with Paxos in the data center. Someone expressed disagreement with the claim that partitions will go away and latencies will reduce in the relatively near future. Marc showed a graph of throughput vs. latency within a data center for different protocols, showing that Parallel Snapshot Isolation is much better than serializability. However, some people said that the implementation of serializability was not optimized.

References

- 1 Yair Sovran, Russell Power, Marcos K. Aguilera, and Jinyang Li. Transactional storage for geo-replicated systems. In *Symp. on Op. Sys. Principles (SOSP)*, pages 385–400, Cascais, Portugal, October 2011. Assoc. for Computing Machinery.

4.7 HAT, not CAP: Highly available transactions: Alan Fekete

Alexey Gotsman and Jennifer Welch

License © Creative Commons BY 3.0 Unported license
© Alexey Gotsman and Jennifer Welch

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.FeketeAlan.Slides.pdf>

Alan advocates a model that might be a sweet spot for storage systems for internet scale systems. He is assuming that partitions will *not* disappear. Many early systems (eg., BigTable, PNUTS, S3, Dynamo, MongoDB, Cassandra, Simple DB, Riak) offered scalability and availability but missed functionality expected in traditional DBMS platforms. Wouldn't it be nice to add more consistency, richer operations, and grouping of operations on multiple items? *The focus of this talk is on ways to group operations on multiple items for internet scale storage systems.*

It has been known at least since 1985 [1] that a system cannot provide always-available serializable transactions if the system can partition. What about providing ACID transactions with weaker Isolation? Serializability may be the ideal, but most single-site DBMS already don't ensure serializability; instead, read-committed is the default.

HAT is a useful model for programmers: a transaction is an arbitrary collection of accesses to arbitrary sets of read-write objects; its semantics is as strong as feasible while ensuring availability even when partitioned. Availability is clearly not possible if the client is partitioned away from its data. However, even in the presence of a partition, if a transaction can reach a replica of each item it needs, then the transaction should be able to commit.

The question was asked if this claim is with high probability. The answer is maybe, if you set your timeouts wrong. Alan and his coauthors haven't proved any theorems yet. They think this is what to aim for.

HAT transactions are all-or-nothing (atomic), causally consistent (including read-your-writes, monotonic reads, write-follows-reads), and provide an isolation level similar to read-committed and repeatable-reads. However, a read does not necessarily see the most recently committed change.

The question was asked whether read-committed forces updates to become visible. The answer is no: in read-committed, if a transaction observes an update, then the updating transaction has committed; there is no recency in the definition.

Alan and coauthors define the semantics with an approach inspired by Adya [2], i.e., a graph of operations with different kinds of edges (e.g., write-read, write-write, read-write, happens-before), and restrictions on the sorts of cycles that can occur. Alan sketched an implementation, in which the client buffers its updates. On commit, it propagates them asynchronously as a group, and tracks causal precedence. The point is mainly to show the possibility of an implementation; lots of engineering will be necessary to get decent performance.

— Q: is this is a model where you ship data to the client, run everything at the client, and the storage system is just storage? — A: yes, but this is not to suggest this is how you *should* implement it. It is just to show the possibility that it can be done even in the presence of partitions.

— Q: Aren't you just reinventing group communication in transactions? There the whole problem was merging; is it the same problem here? — A: Alan doesn't know that they will have that problem. Transaction operations are reads and obliterating writes. It may be that lots of applications may also want to provide user-defined merging; that's a separate issue. Certainly, it is very similar to what happened in partition-tolerant group communication

systems, but with the addition of transactions.

— Q: Are you delaying commit until the partition is reconciled? — A: No: the transaction commits, but the other side doesn't learn about it until later. So the transaction on the other side will not see these changes.

— Q: Can a client observe an older value after getting a newer value? — A: read-committed implementations also have a monotonicity property. Definitions typically don't have that property. But they do have per-item monotonicity, because they add causality.

— Q: Since the client must maintain causality meta-data, how to avoid an enormous space overhead? — A: It's an issue we need to study.

References

- 1 Susan B. Davidson, Hector Garcia-Molina, and Dale Skeen. Consistency in a partitioned network: a survey. *ACM Computing Surveys*, 17(3):341–370, September 1985.
- 2 Atul Adya. *Weak Consistency: A Generalized Theory and Optimistic Implementations for Distributed Transactions*. PhD thesis, Mass. Institute of Technology, Cambridge, MA, USA, March 1999. Appears also as MIT Technical Report MIT/LCS/TR-786.

4.8 Scalable transactional consistency for your cloud data: David Lomet

Alexey Gotsman and Jennifer Welch

License © Creative Commons BY 3.0 Unported license
© Alexey Gotsman and Jennifer Welch

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.LometDavid.Slides.pptx>

The dream is to have transactions anywhere with data in the cloud, mobile device, not having to think about the state of the data. An example was given and discussion ensued with the point being that the world is easier when transactions are available.

The economics of going to the cloud are compelling: cheap power, hardware bulk purchase, low land costs, etc. The cloud backbone is the data center. For transactions and inter-operation, start with two-phase commit (2PC), but 2PC almost never crosses administrative domains, but is only in cluster-based systems.

Vendors provide limited offerings. Transactions for data are guaranteed to exist on the same node (e.g., Microsoft, Amazon, Google) but only eventual consistency is guaranteed for multiple-node data. There is very limited support for transactions across the cloud. The problem is that every data source needs to be a 2PC participant.

Think of the data center (DC) s if it were an SMP/cluster, with high bandwidth and a highly reliable network interconnect. This is not the web or a federated system, so the CAP theorem does not necessarily apply, yet transaction support is very limited.

The intuition of the new idea is *we do not do 2PC with disk, which is an atomic page store*. Put transactions on top of atomic record stores. We need to worry about performance. Details of Deuteronomy were given, which separates transactions from data.

The question was asked whether the contract between the TC and the DC should include locking, but the answer is no, the locking is all confined to the TC.

A prototype has been built, tested, and experimented on. Performance graph shows that latency has big impact on performance. They can get millions of operations per second!

— Q: Is the system designed for a specific workload, or is it general? — A: It is OLAP-focused, but nothing precludes using it in a general purpose way; however, one would need

to think harder about performance.

— Q: You have to propagate LSN to the data store; is it transparent to the data store? — A: You have to enforce idempotence. Also use it to manage the log locally.

— Q: What kind of queries do you send down? Predicates? — A: We only send singleton record operations.

— Q: What is an operation? — A: Insert, delete or update.


— Q: Do you use locking? — A: Yes.

— Q: How do you deal with TC failure? — A: Paxos-style replication. — Q: Isn't that putting the burden of 2PC on Paxos? — A: It does not involve DCs.

— Q: Isn't the TC a bottleneck? — A: Write bandwidth is the main problem for TC, as updates must go through TC. In contrast, a large read do not have to go through TC If you “run out of gas”, instantiate another TC.

4.9 Principles for Strong Eventual Consistency: Marc Shapiro

Vivien Quéma and Amr el Abbadi

License  Creative Commons BY 3.0 Unported license

© Vivien Quéma and Amr el Abbadi

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.ShapiroMarc.Slides.pdf>

Marc Shapiro defined strong eventual consistency as “Eventually, correct replicas eventually reach equivalent states (potentially going through different histories)”.


Doug Terry mentioned that this is the case in Bayou provided that replicas re-order updates themselves. Amr El Abbadi noted that Dynamo allows replicas to diverge, but lets users solve conflicts. He asked whether this is a problem or not.

Marc then introduces conflict-free replicated data types (CRDT), monotonic semi-lattices and discussed state-based CRDTs vs. operation-based CRDT. The main idea in CRDTs is to avoid relying on synchronization. Marc then explained what should be done to handle non-commutative operations. The idea is to extend the semantics to give a deterministic solution that guarantees convergence. CRDTs are used in several key-value stores, in geo-replicated systems (e.g., Walter), etc. Several questions were asked during the talk:

- Doug Terry asked whether operations are necessarily idempotent. Marc replied no.
- André Schiper asked why causal delivery is needed. Marc replied that this is to be able to apply a sequential specification. Causal delivery is ensured using vector clocks.
- Marcos Aguilera asked whether everything could be done with CRDTs. Marc replied that, as Rodrigo Rodrigues pointed out in his talk, not everything can be done without synchronization.

4.10 Composing Lattices and CRDTs: Carlos Baquero

Vivien Quéma and Amr el Abbadi

License  Creative Commons BY 3.0 Unported license

© Vivien Quéma and Amr el Abbadi

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.BaqueroCarlos.Slides.pdf>

Carlos Baquero gave a talk titled “Composing Lattices and CRDTs.”

This talk was a continuation of the previous talk. Carlos emphasized the language aspects of CRDTs. The work presented by Carlos relies on the state-based approach and is based on

order theory and the notion of lattices. Carlos presented a type hierarchy comprising the following types: set, poset, lattice, lattice with bottom. Carlos developed various examples, including lexicographic ordering and the antichain.

Carlos then introduced a notion of “inflation” (ensures monotonically-advancing updates) with examples. He also explained how inflations can be sequentially composed. Several questions were asked during the talk:

- Someone asked whether lattice compositions are universal. Carlos replied that this is an open question.
- Peter van Roy asked whether it is possible to track connections between CRDTs using the theory presented by Carlos. Carlos does not know because he did not try. He said that using this theory, it is possible to know what is updated and that, consequently, it is possible to ship adequate parts of the state. This is a first step towards tracking connections between CRDTs.
- Carla Ferreira asked what exactly they can prove on CRDTs. Carlos replied that it is only possible to know that CRDTs are well defined. It is not possible to prove that CRDTs do something useful. But this is an open question: they did not try to do it.

4.11 Swiftcloud: geo-replication right to the edge: Nuno Preguiça

Doug Terry and Sebastian Burckhardt

License © Creative Commons BY 3.0 Unported license
© Doug Terry and Sebastian Burckhardt

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.PreguicaNuno1.Slides.pdf>

The context of this talk is the latency experienced by the client when accessing the closest datacenter. The trend is to increasingly run code inside clients to (1) improve latency, and (2) improve fault tolerance via disconnected operation.

The basic problem is how to support data sharing in web application. To this end, a useful semantics offers (1) Writes that are atomic and mergeable, (2) Reads that support isolation levels, and (3) Transactions.

The diagram on the slides shows the transaction execution on the client. The key design features include: CRDTs (conflict-free replicated data types), Asynchronous Replication, Multi-Version Server, Version-Vectors, and Client-Assisted Failover.

The audience raised several questions:

- *Are the transactions replayed at the data center?* Yes, for the effects.
- *Must we keep the updates at the client until confirmed by the data center?* Yes.
- *How do you implement the isolation levels?* Using the dependency vector.
- *How could you commit in disconnected mode?* Only in asynchronous mode.
- *What is the size of the vector clock?* The number of data centers.
- *How is causal consistency achieved?* By executing at data center only if dependencies are satisfied.

4.12 Applications for geo-replicated systems — Where are the limits?: Annette Bieniusa

Doug Terry and Sebastian Burckhardt

License © Creative Commons BY 3.0 Unported license

© Doug Terry and Sebastian Burckhardt

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.BieniusaAnnette.Slides.pdf>

Cloud storage has evolved; two emerging principles are the use of (1) object-oriented data stores, and (2) eventually consistent key-value stores.

The stated goal is to make distributed programming as “simple” as concurrent programming (audience laughs). To this end, three application examples are studied in detail.

Example 1

The first example is a social network supporting the operations log in/out, post, send, view wall, manage friends, poll, and like.

There was a question from the audience: How do you choose the granularity of the CRDTs (object vs. field)? The answer was: at the object level; this is mostly a performance question, not a semantic one.

Example 2

The second example is a file system. It uses a sequence CRDT for text files (supporting editing) and a simple register CRDT for non-text files. Also, it introduces a special recursive CRDT for directories.

The slide on execution time shows that the system performs very well if there is a scout on the client, but up to 20x slower when running with remote scouts in a nontrivial configuration.

The audience raised several questions at this point:

- *How do you handle “double creation” of a file?* We assume the user is trying to create the same file (other options are possible as well).
- *What’s the overhead of FUSE in your implementation?* This is shown in the performance results (about 33%).
- *Did you actually implement this?* Yes, everything.
- *How do you implement the FS?* Each subdirectory is its own map.
- *Do you ever renew the cache if the client only reads?* Yes, we have a notification system to inform clients of new updates.

Example 3

The third example is web e-commerce, such as a bookstore. This example contained a number of challenges:


- Limited resource (we should not order books that are not available)
- Top-N (we use approximation)
- We need an index of products (we provide this as service)
- The code mixes small and big transactions (we use SQL to execute big ones on server)
- There needs to be offline conflict resolution

The conclusion mentioned several insights: (1) object abstractions are useful, (2) it is important to balance options: too many, and the user is overwhelmed; too few, and some things are impossible to implement, (3) many users had trouble with object creation, object deletion, and with operations involving a large number of objects.

There was an audience question relating to updates: what happens if you subscribe only to a subset of objects for updates, and those objects are updated along with others in a transaction? The answer is that clients still receive all notifications (even for unsubscribed objects), to preserve causal consistency. This was a discussion point as it raised concerns about the scalability of subscription mechanism for receiving updates.

4.13 Specifying, Reasoning About, Optimizing, and Implementing Atomic Data Services for Distributed Systems: Alexander A. Shvartsman

Doug Terry and Sebastian Burckhardt

License  Creative Commons BY 3.0 Unported license
© Doug Terry and Sebastian Burckhardt

Sharing memory in networked systems is nicer than message passing; otherwise we wouldn't be here talking about consistency! This requires replication for availability and fault-tolerance; with replication comes the challenge of consistency.

The easiest notion for users is the one-copy view (e.g., linearizability), but this is expensive. A cheaper guarantee is: a read sees some subset of the previous writes; but this is too hard to use. Between the “slow but correct” approach, e.g., linearizability, vs. “fast but wrong,” i.e., weak consistency, the desirable goal is “fast enough and correct.”

In dynamic systems new challenges arise due to replicas coming, going, and failing. Such systems must be reconfigurable, and various approaches explored the use of state machine replication, consensus, and group communication. The most efficient techniques appear to be ones based on the ABD approach [2], with extensions for dynamic settings. ABD and quorums provide consistency for small, transient changes; and for larger or more permanent changes the quorums are reconfigured.

The DynaStore [1] algorithm deals with dynamicity by building DAGs of reconfiguration possibilities. An ABD-style exploration looks for a DAG sink. If the number of updates is finite, a sink will be found. This is a promising approach that does not use consensus, although it places constraints on dynamicity. Rambo [4] revises ABD-style operations to make them dynamic, and provides a reconfiguration service that emits a consistent sequence of configurations with the help of consensus; obsolete configurations are garbage-collected in the background. Interestingly, (non-)termination of consensus does not impact the operations in progress. The abstract Rambo algorithm is rigorously proved correct. Several optimizations are proved by “simulation” (a proof technique that shows trace inclusion). Proof-of-concept implementations were methodically derived from specifications. A retargeting of Rambo for mobile settings was explored in GeoQuorums [3], where the reconfiguration does not resort to consensus due to the finite number of configurations.


References

- 1 Marcos K. Aguilera, Idit Keidar, Dahlia Malkhi, and Alexander Shraer. Dynamic atomic storage without consensus. *Journal of the ACM*, 58:7:1–7:32, April 2011.
- 2 Hagit Attiya, Amotz Bar-Noy, and Danny Dolev. Sharing memory robustly in message-passing systems. *Journal of the ACM*, 42(1):124–142, January 1995.
- 3 Shlomi Dolev, Seth Gilbert, Nancy A. Lynch, Alexander A. Shvartsman, and Jennifer L. Welch. GeoQuorums: implementing atomic memory in mobile *ad hoc* networks. *Distributed Computing*, 18(2):125–155, 2005.

- 4 S. Gilbert, N. Lynch, and A. Shvartsman. RAMBO: A robust, reconfigurable atomic memory service for dynamic networks. *Distributed Computing*, 23(4):225–272, December 2010.

4.14 Snapshot Isolation with Eventual Consistency: Douglas B. Terry

Ioannis Nikolakopoulos and Ricardo Jiménez-Peris

License  Creative Commons BY 3.0 Unported license
© Ioannis Nikolakopoulos and Ricardo Jiménez-Peris
Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.TerryDouglas2.Slides.pptx>

Doug Terry presented a cloud storage system, the main goal of which is to provide: multiple consistency levels; read-write transactions on replicated and partitioned data with snapshot isolation; consistency based SLAs. The system features a Geo-Replication scheme where write operations take place in a datacenter that includes primary and secondary replicas, while read operations can take place in remote secondary replicas. The client API includes standard get and put operations, as well as begin/end transactions and sessions. The latter two have different consistency guarantees as a parameter, such as strong, Read-Modify-Write, . . . , Prefix.

At this point Alan Fekete asked if prefix consistency is the lowest sensible level. Doug answered that this happens only due to the specific implementation. Alan followed up asking if serializability is an option. The speaker answered that it is and explained the details in the last part of the talk.

The data are partitioned and there are primary and secondary servers per partition. The transactions implement snapshot isolation even across partitions. Version history is stored for every object as well as timestamps for the latest received write transaction (high-time) and the most recent discarded snapshot (low-time).

At this point there was a question if there is any need for transactions to be tagged as read-only. The answer was negative.

After that Doug got to the description of the key issue which was how to get the Read timestamp. He explained that this depends on the consistency chosen, giving specific examples.

In the case of Bounded Staleness there was a question regarding the need for synchronized clocks. The lecturer answered that in practice it does not matter as values for bounded staleness are much bigger than the clock error.

Furthermore, he analyzed the way to choose between available servers and how read-write transactions take place. The next question was what is the read set. Doug stated that transactions can specify and that by default all tablets will be read. Another comment was also that reads must block while a transaction is validating. The last question regarded the name of the system, which is Pilius.

4.15 ChainReaction: a Causal+ Consistent Datastore based on Chain Replication: Luís Rodrigues

Ioannis Nikolakopoulos and Ricardo Jiménez-Peris

License © Creative Commons BY 3.0 Unported license
© Ioannis Nikolakopoulos and Ricardo Jiménez-Peris
Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.RodriguesLuis.Slides.pdf>

ChainReaction is a system for data store that provides Causal+ consistency using a variant chain replication. The servers are organized in a One-Hop DHT with different objects spanning on different parts of the chain. The changes are requested to a proxy and propagate from the head of the respective chain to the tail. The reads are distributed along the chain for reducing the tail bottleneck, weakening though the consistency. The last node that the client read from is kept in metadata.

At this point there was a question about the size of the metadata. Luís answered that the worst case scenario is one entry per object. However, they are removed by a garbage collection process, and still their size is much less than competitive systems like COPS.

The experimental results showed that the performance is similar to the competition in balanced read and write operations and it gets much better the less the writes are. The next question was what can be done for the write operations to be optimized. Luís explained that this is done by returning the result before the change propagates to the tail. The last question was if the value of the metadata associated with a chain replica can be of any value. The answer was anything between 0 and the end.

4.16 Consistently Spanning the Globe for Fault-tolerance: Amr el-Abadi

Ioannis Nikolakopoulos and Ricardo Jiménez-Peris

License © Creative Commons BY 3.0 Unported license
© Ioannis Nikolakopoulos and Ricardo Jiménez-Peris
Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.ElAbadiAmr.Slides.pdf>

Amr gave an overview of the evolution and the recent situation of data management in the cloud, starting by the fact that databases became a scalability bottleneck around 2000. The first question to the speaker at that point was if he agrees that databases do not scale. He agreed and particularly mentioned that they only scale up as RDBMS achieve ACID and transactions in a single node. However they do not scale out as the key-value stores do.

NoSQL stores was the first wave of solutions for the scalability problem. However, Amr motivated that it would be nice if we achieved in having high level abstractions like “joins” and transactions and that is what we can expect as the new wave of solutions.

Continuing, the speaker presented different approaches on splitting RDBMS systems to multiple nodes, both static (ElasTras, SQL Azure, Megastore) and dynamic. Then he presented solutions for elasticity and fault tolerance via replication and how this solutions can be classified according to their consistency guarantees.

4.17 Consistency without consensus and linearizable resilient data types: Kaushik Rajan

Maurice Herlihy and Allen Clement

License © Creative Commons BY 3.0 Unported license

© Maurice Herlihy and Allen Clement

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.RajanKaushik1.Slides.pptx>

Paper <http://www.dagstuhl.de/mat/Files/13/13081/13081.RajanKaushik1.Paper.pdf>

Kaushik Rajan gave a talk titled “Consistency without consensus: linearizable resilient data types (LRDT).” He observed straight away that the title itself had evolved over the course of the workshop; many of the previous talks had introduced ideas and themes that were relevant and dramatically simplified the background and motivation for his talk.

Kaushik began the talk with a simple shared shopping cart example and observed that it is important that (a) the multiple versions of the cart are replicas of each other and (b) that it is frequently infeasible to run consensus to ensure consistency across replicas. Given these competing desires, he addresses the challenge of identifying when it is (im)possible to construct linearizable data types.

Kaushik then identified two key properties of operations on a specific data type: commutativity (intuitively $S + a + b = S + b + a$) and nullification (intuitively $S + a + b = S + b$). Kaushik explained that if at least one of these properties holds for every pair of update operations, then a LRDT is possible and impossible if there exists a pair of update operations for which neither property holds.

The positive construction relies on generalized lattice agreement. In lattice agreement, each replica may propose a value and when it does so waits for a majority of replicas to respond. Once a majority of replicas has responded, the value is accepted. The key to successfully building LRDTs using lattice agreement is to run an infinite sequence of instances of the protocol and ensure that subsequent executions of the protocol always return a superset of the operations returned by the previous instance.

Kaushik concluded the talk by presenting a graph data type implemented with lattice agreement. An interesting point of this datatype is that if the operations *AddEdge* and *RemoveVertex* are both included in the supported operations, then an LRDT graph is not possible; if either operation is removed then linearizability is achievable. This final point generated a fair bit of discussion.

4.18 ACID and modularity in the cloud: Liuba Shrira

Maurice Herlihy and Allen Clement

License © Creative Commons BY 3.0 Unported license

© Maurice Herlihy and Allen Clement

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.ShriraLiuba.Slides.ppt>

Liuba Shrira discussed ACID and modularity in the cloud. The talk highlighted the key tension surrounding transactions in cloud services: while they make life easy for application developers to reason about their system, efficiently supporting ACID transactions in multi-writer cloud services is non-trivial. At the core of this difficulty is allowing for disconnected operation and transactions to be issued/executed at different machines.

Liuba identified four basic strategies for implementing transactions: forcing serial execution, pessimistic concurrency control, optimistic concurrency control, and type-specific

concurrency control. She observed that while type specific techniques were popular topics of conversation in the 80s and early 90s, the techniques were not generally adopted for one simple reason: conventional wisdom says that concurrency control mechanisms must live in the concurrency control engine of the database. Given the extreme efforts that data base engineers go to to ensure that the database performs well, inserting application specific code into the finely optimized database engine was a non-starter.

In response to this tension, Liuba proposes a tiered architecture where type-specific transaction coordination is handled at client devices while all low-level (pessimistic) concurrency control is handled only at the servers. The two keys to Exo are client caches and reservations.

Clients in Exo are treated as caches, locally executing transactions which are pushed to the server at the next opportunity. The key step that allows these transactions to commit locally at the clients is reservations—essentially escrow portions of the object stored at the server that clients are able to acquire in advance. As long as a client has a reservation, it can perform and commit local transactions against the (portion of) the object managed by the reservation. The locally committed transactions are then guaranteed to be non-conflicting when they are reported back to the server (provided that the reservation is returned before it expires).

The Exo system demonstrates that type-specific concurrency control can be implemented outside of the optimized concurrency control mechanisms. Of specific importance in the context of modern systems is that Exo-leasing converts server-side concurrency control to client-side concurrency control, allowing for efficient eventually consistent systems.

4.19 Conditions for Strong Synchronization in Concurrent Data Types: Maged Michael

Alex Shvartsman and Luís Rodrigues

License © Creative Commons BY 3.0 Unported license
© Alex Shvartsman and Luís Rodrigues

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.MichaelMaged.Slides.pdf>

The talk started by addressing the problem of idempotent work stealing. In this problem one needs to maintain a data structure where available tasks are inserted and then removed by workers. Workers may steal tasks from another worker and, in the idempotent version of the problem, it is actually fine if two workers end up performing the same task. The work addressed the specific question of whether there are algorithms that do not require the task owner to execute expensive store-load fences or atomic operations. The question is answered in the affirmative, and the talk briefly addressed algorithms that only require the use of CAS in the steal method (for different policies, such as LIFO, FIFO, and double-ended). On this topic, Peter van Roy questioned the relevance of the idempotent work stealing for long running tasks (where multiple executions could be detrimental for performance). Maged noted that idempotent work stealing may only be applied in some settings.

The speaker then proceeded to address the more general problem of characterizing what problems cannot be solved without strong synchronization. In this context, the notion of Strong Non-Commutativity (SNC) was defined and it has been shown that problems with such characteristics require the use of strong synchronization. The talk discussed some ways to circumvent this limitation, such as changing the API of the data structure, or using semantics to build idempotent types, such that the operations do not exhibit SNC. In this context, David Lomet and Marcos Aguilera made a number of interesting remarks and questions regarding the different ways the designer could use to convert a SNC-API to an non-SNC API.

4.20 Commutativity, Inversion, and other Stories of Consistency and Betrayal: Maurice Herlihy

Alex Shvartsman and Luís Rodrigues

License © Creative Commons BY 3.0 Unported license
© Alex Shvartsman and Luís Rodrigues

The talk started by making a brief historical overview of how some of the consistency, performance, and fault-tolerant concerns that appear in concurrent applications have been addressed by the distributed and multicore programming communities.

Then the talk highlighted the performance limitations that can result from using software transactions to build concurrent programs that only consider read-write objects. This was illustrated by a simple object that returns unique ids (not necessarily consecutive). If this object is implemented using a read-write shared variable, transactions will encounter an unique id conflict. On the other hand, if this is exposed as an object that offers a commutative “getId” operation, the same transactions may not conflict. This reasoning leads to the conclusion that the entanglement between thread-level synchronization and transaction-level synchronization kills concurrency. Using this motivation, the talk advocated the use of an hierarchical approach, where thread level concurrency could be implemented by fine-grain, optimized, low level mechanisms and exposed as a black box the transaction level concurrency control mechanisms. Under this model, transaction recovery needs to be based on an operation log, and be implemented by applying the operations’ inverses. That talk also addressed the problem of supporting partial rollback and the most suitable abstractions for that purpose, conjecturing that checkpoints and rollback to a given checkpoint could be a suitable alternative to nested transactions (an abstraction that was presented as “widely admired but not widely implemented”).

In this context, David Lomet asked about the differences between the proposed approach and multi-level concurrency control schemes that have been designed for databases. Maurice agreed that there are similarities between the approaches.

Alan Fekete made a note about the difficulties in deriving the appropriate inverse operations to support recovery. Maurice clarified that in Scala, closures were used for this purpose.

Marcos Aguilera commented that without additional structure, doing partial recovery based on arbitrary checkpoints could result in code as hard to understand as code using goto’s. Maurice agreed that this was still a largely unexplored territory, and that structured approaches would need to be designed to take advantage of this approach.

Alexey Gotsman asked if some of the code used to implement concurrent objects was not redundant when considering that the object was going to be accessed in the context of transactions. Maurice answered that it was a reasonable price to pay for treating these library objects as black boxes.

4.21 Concurrent Data Representation Synthesis: Mooly Sagiv

Mike Dodds and Marcos Aguilera

License © Creative Commons BY 3.0 Unported license
© Mike Dodds and Marcos Aguilera

Mooly Sagiv presented a technique for synthesising fine-grained concurrent data-structures from high-level relational specifications.

Sagiv began by observing that concurrent data-structures are often used incorrectly in composite structures. He cited his paper from OOPSLA, which found that 38% of presumed linearizable algorithms in real code were in fact not linearizable. He suggested that a common failure was to use sequences of atomic operations and expect the resulting structure to be linearizable.

The aim of Sagiv's approach is to derive composite data-structures automatically out of linearizable container structures and a relational specification language. His target is low-level concurrent structures in Linux and similar. For a running example, he examined a Linux filesystem.

Sagiv's language, called RelScala, is translated down into Scala code. Data in Sagiv's approach is represented by a relation, combined with a DAG. The DAG is a high-level shape descriptor, used to represent the structure of the data in memory. Edges in the DAG correspond to sub-relations. For example, in the file-system, one edge accesses the `fs` portion of the relation, while another accesses the `inuse` portion. Sagiv's tool uses the DAG to automatically synthesize a data structure built out of primitive containers, together with methods for accessing the structure, where the methods rely on two-phase locking for concurrency control.

Alexey Gotsman asked why the DAG needed to encode multiple paths to a given node. Sagiv responded that applications such as Linux often feature multiple traversals of the data.

Sagiv described two modes of his approach: the user can define the DAG by hand, or the Autotuner tool can test many possible DAGs, primitive containers, lock placements, and lock implementations, and determine the best combination empirically given a particular workload.

Sagiv presented some performance results. One counter-intuitive result was that a representation with sharing performed better on a sequential processor, whereas copying worked better on a multiprocessor. Sagiv speculated that this resulted from interaction with the cache.

Marcos K. Aguilera and Bettina Kemme both asked why not simply use a database system. Sagiv responded that his approach would be faster by tuning for particular workloads. David Lomet commented that general databases can't control the workload, so they can't specialise.

Sagiv commented early in the talk that he also hoped to help concurrent database implementors, and Bettina Kemme asked how. Sagiv argued that low-level concurrent datastructures from Linux are highly optimised for particular applications. If one could compile an in-memory database into such data-structures, it might be possible to achieve better performance.

Alexey Gotsman asked whether the two-phase locking approach would be sufficient for an application such as Linux. Sagiv said that it isn't clear, but Linux does lots of things that his system couldn't currently handle.

4.22 Distributed unification as a basis for transparently managing consistency and replication in distributed systems: Peter van Roy

Mike Dodds and Marcos Aguilera

License © Creative Commons BY 3.0 Unported license

© Mike Dodds and Marcos Aguilera

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.VanRoyPeter.Slides.pdf>

Peter van Roy presented a model called deterministic data-flow programming, discussed the unification algorithm which drives this model, and drew connections to CRDT data-types for replication and consistency. The deterministic model is a form of concurrent functional programming. In it, variables can only be assigned once, and synchronisation is achieved by forcing threads to wait for variables to be assigned. Peter presented this model in the context of the Oz multi-paradigm language.

Much of Peter's talk discussed the unification algorithm underlying the deterministic data-flow programming model. The unification algorithm is a constraint solver for certain kinds of equality constraints. Peter first discussed a sequential algorithm based on rational trees (trees with back edges). He presented a set of operational semantics rules defining this algorithm, then observed that the algorithm could be distributed by changing only one rule: Bind. Doing this results in a distributed unification algorithm.

At the end of the talk, Peter discussed adapting this algorithm to CRDT data-types.

Marc Shapiro asked whether concurrent binding would need to search all the replicas. Peter said there exists a master node for each variable, which controls synchronisation. Shapiro said that this property does not hold for CRDTs, and wondered what the connection might be. Peter responded that his aim was to remove synchronisation.

Another participant asked whether the unification algorithm could be seen as movement up a lattice. Peter agreed with this, and observed that the algorithm could work with any monotonic process, for example adding edges to a graph.

4.23 Time bounds for shared objects in partially synchronous systems: Jennifer Welch

Pierre Sutra and Achour Mostefaoui

License © Creative Commons BY 3.0 Unported license

© Pierre Sutra and Achour Mostefaoui

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.WelchJennifer1.Slides.pptx>

Jennifer Welch presented several lower bound results on the cost of building atomic shared data structures in a partially-synchronous system. The focus was on objects of arbitrary type, with axiomatic specifications of the operations. This work extends previous work on the difference between sequential consistency and linearizability. Jennifer started with classical results on time complexity (lower and upper bounds) to execute operations on a logically shared atomic memory. Then, she presented new results, which improve the lower bounds on elapsed time for executing operations on a linearizable object (such as a queue or a stack). The proof technique is the classical shifting technique (indistinguishably argument) and an extension of the classical technique to allow larger lower bounds.

The proposed bounds are tight or almost tight in many cases. The new algorithms split operations into accessors (read), mutators (write), or both (read-modify-write). This talk ends with several open problems: How to tighten gaps between lower and upper bounds? Is

it possible to consider clock drift, failures, churn, etc., in the results? How to extend those results to cover other consistency criteria?

Hagit Attiya points out that those results refine the CAP impossibility result.

4.24 Reduction theorems for proving serialisability with application to RCU-based synchronisation: Hagit Attiya

Pierre Sutra and Achour Mostefaoui

License © Creative Commons BY 3.0 Unported license
© Pierre Sutra and Achour Mostefaoui

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.AttiyaHagit.Slides.pptx>

Hagit Attiya presented a reduction theorem for proving serializability, with application to RCU-based synchronization.

The talk starts with the core idea of sequential reduction: under certain assumptions, one can show that if property P holds in sequential executions, then P holds in all executions. Hagit showed that this reasoning is correct for local locking policies (e.g., tree locking or two-phase locking), i.e., policies that do not employ a centralized concurrency control mechanism. Consequently, for any program M respecting a local locking policy, if M maintains its invariants during all sequential executions, then M maintains its invariants during all interleaved executions.

The core of the talk was the reduction theorem. The proof of this theorem makes use of a classical indistinguishability argument.

RCU (for Read-Copy-Update) is a mechanism that allows read-only transactions to read data, even while they are locked for update. Linux developers intensively use RCU-based synchronization. Hagit pointed that this mechanism is not well understood; for instance scan operations in presence of concurrent updates.

She presented work in progress that aims at applying the above reduction theorem to RCU-based synchronization. The idea is to apply the theorem to sub-executions which contain only updates, then to superimpose individual steps of the read-only operations.

At the end of the talk, Bernadette Charron-Bost asked how the reduction theorem relates to Lipton's theorem [1].

References

- 1 Richard J. Lipton. Reduction: a method of proving properties of parallel programs. *Communications of the ACM*, 18(12):717–721, December 1975.

4.25 Abstractions for Transactional Memory: Noam Rinetzky

Annette Bieniusa and Peter van Roy

License © Creative Commons BY 3.0 Unported license
© Annette Bieniusa and Peter van Roy

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.RinetzkyNoam.Slides.pdf>

Noam presented a technique called observational refinement for decomposing correctness proofs for Transactional Memory (TM) algorithms implementing opacity. With observational refinement, all possible views of an implementation are contained in the set of possible views admitted by the specification. In this setting, a view is the restriction of an execution

history to a thread. Specifically, under opacity every history has an equivalent sequential history, including aborted transactions, such that real-time order is preserved. Noam then introduced a programming language where global variables are only accessed outside atomic blocks, atomic variables only inside atomic blocks. For this language, he sketched a proof of soundness and completeness for observational refinement with respect to opacity based on well-formed traces.


Annette Bieniusa wanted to know what proof obligations remain for obtaining a correctness proof for a concrete TM implementation. Noam explained that using their results it suffices to show that the language implements opacity.

Doug Terry asked how aborted transactions can be observable, even though they do not have any effect. Noam clarified that the language definition allows aborted transactions to issue side-effects by modifying local state, similar to “nested top actions” in database systems.

Hagit Attiya then underlined the relevance of this model as upcoming Hardware Transactional Memory systems are implementing these semantics.

4.26 Idempotent transactional workflow: Ganesan Ramalingam

Annette Bieniusa and Peter van Roy

License  Creative Commons BY 3.0 Unported license

© Annette Bieniusa and Peter van Roy

Slides <http://www.dagstuhl.de/mat/Files/13/13081/13081.RamalingamGanesan.Slides.pptx>

This talk focused on a decentralised technique for realizing idempotent transactional workflows over partitioned data.

Data partitioning is commonly used to achieve horizontal scaleout. Applications can potentially leave persistent data in an inconsistent state if the applications fail in the middle. This problem is particularly acute when the persistent data is partitioned. ACID transactions are one solution to the problem of ensuring consistency of persistent data in the presence of application (or transaction) failures. However, when data is partitioned, this requires the use of distributed transactions, which can be a performance concern.

The talk observed that transactional workflows are a common and useful idiom in applications that work with partitioned data. In its simplest form, a workflow is a sequential fault-tolerant composition of (ACID) transactions. These workflows are often required to be idempotent. An idempotent transactional workflow was presented as a useful language construct. The talk described a decentralized implementation technique for implementing such workflows without using consensus or any equivalent coordination across the different partitions. This approach can be extended with compensating actions, automatic retry, and checkpointing.

Many questions were asked during the talk:

- Q:** What happens if the first transaction commits and the second doesn't, you need to roll back the first one? **A:** Yes, to handle that case the programmer must provide compensations.
- Q:** You could use transaction chopping, only the first can abort (all checking is put there), if it commits the rest will too. **A:** Our approach handles cases where this doesn't work.
- Q:** This corresponds to multilevel transaction model of the 1990s, which has compensations. **A:** Yes.

Q: Do you need a scheduler for guaranteeing the order of the flow? **A:** We only need a scheduler for performance, correctness is guaranteed by the model.

Q: How do you guarantee global uniqueness of ids? **A:** It's up to the programmer.

4.27 BubbleStorm: replication, updates and consistency in rendezvous information systems: Alejandro Buchmann and Robert Rehner

Annette Bieniusa and Peter van Roy

License © Creative Commons BY 3.0 Unported license
© Annette Bieniusa and Peter van Roy

BubbleStorm is a peer-to-peer system that organizes its peers probabilistically in order to provide different forms of document management, including publish/subscribe and document query, in an environment with high churn. BubbleStorm uses bubbles, i.e., range-limited flooding, in its routing algorithm. Publication bubbles must intersect with query bubbles, which is guaranteed with high probability through the topology management. To manage churn, the system relaxes consistency of its replicated nodes.

BubbleCast is the algorithm used to build the search trees. It stores in non-persistent fashion queries, events, notifications, position updates, and caches. Maintainer-based replication is organized by a manager in a storage pool. When a manager leaves, data is flushed or destroyed. Collective replication is durable and performed by a random set of nodes. Information is flooded using Lamport clocks for consistency. There is a flexible evaluation framework for simulation and deployment. They implemented and demoed a first-person space shooter game based on a “vision range.”

Many questions were asked about this system.

Q: What is the difference with quorum systems? **A:** Placement of replicas.

Q: How much of the system can be used as building blocks? **A:** All the lower layers, event scheduler, network. communication, have been used in many student projects.

Q: What about the visualization interface? **A:** Yes, and in addition we have a statistics interface.

Q: How would you measure inconsistencies in your system? Using logs? **A:** Yes, we could do this with postprocessing, with a testbed application that uses timestamps in a database. Note that a lot of information is gathered in various parts of the system, including the simulator, and we use it to gather different kinds of statistics.


Q: Could your analysis tool for communication patterns be used to analyze other systems? **A:** It has a clean interface, so this should be possible.

Q: What is G-Lab? **A:** A German system similar to PlanetLab. It is closer to (German) users and more stable than PlanetLab. It may be possible to use it outside of Germany, this is a legal issue.

5 Breakout sessions

5.1 Breakout sessions on distributed applications

Pawel Wojciechowski and Noam Rinetzky

License  Creative Commons BY 3.0 Unported license
© Pawel Wojciechowski and Noam Rinetzky

Participants were divided into four breakout groups. The common topic was the consistency levels required in a distributed application, either a multiplayer online game or an e-commerce application. All groups selected the games, since they typically show a great deal of variety of interaction and synchronization patterns.

Group 1

Group members Annette Bieniusa and Yiannis Nikolakopoulos discussed a massive multiplayer online game, in which groups of users travel a virtual world performing various tasks, such as looking for weapons, killing monsters, etc. A player holds replicas of immutable objects, describing the static world (trees, buildings, etc.), and mutable objects such as players, monsters, weapons, gifts, etc. Object attributes include position, access order, and ownership.

A player holds the master copy of its own coordinates, and needs to see only those players that are located in the immediate vicinity. An object or field is assigned a specific consistency level; it may change as the game evolves. For example, a player may observe another one that is sufficiently close (in space or time).

These requirements lead to the monotonic reads for writers and bounded staleness for read-only replicas. Consider the special case of fighters against monsters: since it is generally not important who hit the monster first, this commutativity the use of CRDTs.

A player owns items like (virtual) money, weapons, etc. These should be persistent, and transfer of ownership must be transactional, e.g., using two-phase-commitment and conflict detection. An interesting issue is picking up items: if two players try to pick up the same item at the same time, then normally the closer player wins (bounded staleness).

Group 2

André Schiper summarized the discussion in his group. The discussion was focused on the design of a game prototype developed by Alejandro Buchmann and his students (see their talk later in the workshop). It is a P2P multiplayer shooter game. Each player has a sphere of visibility. Another player inside my visibility sphere can interact directly with me, requiring strong consistency or bounded staleness. Eventual consistency suffices for players outside my sphere since we do not interact. As players move, the contents of a sphere changes dynamically. Players from different teams may interact by exchanging messages.

The solution to picking-up items is similar to Bettina's group, but, sometimes, strong consistency may be needed, not just bounded staleness.

Several design problems were discussed: How to combine different levels of consistency? How to simplify programming? Ideally, application programmers should not be burdened by consistency issues. They should be able to assume ideal (strong) consistency, but give criteria for relaxing consistency. The system should be able to switch between the different kinds of consistency.

Another issue is persistence. What should persist across sessions? For example, individual shots are not persistent, but the *results* of shooting definitely must be.

André pointed out that game state and what users observe are separate. Alejandro proposed to have certain criteria for changing consistency levels. The criteria could benefit from setting thresholds that would tell the system when to switch dynamically between different levels of consistency.

Group 3

Ganesan Ramalingam reports for many games, weak consistency is sufficient. His group started with a simple game called Wordament, in which players try (in parallel) to write words using a set of letters. There is no interaction between the players. Each player sends messages with suggested words. The game establishes an ordering between messages from different users, and give feedback. The first player who identifies a word gets a score.

It would be nice to observe real-time order on operations. A conflict, such as multiple players finishing concurrently, can be resolved by giving them the same score. Note that strong consistency is easy to achieve, because everything is done on the server.

The second scenario is a game similar to “Angry Birds,” but with multiple players. If multiple players shoot the same bird, the first one to hit it gets the score. This can be resolved by a central server, as there is enough time to compute a non-ambiguous result (strong consistency).

The third game consists of users collaborating to solve a puzzle. Here, we need a state merge function, as changes done by different users might conflict. The application includes constraints, e.g., in sudoku, no digit is allowed to be used more than once. The game informs the user if its move was overridden; thus, there is no need for strong consistency.

Other techniques can be useful. For instance, a player may use Escrow to make reservations for future operations; for instance, a player may mark an area of the puzzle as his. If the reservation is successful, he can proceed under bounded staleness.

Group 4

Marc Shapiro reports that his group considered both games and e-commerce applications, which share some elements (think of virtual money, etc.). Games were considered more exciting, because state is more complicated and there is more interaction. Games may be easier, since correctness constraints are set by the designer, failures are acceptable, and anonymity is accepted.

The game design discussed was similar to existing commercial systems. The virtual world is divided into disjoint rooms, where all players in a same room are on the same server. Putting multiple users on the same server allows to achieve strong consistency cheaply; there is weak consistency between rooms. They also discussed fairness (e.g., players with shorter network round-trip-time can be slowed down) and functional features (e.g., what anomalies does the game tolerate?).

Alejandro concluded that it was interesting to see that there are so many different levels of consistency which are *not* necessarily the same in every game — different games require different levels of consistency.

5.2 Breakout Groups and Discussion of Workshop Followup

Alex Shvartsman, Lués Rodrigues, Pierre Sutra and Achour Mostefaoui

License © Creative Commons BY 3.0 Unported license
© Alex Shvartsman, Lués Rodrigues, Pierre Sutra and Achour Mostefaoui

Doug Terry announced the formation of four breakout groups for the after-lunch brainstorm sessions. The topics to be discussed include: *(i)* Consistency models, tools; *(ii)* Automatic analysis of consistency requirements; *(iii)* Performance impact of consistency models; *(iv)* Theory and potential help for developers.

The break-out groups are to answer the question: “What can the research community do to help application developers understand the consequences of choosing a particular consistency?”

Group 1

Members of Group 1 identified the basic characteristics of an application for which consistency is important. They listed the nature of operations (idempotence, commutativity), the atomicity of groups of operations, the ordering between operations, and the staleness of reads.

Marc Shapiro pointed out that checking some of these properties cannot be done locally, since they are inherently global.

Group 1 then listed several questions related to these properties. In particular, how to extract the above characteristics from the application, and how to capture design patterns developers use to build concurrent programs. Solutions include static analysis and the use of synthetic workloads. The results returned by these tests can be both quantitative (e.g., performance of some workload) and qualitative (e.g., executing a workload throws an exception).

The report closed with an observation by Alan Fekete: In the context of databases, several studies of the performance difference between consistency levels conclude that the difference is small, because the bottleneck is disk access.

Group 2

André Schiper listed several ideas studied by Group 2. The first one is that model checking may help understanding the consequences of concurrency. Another to look at design patterns promoting good programming practices for weakly-consistent applications.

Somebody pointed out that this is a non-issue: a developer should always first start with atomicity; then, if performance is not sufficient, think about choosing another consistency criterion.

Group 2 proposes that, to help with the development of concurrent program, programmers have to be trained with a non-strongly consistent API (e.g., Cassandra).

This leads to another question: When should a programmer make the decision of what consistency is needed, and how to introduce it into the program ?

At the end of the presentation, André Schiper underlined that partitioning was out of scope, since it is well understood and extensively covered in other studies.

Group 3

This group observes that developers do not understand what a consistency level means. They always assume that APIs expose strongly consistent operations. Furthermore, a programmer should consider what is executable under weak consistency at the level of the specification, and *not* at the level of the implementation. An ideal programmer, and by extension an ideal system, would choose among different consistency criteria based on the specification. For instance, one could consider a program whose integrity properties would vary according to the consistency criterion employed in the storage system.

A request is to make storage systems more testable, by forcing rare consistency violations to occur.

Group 4

Group 4 started with the following analogy: “A developer may turn a knob to change the consistency level of her application. What should we tell to the developer ?” They listed several refinements of this discussion:

- For each position of the knob, is it possible to give useful information to the developer?
- Can we build a tool that will tell the developer what will go wrong with her application when turning the knob? Several persons in the audience pointed out that in most cases this is non-tractable.
- Are assertions enough to understand correctness of a concurrent program?
- Can we tell if a program can be safely restarted?

To address the above questions, the group discussed the properties of such a “magic” tool. It would vary three properties: consistency, availability and performance (throughput or response time). This tool would use semantic analysis of the application, with annotations from the developer, and typical workloads. Annotations are necessary to make the analysis tractable. It might work as a model checker, testing application invariants while varying consistency of the APIs used by the application.

6 Workshop Followup**6.1 Consensus paper**

Alex Shvartsman and Luís Rodrigues

License © Creative Commons BY 3.0 Unported license
© Alex Shvartsman and Luís Rodrigues

Marc Shapiro offered for discussion of what should be the written outcome of the workshop. The options on the table were: (a) do nothing; (b) write a common “consensus” paper that would summarize and integrate the different perspectives discussed during the workshop; (c) produce a collection of (individual, independent) papers; (d) to produce a coherent book with original material, along the lines of the book produced after the Monte Verità 2007 workshop [1]. Hagit Attyia proposed to start with the consensus paper, then aim to produce an original content book. After a lively discussion it was agreed to start on a common consensus paper, then consider producing the book based on the outcome of the first task.

References

- 1 Bernadette Charron-Bost, Fernando Pedone, and André Schiper, editors. *Replication: Theory and Practice*, volume 5959 of *Lecture Notes in Comp. Sc.* Springer-Verlag, 2010. A 30-Year Perspective on Replication, Monte Verità, Ascona, Switzerland, November 2007.

Participants

- Marcos K. Aguilera
Microsoft – Mountain View, US
- Hagit Attiya
Technion – Haifa, IL
- Carlos Baquero
Univ. de Minho – Braga, PT
- Annette Bieniusa
TU Kaiserslautern, DE
- Alejandro P. Buchmann
TU Darmstadt, DE
- Sebastian Burckhardt
Microsoft – Redmond, US
- Bernadette Charron-Bost
Ecole Polytechnique –
Palaiseau, FR
- Allen Clement
MPI für Softwaresysteme –
Saarbrücken, DE
- Mike Dodds
University of York, GB
- Amr El-Abbadi
University of California – Santa
Barbara, US
- Alan Fekete
The University of Sydney, AU
- Pascal Felber
Université de Neuchâtel, CH
- Carla Ferreira
Universidade Nova de Lisboa, PT
- Alexey Gotsman
IMDEA Software – Madrid, ES
- Maurice Herlihy
Brown Univ. – Providence, US
- Ricardo Jimenez-Peris
Univ. Politec. de Madrid, ES
- Bettina Kemme
McGill University, CA
- Petr Kuznetsov
TU Berlin, DE
- David B. Lomet
Microsoft Res. – Redmond, US
- Maged M. Michael
IBM TJ Watson Research Center
– Yorktown Heights, US
- Achour Mostefaoui
Univ. de Nantes, FR
- Yiannis Nikolakopoulos
Chalmers UT – Göteborg, SE
- Fernando Pedone
University of Lugano, CH
- Nuno Prego
Universidade Nova de Lisboa, PT
- Vivien Quema
INRIA Rhône-Alpes, FR
- Kaushik Rajan
Microsoft Research India –
Bangalore, IN
- Ganesan Ramalingam
Microsoft Research India –
Bangalore, IN
- Robert Rehner
TU Darmstadt, DE
- Noam Rinetzky
Tel Aviv University, IL
- Luís Rodrigues
Technical Univ. – Lisboa, PT
- Rodrigo Rodrigues
Universidade Nova de Lisboa, PT
- Nicholas Rutherford
UC Louvain-la-Neuve, BE
- Mooly Sagiv
Tel Aviv University, IL
- André Schiper
EPFL – Lausanne, CH
- Marc Shapiro
INRIA & LIP6 – Paris, FR
- Liuba Shrira
Brandeis Univ. Waltham, US
- Alexander A. Shvartsman
University of Connecticut –
Storrs, US
- Pierre Sutra
Université de Neuchâtel, CH
- Douglas B. Terry
Microsoft – Mountain View, US
- Peter Van Roy
UC Louvain-la-Neuve, BE
- Kapil Vaswani
Microsoft Research India –
Bangalore, IN
- Marko Vukolic
EURECOM – Biot, FR
- Jennifer L. Welch
Texas A&M University – College
Station, US
- Pawel T. Wojciechowski
Poznan Univ. of Technology, PL



Communication Complexity, Linear Optimization, and lower bounds for the nonnegative rank of matrices

Edited by

LeRoy B. Beasley¹, Hartmut Klauck², Troy Lee³, and Dirk Oliver Theis⁴

1 Utah State University, US, leroy.b.beasley@usu.edu

2 Nanyang TU – Singapore, SG, cqthk@nus.edu.sg

3 National University of Singapore, SG, troyjlee@gmail.com

4 Universität Magdeburg, DE, dirk.oliver.theis@ut.ee

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13082 “Communication Complexity, Linear Optimization, and lower bounds for the nonnegative rank of matrices”.

Seminar 17.–22. February, 2013 – www.dagstuhl.de/13082

1998 ACM Subject Classification G.1.6 optimization, F.1.3 complexity measures and classes

Keywords and phrases nonnegative rank, combinatorial optimization, communication complexity, extended formulation size

Digital Object Identifier 10.4230/DagRep.3.2.127

1 Executive Summary

LeRoy B. Beasley

Hartmut Klauck

Troy Lee

Dirk Oliver Theis

The nonnegative rank is a measure of the complexity of a matrix that has applications ranging from Communication Complexity to Combinatorial Optimization. At the time of the proposal of the seminar, known lower bounds for the nonnegative rank were either trivial (rank lower bound) or known not to work in many important cases (bounding the nondeterministic communication complexity of the support of the matrix).

Over the past couple of years in Combinatorial Optimization, there has been a surge of interest in lower bounds on the sizes of Linear Programming formulations. A number of new methods have been developed, for example characterizing nonnegative rank as a variant of randomized communication complexity. The link between communication complexity and nonnegative rank was also instrumental recently in proving exponential lower bounds on the sizes of extended formulations of the Traveling Salesman polytope, answering a longstanding open problem.

This seminar brought together researchers from Matrix Theory, Combinatorial Optimization, and Communication Complexity to promote the transfer of tools and methods between these fields. The focus of the seminar was on discussions, open problems and talks surveying the basic tools and techniques from each area.

In the short time since the seminar, its participants have made progress on a number of open problems.



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Communication Complexity, Linear Optimization, and lower bounds for the nonnegative rank of matrices, *Dagstuhl Reports*, Vol. 3, Issue 2, pp. 127–143

Editors: LeRoy B. Beasley, Hartmut Klauck, Troy Lee, and Dirk Oliver Theis



DAGSTUHL
REPORTS Dagstuhl Reports

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

Program Overview

Background lectures on the connection between matrix factorizations to Communication Complexity and to Combinatorial Optimization were given by the organizers. More importantly, a number of participants contributed their latest research on factorization ranks. In this section, we summarize these talks.

Extended Formulations and Linear Optimization

Hamza Fawzi

Many lower bounds on the nonnegative rank only make use of the zero/nonzero pattern of the matrix. For certain applications, in particular for the extended formulation size lower bounds for approximation problems, nonnegative rank lower bounds need to be shown for matrices that are strictly positive. Hamza discussed an interesting approach to nonnegative rank lower bounds via conic programming that does not only rely on the zero/nonzero structure of the matrix. The bound is in many ways analogous to the trace norm lower bound for rank, but making use of the stronger fact that the factorization is nonnegative leads to a copositive program rather than a semidefinite one. For computing the bound in practice, Hamza discussed ways to approximate the bound by semidefinite programs, and examples of using this in practice.

Sam Fiorini

There is a rich theory on the hardness of approximating NP-optimization problems up to certain factors, given complexity assumptions like $P \neq NP$. Very recently a similar topic has emerged in the study of polytopes. Sam talked about tradeoffs between the approximation ratio and the size of linear formulations. One notable result in Sam's talk was that approximating CLIQUE to within $n^{1/2-\epsilon}$ requires extended formulations of exponential size.

Complexity

Nati Linial

On the first day, Nati Linial treated us to a survey of higher dimensional analogs of familiar combinatorial objects. For example, we are very familiar with permutation matrices, those matrices with entries from $\{0, 1\}$ with exactly one 1 in every row and column, and know that there are $n! = ((1 + o(1))n/e)^n$ many of them. What about 3-dimensional tensors with entries from $\{0, 1\}$ and exactly one 1 along every row, column and shaft? Such 2-dimensional permutations turn out to coincide with latin squares and it is known that there are $((1 + o(1))n/e^2)^{n^2}$ many of them. This relies on some beautiful work on the minimum permanent of doubly stochastic matrices. Nati conjectures that the formula generalizes to count the number of d -dimensional permutations, described by a $d+1$ -tensor with one 1 along every line. That is, that the number of d -dimensional permutations is $((1 + o(1))n/e^d)^{n^d}$. He is able to show such an upper bound, but the lower bound remains open.

Sebastian Pokutta

In order to prove that extended formulations for approximating optimization problems need to be large, communication and information complexity are important tools. In his talk Sebastian described a new approach on how to prove lower bounds on the nonnegative rank of matrices corresponding to the unique disjointness problem when perturbed. He gave tight lower bounds using a new information theoretic fooling set method.

Since the seminar, Sebastian and his co-author Gabór Braun have made available a preprint containing these results [3].

Hans Raj Tiwary

There are entire books of NP-complete problems and explicit reductions between them. For the extension complexity of the associated polytopes, however, this book is still slowly being written—usually by arguing that P is a projection of Q or finding P as a face of Q . Hans discussed the intriguing possibility of automatically turning an NP gadget reduction into a polytope reduction. While still not a general theory, Hans can currently do this for many NP-hard problems and their associated polytopes.

Nicolas Gillis

Nicolas spoke about the problem of actually computing a non-negative factorization of a nonnegative matrix. This talk was important to seminar participants on small matrices, allowing them to test the quality of their lower bounds against upper bounds. On small matrices, these upper bounds can be found computationally. The problem also has applications to compression of images, to identifying topics in documents, even to identifying the mineral composition of rocks from spectral data (hyper-spectral imaging). Nicolas discussed specifically the case of separable matrices. An n -by- n matrix M is r -separable if it has a factorization $M = WH$ where W is n -by- r , H is r -by- N and moreover W is a subset of the columns of M . Such types of factorization can be more useful in practice. Nicolas talked about a linear programming approach to this problem that is polynomial time and moreover outperforms previous approaches in practice.

Matrix Theory

Alexander Guterman

Alexander Guterman gave a survey talk on various matrix ranks over semirings. A big focus was on tropical algebra over the real number with operations $a \oplus b = \max a, b$ and $a \otimes b = a + b$. Tropical algebra provides a way of formulating many hard combinatorial optimization problems (like scheduling problems) in terms of a very elegant linear algebraic type language. In tropical linear algebra there are varying notions of linear independence, for example Gondran-Minoux independence, weak linear independence, and strong linear independence. Each of these gives rise to a different notion of rank of a matrix and a hierarchy of these ranks is known.

Yaroslav Shitov

Yaroslav continued talking about tropical matrix rank, in particular the tropical factorization rank. This is defined as the minimum k such that $A = B \otimes C$ for a n -by- k matrix B and k -by-

n matrix C . Note that in tropical matrix multiplication $(B \otimes C)(i, j) = \min_t B(i, t) + C(t, j)$. Yaroslav mentioned a very interesting application of the tropical factorization rank. Say that we are given an instance of the traveling salesman problem, with distances specified by a matrix A , and moreover we are given a tropical factorization $A = B \otimes C$ that witnesses that A has constant factorization rank. Then the resulting traveling salesman instance can be solved in polynomial time! This is a result of Barvinok, Johnson, Woeginger, and Woodroofe. Yaroslav also showed that the problem of detecting if the tropical factorization rank of a matrix is at most 8 is NP-hard.

Richard Robinson

In his talk, Richard Robinson gave a characterization, among all nonnegative matrices, of the extreme-ray / facet slack matrices of polyhedral cones, and vertex/facet slack matrices of polytopes. This characterization leads to an algorithm for deciding whether a given matrix is a vertex/facet slack matrix. The underlying decision problem is equivalent to the polyhedral verification problem whose complexity is unknown.

2 Table of Contents

Executive Summary

LeRoy B. Beasley, Hartmut Klauck, Troy Lee, Dirk Oliver Theis 127

Overview of Talks

New lower bounds on nonnegative rank using conic programming
Hamza Fawzi 133

Approximation Limits of Linear Programs (Beyond Hierarchies)
Samuel Fiorini 133

Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization
Nicolas Gillis 134

On the Geometric Interpretation of the Nonnegative Rank
François Glineur 134

Matrix ranks over semirings
Alexander Guterman 135

Constructing Extended Formulations for Stable Set Polytopes via Decomposition Rules
Kanstantsin Pashkovich 135

On lower bounds for extended formulations
Sebastian Pokutta 136

Which nonnegative matrices are slack matrices?
Richard Robinson 136

Matrix factorization over semirings
Yaroslav Shitov 136

On the extension complexity of combinatorial polytopes
Hans Raj Tiwary 137

Problem discussion sessions, and subsequent developments

Real vs. rational nonnegative rank 137

Square-root rank 138

Positive semidefinite rank of matrices defined by polynomials 138

A query complexity problem 138

A rigidity-type question 139

Extension complexity of stable set polytopes of split-graph-free perfect graphs . . . 139

Matrices with low non-negative rank are a low-dimensional subset in the manifold of rank-bounded matrices 139

Fooling-sets and rank 140

Polygons — or, more generally, rank-3 matrices 140

Euclidean distance matrices 140

How does nonnegative rank behave under tensor products? 140

132 13082 – Communication Complexity, Linear Optimization, and lower bounds ...

Vertex/facet slack matrices vs. general nonnegative matrices 141

Extensions/factorizations over the positive semidefinite cone 141

Conclusion 141

Participants 143

3 Overview of Talks

3.1 New lower bounds on nonnegative rank using conic programming

Hamza Fawzi (MIT – Cambridge, US)

License © Creative Commons BY 3.0 Unported license
© Hamza Fawzi

Joint work of Fawzi, Hamza; Parrilo, Pablo

Main reference H. Fawzi, P.A. Parrilo, “New lower bounds on nonnegative rank using conic programming,” arXiv:1210.6970v1 [math.OC], 2012.

URL <http://arxiv.org/abs/1210.6970v1>

We propose a new lower bound on the nonnegative rank which, unlike most existing lower bounds, does not explicitly rely on the matrix sparsity pattern and applies to nonnegative matrices with arbitrary support. Our lower bound is expressed as the solution of a copositive programming problem and can be relaxed to obtain polynomial-time computable lower bounds using semidefinite programming. The idea involves computing a certain nuclear norm with nonnegativity constraints which allows to lower bound the nonnegative rank, in the same way the standard nuclear norm gives lower bounds on the standard rank. We compare our lower bound with existing ones, and we show examples of matrices where our lower bound performs better than currently known ones.

3.2 Approximation Limits of Linear Programs (Beyond Hierarchies)

Samuel Fiorini (University of Brussels, BE)

License © Creative Commons BY 3.0 Unported license
© Samuel Fiorini

Joint work of Braun, Gabor; Fiorini, Samuel; Pokutta, Sebastian; Steurer, David

Main reference G. Braun, S. Fiorini, S. Pokutta, D. Steurer, “Approximation Limits of Linear Programs (Beyond Hierarchies),” arXiv:1204.0957v2 [cs.CC], 2013.

URL <http://arxiv.org/abs/1204.0957v2>

We develop a framework for proving approximation limits of polynomial-size linear programs from lower bounds on the nonnegative ranks of suitably defined matrices. This framework yields unconditional impossibility results that are applicable to any linear program as opposed to only programs generated by hierarchies. Using our framework, we prove that $O(n^{1/2-\epsilon})$ -approximations for CLIQUE require linear programs of size $2^{n^{\Omega(\epsilon)}}$. (This lower bound applies to linear programs using a certain encoding of CLIQUE as a linear optimization problem.) Moreover, we establish a similar result for approximations of semidefinite programs by linear programs.

Our main technical ingredient is a quantitative improvement of Razborov’s rectangle corruption lemma (1992) for the high error regime, which gives strong lower bounds on the nonnegative rank of certain perturbations of the unique disjointness matrix.

3.3 Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization

Nicolas Gillis (UC Louvain-la-Neuve, BE)

License © Creative Commons BY 3.0 Unported license
© Nicolas Gillis

Main reference N. Gillis, R. Luce, “Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization,” arXiv:1302.4385v1 [stat.ML], 2013.

URL <http://arxiv.org/abs/1302.4385v1>

Nonnegative matrix factorization (NMF) has been shown recently to be tractable under the separability assumption [1], which amounts for the columns of the input data matrix to belong to the convex cone generated by a small number of columns. Since then, several algorithms have been proposed to handle this subclass of NMF problems under any small perturbation of the input matrix, see for example [2] and the references therein. In particular, [3] proposed a linear programming (LP) model, referred to as HottTopixx; see also [4]. However, HottTopixx has two important drawbacks: (i) the input matrix has to be normalized, and (ii) the factorization rank has to be known in advance. In [5], we generalize HottTopixx in order to resolve these two drawbacks, that is, we propose a new LP model which does not require normalization and detects the factorization rank automatically. Moreover, the new LP model is more flexible, significantly more tolerant to noise, and can easily be adapted to handle outliers and other noise models. Finally, we show on several synthetic datasets that it outperforms HottTopixx while competing favorably with two state-of-the-art methods.

References

- 1 S. Arora, R. Ge, R. Kannan, and A. Moitra, ‘Computing a Nonnegative Matrix Factorization – Provably’, STOC 2012.
- 2 N. Gillis and S.A. Vavasis, ‘Fast and Robust Recursive Algorithms for Separable Nonnegative Matrix Factorization’, arXiv:1208.1237.
- 3 V. Bittorf, B. Recht, C. Re, and J.A. Tropp, ‘Factoring Nonnegative Matrices with Linear Programs’, NIPS 2012.
- 4 N. Gillis, ‘Robustness Analysis of HottTopixx, a Linear Programming Model for Factoring Nonnegative Matrices’, arXiv:1211.6687.
- 5 N. Gillis and R. Luce, ‘Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization’, <http://arxiv.org/abs/1302.4385>.

3.4 On the Geometric Interpretation of the Nonnegative Rank

Francois Glineur (UC Louvain, BE)

License © Creative Commons BY 3.0 Unported license
© Francois Glineur

Joint work of Gillis, Nicolas; Glineur, Francois

Main reference On the Geometric Interpretation of the Nonnegative Rank, Nicolas Gillis, François Glineur, Linear Algebra and its Applications, Volume 437, Issue 11 (1 December 2012), Elsevier.

A geometric bound for the nonnegative rank Nicolas Gillis and François Glineur, UCLouvain (Belgium)

We start with a brief introduction to the nonnegative rank and recall what is currently known about its computational complexity. We then present a geometric view of the nonnegative rank computation, which leads us to introduce the concept of restricted nonnegative rank. This allows us to derive a new lower bound for the nonnegative rank. In particular, it

is easily computable for slack matrices, and provides a lower bound on the size of extended formulations for any d -polytope with a given number of facets and vertices. To conclude, we report the results of recent computational experiments attempting to factorize numerically slack matrices of low-dimensional polytopes.

Paper about the first half of the talk: On the Geometric Interpretation of the Nonnegative Rank, Nicolas Gillis, François Glineur, *Linear Algebra and its Applications*, Volume 437, Issue 11 (1 December 2012), Elsevier. 10.1016/j.laa.2012.06.038

Code to attempt factorisation of slack or other nonnegative matrices
<http://sites.google.com/site/nicolasgillis/code> (direct link <http://bit.ly/13hvCA2>)

3.5 Matrix ranks over semirings

Alexander Guterman (Moscow State University, RU)

License  Creative Commons BY 3.0 Unported license
 © Alexander Guterman

Tropical algebra (sometimes called max algebra) is a set of real numbers equipped with the maximum operation instead of usual addition and addition instead of usual multiplication. Under these operations this is an algebraic structure called a semiring. The other typical examples of such structures are non-negative integers, non-negative reals, boolean algebras. Semirings naturally appear in different problems of communication complexity, scheduling theory, optimization, dynamical systems, etc. Semiring arithmetics allows to reduce non-linear problems to the linear problems but over semirings. To investigate these problems it is necessary to develop linear algebra over semirings. This subject is very actual nowadays. Different rank functions over various classes of semirings are intensively investigated during the last decades. We plan to introduce and investigate some of them, in particular, factor rank, tropical rank, nonnegative rank, determinantal rank, Gondran-Minoux rank. We plan to compare these functions and discuss their interrelations. Among the other topics we shall discuss our recent joint research results with Marianne Akian, LeRoy Beasley, Stéphane Gaubert, and Yaroslav Shitov.

3.6 Constructing Extended Formulations for Stable Set Polytopes via Decomposition Rules

Kanstantsin Pashkovich (University of Padova, IT)

License  Creative Commons BY 3.0 Unported license
 © Kanstantsin Pashkovich
 Joint work of Conforti, Michele; Gerards, Bert

We develop decomposition/composition tools for describing stable set polytopes as polynomially sized linear programs. Some of these tools are well-known but need some extra work to yield polynomial “decomposition schemes”. We apply the tools to graphs that contain no even hole and no cap.

3.7 On lower bounds for extended formulations

Sebastian Pokutta (Universität Erlangen-Nürnberg, DE)

License © Creative Commons BY 3.0 Unported license
© Sebastian Pokutta

Joint work of Braun, Gabor; Pokutta, Sebastian

Main reference G. Braun, S. Pokutta, “Common information and unique disjointness,” ECCC TR13-056, 2013.

URL <http://eccc.hpi-web.de/report/2013/056/>

Communication complexity and information theoretic approaches have been at the core of many recent lower bound proofs for the size of extended formulations of certain polytopes. One of the most important problems in this context is the unique disjointness problem which is closely related to the extension complexity of the correlation polytope and (a natural encoding of) the clique problem. We will provide an overview of recent results by Braun, Fiorini, Pokutta and Steurer as well as those of Braverman and Moitra. Based on the BM approach we then present a generalized information theoretic framework which decouples polyhedral/geometric aspects from the underlying combinatorics and opens up several routes for establishing more general lower bounds.

3.8 Which nonnegative matrices are slack matrices?

Richard Robinson (University of Washington, US)

License © Creative Commons BY 3.0 Unported license
© Richard Robinson

Joint work of Gouveia, João; Grappe, Roland; Kaibel, Volker; Pashkovich, Kanstantsin; Robinson, Richard Z.; Thomas, Rekha R.

Main reference J. Gouveia, R. Grappe, V. Kaibel, K. Pashkovich, R.Z. Robinson, R.R. Thomas, “Which Nonnegative Matrices Are Slack Matrices?,” arXiv:1303.5670v1 [math.OA], 2013.

URL <http://arxiv.org/abs/1303.5670v1>

In this note we characterize the slack matrices of cones and polytopes among all nonnegative matrices. This leads to an algorithm for deciding whether a given matrix is a slack matrix. The underlying decision problem is equivalent to the polyhedral verification problem whose complexity is unknown.

3.9 Matrix factorization over semirings

Yaroslav Shitov (Moscow State University, RU)

License © Creative Commons BY 3.0 Unported license
© Yaroslav Shitov

Joint work of Guterman, Alexander; Shitov, Yaroslav

My recent work in matrix theory has been devoted to studying matrix factorizations over nonnegative numbers, and also over tropical and other semirings. The problem of factoring tropical matrices is useful in tropical geometry as well as finds applications in optimization and phylogenetics. Thus a question of describing the computational complexity of tropical factorization arises. That question has been answered recently, and it has been shown that tropical factorization is hard. I also studied different problems on nonnegative matrix factorizations, and a number of techniques from semiring linear algebra allowed to make some progress on those problems. The behavior and computational complexity of rank functions on tropical matrices have been discussed also in our recent joint paper with Alexander Guterman.

3.10 On the extension complexity of combinatorial polytopes

Hans Raj Tiwary (University of Brussels, BE)

License  Creative Commons BY 3.0 Unported license
© Hans Raj Tiwary

Joint work of Avis, David; Tiwary, Hans Raj

Main reference D. Avis, H.R. Tiwary, “On the extension complexity of combinatorial polytopes,”
arXiv:1302.2340v2 [math.CO], 2013.

URL <http://arxiv.org/abs/1302.2340v2>

In this talk I will describe a lifting argument to show exponential extension complexity for a number of NP-complete problems including subset-sum and three dimensional matching. We obtain a relationship between the extension complexity of the cut polytope of a graph and that of its graph minors. Using this we are able to show exponential extension complexity for the cut polytope of a large number of graphs, including those used in quantum information and suspensions of cubic planar graphs.

4 Problem discussion sessions, and subsequent developments

Here we report on the status of questions which were presented during the Problem Sessions. Near the end of this section, we discuss developments on problems, which were not presented during the Problem Sessions, but discussed during the seminar.

4.1 Real vs. rational nonnegative rank

Presented by Dirk Oliver Theis; problem based on a problem by Cohen and Rothblum from 1991. Give a non-trivial bound for $\text{rk}_{\mathbb{Q}_+}(A) - \text{rk}_{\mathbb{R}_+}(A)$! For example, is it true that for every rational nonnegative matrix A we have $\text{rk}_{\mathbb{Q}_+}(A) \leq \text{rk}_{\mathbb{R}_+}(A) + 1$?

The original question asks for equality between the two ranks, but currently no non-trivial bounds for $\text{rk}_{\mathbb{Q}_+}(A) - \text{rk}_{\mathbb{R}_+}(A)$ or even $\text{rk}_{\mathbb{Q}_+}(A) / \text{rk}_{\mathbb{R}_+}(A)$ are known.

In the discussion, Nati Linial pointed to Micha Perles discovery of non-rational polytopes, and the studies by Richter-Gebert and others of the realization spaces of polytopes.

A related problem is the following.

Complex vs. real positive semidefinite rank. One can ask a similar question for the positive semidefinite rank. The positive semidefinite rank over \mathbb{R} of a matrix $A \in \mathbb{R}^{m \times n}$ is the minimal r such that there are $B_i \in \mathbb{R}^{r \times r}$ for $i = 1, \dots, m$ and $C_j \in \mathbb{R}^{r \times r}$ for $j = 1, \dots, n$ such that $A(i, j) = \text{Tr}(B_i^* C_j)$. The positive semidefinite rank over \mathbb{C} is defined analogously with $B_i, C_j \in \mathbb{C}^{r \times r}$. Is the positive semidefinite rank over \mathbb{R} equal to positive semidefinite rank over \mathbb{C} ? This is the simplest in a family of questions: The same needs to be asked for rational vs. real positive semidefinite rank. In the discussion, Nicolas Gillis pointed out that this question also must be settled for the copositive ranks. In this case one looks for a factorization $A(i, j) = \text{Tr}(B_i^* C_j)$ where each B_i is copositive and each C_j is completely positive.

4.2 Square-root rank

Proposed by Richard Robinson. Given nonnegative matrix $M \in \mathbb{R}_+^{p \times q}$, we say that $A \in \mathbb{R}^{p \times q}$ is a Hadamard square-root if $(A)_{ij}^2 = M_{ij}$. Define $\text{rk}_{\sqrt{\cdot}}(M)$ as the minimum rank of A such that A is Hadamard square-root of M . This is equivalent to a version of the positive semidefinite rank where the matrices in the factorization are constrained to have rank 1. Hence, $\text{rk}_{PSD}(M) \leq \text{rk}_{\sqrt{\cdot}}(M)$ holds. For vertex-facet slack matrices of polytopes, $\text{rk}(S) \leq \text{rk}_{\sqrt{\cdot}}(S)$. A large number of observations have led Richard to conjecture the following.

If $\text{rk}(S) = \text{rk}_{\sqrt{\cdot}}(S)$ holds for the slack matrix S of a polytope, then the entries in the hadamard square-roots in $\text{rk}_{\sqrt{\cdot}}(S)$ can be taken to be nonnegative.

In the discussion, Samuel Fiorini suggested to look specifically at the matrix $M_{ab} = (1 - a^T b)^2$.

4.3 Positive semidefinite rank of matrices defined by polynomials

Proposed by Troy Lee. What is the positive semidefinite rank of the matrix

$$M(x, y) = (x^t y - 1)(x^t y - 2),$$

where x, y range over all $\{0, 1\}^n$?. The motivation is that such a matrix M is a submatrix of the slack matrix of the correlation polytope. One can define a whole family of submatrices of the slack matrix of the correlation polytope by taking a quadratic polynomial p which is nonnegative on nonnegative integers, and letting $M(x, y) = p(|x \cap y|)$. In the discussion, Sam Fiorini pointed out that to show a strong lower bound on M one would have to focus on more than just the entries of the matrix which take values in some small set. He also mentioned that this matrix can be approximated by one that does have low positive semidefinite rank, namely the matrix $N(x, y) = (x^t y - 3/2)^2$.

A toy version of this problem asks about the positive semidefinite rank of the n -by- n matrix $M_n(i, j) = (i - j - 1)(i - j - 2)$. Seminar participant João Gouveia [8] answered a question of Lee and Theis [11] by showing that the psd rank of M_n goes to infinity with n .

4.4 A query complexity problem

Proposed by Raghav Kulkarni. For $f: \{0, 1\}^n \rightarrow \{0, 1\}$ and $z \in \{0, 1\}^n$, we say that i th bit of z is *sensitive* if $f(z_1, \dots, \bar{z}_i, \dots, z_n) \neq f(z_1, \dots, z_i, \dots, z_n)$. Let $s(f, z)$ be the number of sensitive bits of z and $s(f) = \max_z \{s(f, z)\}$ the maximum number of sensitive bits of any argument. These concepts arise in the context of decision tree complexity. For $x, y \in \{0, 1\}^n$ let $f(x, y)$ be a 2-parameter function, and let $M_f(x, y) = f(x, y)$ be the corresponding matrix. Raghav conjectures that $\log \text{rk}_{\mathbb{R}}(M_f) \leq \text{poly}(s(f))$.

In the following discussion, Hartmut Klauck asked about block sensitivity and Raghav said the conjecture is true with sensitivity replaced by block sensitivity. Hartmut also suggested easier versions of the conjecture where, for example, the rank is replaced by sign rank which is the minimum rank of a matrix that entrywise agrees with the target matrix in sign. Nati Linial then asked if assuming the log-rank conjecture is true implies anything for this conjecture.

4.5 A rigidity-type question

Proposed by Adi Shraibman. The famous matrix rigidity problem of Valiant asks to explicitly construct matrices with high rank and such that if a constant fraction of the entries are arbitrarily changed, the rank remains high. While probabilistic constructions exist, finding explicit constructions remains a hard open problem.

Consider the following variant of the problem known as discrepancy games. Here you start with an empty $n \times n$ matrix. Two players, Balancer (+1) and Unbalancer (−1), take turns assigning entries of the matrix to their associated value. Balancer wants to make all combinatorial rectangles balanced, while Unbalancer wants to make them unbalanced. In this game it is known that Balancer can get ensure an upper bound of $s^{3/4}$ on discrepancy after s rounds.

Here is another variant that is open. In this case we begin with a $\{-1, +1\}$ valued matrix with discrepancy $n^{3/2}$. Say a Hadamard matrix. Balancer picks certain +1's. Unbalancer picks certain −1's. Over the course of the game, can Balancer maintain discrepancy to be less than $s^{3/4}$ after s moves? In this variation you cannot rely on strategy stealing.

4.6 Extension complexity of stable set polytopes of split-graph-free perfect graphs

Proposed by Samuel Fiorini. A split graph is a graph in which the vertices can be partitioned into a clique and an independent set. Let H be a split graph, and consider the class of all graphs G not containing H as an induced subgraph. What is the extension complexity of the stable set polytopes of this class of graphs? This problem is motivated by a recent result of Bousquet, Lagoutte, and Thomassé [2]. They provide a certificate of size $O(\log(n))$ proving that a clique and an independent set do not intersect for H -free graphs.

4.7 Matrices with low non-negative rank are a low-dimensional subset in the manifold of rank-bounded matrices

It is an easy fact that the nonnegative rank is semicontinuous: If A_j tends to A , then $\text{rk}_+(A) \leq \liminf \text{rk}_+(A_j)$. Let n, k be nonnegative integers such that $3 \leq k \ll n$, and consider the set of $n \times n$ matrices

$$\{A \in \mathbb{R}_+^{n \times n} \mid \text{rk } A = k, \text{rk}_+ A \leq n - 1\}.$$

Does this set contain interior points within the manifold $\{A \in \mathbb{R}_+^{n \times n} \mid \text{rk } A = k\}$ of nonnegative rank- k matrices? (The “ $n - 1$ ” is somewhat arbitrary, and should be replaced by an appropriate function of n .) This question asks for the dimension of the set of matrices of “small” nonnegative rank as a semialgebraic subset of the variety of rank- k $n \times n$ matrices. In the discussions, people expressed that the intuitively obvious answer to the question is yes. But a recent result of Yaroslav Shitov proves that, for $k = 3$, *every* such matrix A has nonnegative rank at most $6n/7$ [14]. However, the question with “ $n - 1$ ” may still be true for large k .

4.8 Fooling-sets and rank

Let A be an $n \times n$ matrix over a field \mathbb{K} satisfying $A_{kk} = 1$ for all k and $A_{k\ell}A_{\ell k} = 0$ whenever $k \neq \ell$. Dietzfelbinger et al. (1996) proved that $n \leq \text{rk}(A)^2$. The question raised by Dietzfelbinger et al. is whether this bound is asymptotically ($n \rightarrow \infty$) tight.

This problem was fully settled by Friesen and Theis in the case of nonzero characteristic shortly before the seminar [7]. In summary, the following is known.

	characteristic of \mathbb{K}		
	0	2	≥ 3
A 0/1 entries	open	tight	open
A arbitrary entries	open	tight	tight

The major open question is when the characteristic is 0 and the entries are arbitrary. At the time of the seminar, the best separation was by Klauck and de Wolf [10] who gave an example where $\text{rk}(A) \leq n^{0.613\dots}$ with integral entries of small modulus. After the seminar, in the case of characteristic zero, Troy Lee was able to improve the best known bound to $\text{rk}(A) \leq n^{0.594\dots}$ with a method that warrants future investigation.

4.9 Polygons — or, more generally, rank-3 matrices

A problem which was discussed intensely during the seminar was the extension complexity of polygons, where, at the time of the seminar, a lower bound of $\Omega(\sqrt{n})$ was known, and the trivial upper bound n .

The following problem by Beasley & Laffey [1] is both more general and more specific: Given any sub-semiring S of \mathbb{R}_+ and $n \geq 6$, is there a matrix $A \in \mathbb{M}_{n \times n}(S)$ such that $\text{rk} A = 3$ and $\text{rk}_S(A) = n$? At the time of the seminar, this was open even for $S = \mathbb{R}_+$. Following the seminar, participant Yaroslav Shitov [14] has settled this problem for the semiring \mathbb{R}_+ : every rank-3 nonnegative $n \times n$ matrix with $n \geq 7$ has nonnegative rank at most $6n/7$.

4.10 Euclidean distance matrices

Do “generic”/“random” Euclidean distance matrices¹ have full nonnegative rank?²

For $d = 1$, Shitov’s above mentioned result gives a negative answer to the question.

4.11 How does nonnegative rank behave under tensor products?

This question is interesting on its own, and also has application to communication complexity. A very strong conjecture, discussed at the workshop, would be that the nonnegative rank is multiplicative, as the rank is. That is, that $\text{rk}_+(A \otimes B) = \text{rk}_+(A) \text{rk}_+(B)$.

Collaboration between two seminar participants, Nicolas Gillis and Hamza Fawzi showed that this strong conjecture is false. Specifically, Nicolas wrote software for computing the

¹ A d -dimensional, Euclidean distance matrix of size n is defined by points x_1, \dots, x_n in d -dimensional Euclidean space. Its entries are $(\|x_k - x_\ell\|)_{k,\ell}$.

² An affirmative proof of this for $d = 1$ by Lin and Chu (2011) is fatally flawed.

nonnegative rank of small-size matrices, and using this software Hamza was able to disprove the conjecture.

4.12 Vertex/facet slack matrices vs. general nonnegative matrices

Some lower bounds can be proved for matrices which arise from vertex/facet slack matrices of polytopes. It is an open question whether some of the bounds behave fundamentally different in the case of vertex/facet slack matrices (cf. e.g., 4.2). As a first step towards resolving this type of questions, a linear-algebraic characterization of these matrices was obtained by João Gouveia, Richard Robinson, and Rekha Thomas, and presented during the seminar. It turned out that Volker Kaibel, Roland Grappe, and Kanstantsin Pashkovich had a similar approach. Their results were combined in the recent paper [9].

4.13 Extensions/factorizations over the positive semidefinite cone

In the year preceding the seminar, matrix factorizations over other cones than $(\mathbb{R}_+)^r$ have gained importance, specifically the cone of positive semidefinite matrices. From the combinatorial optimization point of view, a very basic question there is, whether there exist polytopes whose extension complexity is exponential in the dimension — the same question for $(\mathbb{R}_+)^r$ was settled by Rothvoß [13]. In a very recent paper, seminar participant Sebastian Pokutta, together with two coauthors, Jop Briët and Daniel Dadush, have answered that question in the affirmative [4].

In the context of factorizations over other cones, recently, the conference participants Samuel Fiorini and Hans Raj Tiwary [6] have observed that a 2012 theorem by Alexander Maksimenko [12] together with a result by Samuel Burer [5] implies that every 0/1 polytope whose vertex set can be described by a polynomial predicate has a polynomial sized copositive extension.

5 Conclusion

A natural approach to solving hard combinatorial optimization problem is to give a formulation as a linear program and solve it using standard techniques. An important topic initiated by Yannakakis is to investigate the size of extended formulations of optimization problems.

Recently the theory of representing hard optimization problems via extended formulations has seen much progress. Techniques from communication complexity and matrix theory have been essential to investigate how large extended formulations need to be, finally improving on Yannakakis' seminal results.

The seminar brought together researchers from the areas of optimization theory, complexity theory, and matrix theory, to further collaboration on these and newly emerging topics. Exciting progress was reported on proving lower bounds for the nonnegative rank, on the hardness of approximation using extended formulations, and on new notions of matrix ranks.

For the future we hope that similar progress will soon be made on the topic of using semidefinite programming to solve hard optimization problems. Intriguingly this problem is connected to quantum communication complexity.

Dagstuhl provided a wonderful environment for many informal discussions as well as talks, plus an exciting open problems session. The opportunity to have this seminar was well appreciated by the participants, many of them who were new to the center.

References

- 1 LeRoy B. Beasley and Thomas J. Laffey. Real rank versus nonnegative rank. *Linear Algebra Appl.*, 431(12):2330–2335, 2009.
- 2 N. Bousquet, A. Lagoutte, and S. Thomassé. Clique versus independent set. Technical Report arXiv:1301.2474, arXiv, 2013.
- 3 G. Braun and S. Pokutta. Common information and unique disjointness. Technical Report TR13-056, ECCO, 2013.
- 4 J. Briët, D. Dadush, and S. Pokutta. On the existence of 0/1 polytopes with high semidefinite extension complexity. Technical Report arXiv:1305.3268, arXiv, 2013.
- 5 S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *math. Program., Ser. A*, 120:479–495, 2009.
- 6 S. Fiorini and H. Tiwary. Personal Communication, 2013.
- 7 Mirjam Friesen and Dirk Oliver Theis. Fooling sets and rank in nonzero characteristic. arXiv:1305.2468 (accepted for EuroComb’13), 2013.
- 8 J. Gouveia. Personal Communication, 2013.
- 9 J. Gouveia, R. Grappe, V. Kaibel, K. Pashkovich, R. Robinson, and R. Thomas. Which nonnegative matrices are slack matrices. Technical Report arXiv:1303.5670, arXiv, 2013+.
- 10 Hartmut Klauck and Ronald de Wolf. Fooling one-sided quantum protocols. In *Proceedings of the 30th Symposium on Theoretical Aspects of Computer Science*, 2013.
- 11 T. Lee and D. O. Theis. Support based bounds for positive semidefinite rank. Technical Report arXiv:1203.3961, arXiv, 2012.
- 12 A.N. Maksimenko. An analog of the Cook theorem for polytopes. *Russian Mathematics*, 86(8):28–34, 2012.
- 13 Thomas Rothvoß. Some 0/1 polytopes need exponential size extended formulations. arXiv:1105.0036, 2011.
- 14 Y. Shitov. An upper bound for the nonnegative rank. Technical Report arXiv:1303.1960, arXiv, 2013.

Participants

- LeRoy B. Beasley
Utah State University, US
- Hamza Fawzi
MIT – Cambridge, US
- Samuel Fiorini
University of Brussels, BE
- Anna Gál
University of Texas at Austin, US
- Nicolas Gillis
UC Louvain-la-Neuve, BE
- Francois Glineur
UC Louvain, BE
- Joao Gouveia
University of Coimbra, PT
- Alexander Guterman
Moscow State University, RU
- Volker Kaibel
Universität Magdeburg, DE
- Stephen Kirkland
Nat. University of Ireland, IE
- Hartmut Klauck
Nanyang TU – Singapore, SG
- Raghav Kulkarni
National Univ. of Singapore, SG
- Thomas Laffey
University College – Dublin, IE
- Troy Lee
National Univ. of Singapore, SG
- Lek-Heng Lim
University of Chicago, US
- Nathan Linial
The Hebrew University of
Jerusalem, IL
- Pablo Parrilo
MIT – Cambridge, US
- Kanstantsin Pashkovich
University of Padova, IT
- Sebastian Pokutta
Univ. Erlangen-Nürnberg, DE
- Richard Robinson
University of Washington, US
- Yaroslav Shitov
Moscow State University, RU
- Adi Shraibman
Academic College of Tel Aviv
Yafo, IL
- Dirk Oliver Theis
University of Tartu, EE
- Rekha R. Thomas
University of Washington, US
- Hans Raj Tiwary
University of Brussels, BE
- Stefan Weltge
Universität Magdeburg, DE



Analysis, Test and Verification in The Presence of Variability

Edited by

Paulo Borba¹, Myra B. Cohen², Axel Legay³, and Andrzej Wąsowski⁴

1 Federal University of Pernambuco – Recife, BR, phmb@cin.ufpe.br

2 University of Nebraska – Lincoln, US, myra@cse.unl.edu

3 University of Liège, BE, alegay@irisa.fr

4 IT University of Copenhagen, DK, wasowski@itu.dk

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13091 “Analysis, Test and Verification in The Presence of Variability”. The seminar had the goal of consolidating and stimulating research on analysis of software models with variability, enabling the design of variability-aware tool chains. We brought together 46 key researchers from three continents, working on quality assurance challenges that arise from introducing variability, and some who do not work with variability, but that are experts in their respective areas in the broader domain of software analysis or testing research. As a result of interactions triggered by sessions of different formats, the participants were able to classify their approaches with respect to a number of dimensions that helped to identify similarities and differences that have already been useful to improve understanding and foster new collaborations among the participants.

Seminar 24. February to 1. March, 2013 – www.dagstuhl.de/13091

1998 ACM Subject Classification D.2.4 Software/Program Verification, D.2.5 Testing and Debugging, D.2.13 Reusable Software, D.3.1 Formal Definitions and Theory, F.3.1 Specifying and Verifying and Reasoning about Programs, F.3.2 Semantics of Programming Languages

Keywords and phrases Verification, Program Analysis, Testing, Semantics of Programming Languages, Software Engineering

Digital Object Identifier 10.4230/DagRep.3.2.144

Edited in cooperation with Leopoldo Teixeira

1 Executive Summary

Paulo Borba

Myra B. Cohen

Axel Legay

Andrzej Wąsowski

License © Creative Commons BY 3.0 Unported license
© Paulo Borba, Myra B. Cohen, Axel Legay, and Andrzej Wąsowski

The seminar “Analysis, Test and Verification in The Presence of Variability” that took place at Schloss Dagstuhl from February 24 to March 1, 2013, had the goal of consolidating and stimulating research on analysis of software models with variability, enabling the design of variability-aware tool chains. We brought together 46 key researchers from three continents, working on quality assurance challenges that arise from introducing variability, and some



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Analysis, Test and Verification in The Presence of Variability, *Dagstuhl Reports*, Vol. 3, Issue 2, pp. 144–170
Editors: Paulo Borba, Myra B. Cohen, Axel Legay, and Andrzej Wąsowski



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

who do not work with variability, but that are experts in their respective areas in the broader domain of software analysis or testing research. The participants ranged from those in senior academic positions to successful graduate students. We also enjoyed the presence of several relevant experts from the software development industry.

The seminar included:

1. Invited presentations on state of the art research in SPL testing and verification

The presentations were delivered by experts in variability research. The topics included classifying and unifying product-line analyses, combinatorial interaction testing, model-based testing, analysis of programs with variability and model checking with variability.

Material relevant to the topic of this Dagstuhl was organized in a recent classification by Thüm and coauthors [4]. The Dagstuhl seminar opened with a presentation of this classification, which created a common ontology for later presentations and discussions. This was very helpful for participants who had different areas of expertise.

2. A keynote presentation on the Challenges and Science of Variability

We organized a special keynote shared with the German FOSD meeting, that took place in parallel at the Schloss Dagstuhl facilities. The keynote speaker, Professor Don Batory, called for creating a simple meta-theory identifying and relating the core concepts and properties of variability science, i.e. the body of knowledge created by the community of researchers studying engineering of highly configurable systems. During the workshop, several candidates for the starting point of such theory were mentioned, such as using simple models in constructive logic [2], choice calculus [3] or Clafer [1].

3. A series of presentations on recent results in Variability Analysis

The bulk of the programme was filled with a mixture of research presentations about recent research advances in verification, analysis and test of software with variability. This function of the seminar was particularly important, as the usual dissemination outlets for these contributions are often disjoint – much of the work is normally presented in domain specific publication channels devoted to only test, verification or programming languages. For many participants the seminar created an opportunity to learn about advances at addressing similar problems in the neighboring research communities – an experience that is rarely possible outside of Dagstuhl.

4. A session of student presentations

In order to enrich the presentations by senior researchers with a stream of fresh ideas, we organized a special session devoted to short student presentations. The presenters were selected from the participants of the German FOSD meeting. For many of the students it was a rare opportunity to share their ideas with international authorities in their work area. The topics of these lightning presentations were closely related to the seminar goals and included among others, discussions of experimental evaluation of product line analysis strategies, static analysis, type checking for variability, and performance prediction for configurable systems. The session enabled closer integration between the participants of the two events. Many discussions between the two groups continued throughout the week.

5. Dynamically planned sessions on how to address the challenges, how to transfer knowledge, tools, and benchmarks between research areas

The first session (run by Professor Krzysztof Czarnecki) was devoted to extracting challenges for variability analysis out of industrial requirements. Participants from industry and participants from academia involved in industrial projects provided background on requirements known from projects in avionics, automotive and risk assessment domains. These were further discussed to identify research challenges for future work. The discussions were continued in a breakout session on product lines of safety critical systems. Other breakout sessions included dynamic product lines, generic representation of variability, and testing and modeling variability.

Overall, a core set of techniques were discussed at this seminar which include program analysis, model checking, type checking, and testing. We believe that the seminar fruitfully mixed computer science and software engineering researchers from several research sub-domains, allowing them to derive interesting basic research problems stemming from practical needs all related to how variability impacts their respective domains, with the sub-goal of inspiring the use of the latest research advances in software analysis technology to advance variability management tools.

Results

The different kinds of interactions offered by the seminar helped the participants to relate work covering different aspects in a number of dimensions such as:

1. An overall approach to thinking about variability, as defined by Thüm's classification [4] of analysis into product based, family based, feature based and hybrids;
2. Core techniques: testing, verification, refactoring, model checking, static analysis;
3. Mechanisms for representing variability: if-defs, deltas, generic representation, etc.;
4. Application domains;
5. The nature of variability: static product lines, dynamic product lines, configurable systems.

The seminar also produced a bibliography of core readings on the topic, that can enable new graduate students to engage more quickly in this area of research.

Trying to classify approaches with respect to these dimensions helped to identify similarities and differences among different techniques (static analysis, model checking, testing, and verification). This, in turn, might trigger new collaborations and research results. The presentations and the ad-hoc discussion sessions helped people to clarify differences and similarities among configurable systems and dynamic and static product lines, with similar consequences to the ones described above. More generally, of course, the Dagstuhl provided the benefit of mixing young and experienced researchers, from different countries and research areas.

An informal survey among a handful of participants has shown that each of them have started 2-3 new collaborations as a result of the seminar. These collaborations took the form of initiated research papers, mutual research visits, or student exchanges. In one anecdotal case, a researcher started a collaboration with a colleague sitting in the same corridor at his home university— but apparently one had to meet in Dagstuhl to enable the exchange of ideas. We can thus expect a new wave of research results in this area to flourish about a year from the seminar time. Because of this success, we intend to organize a follow up event in

several years, be it under the Schloss Dagstuhl programme or under some other appropriate venue.

References

- 1 Kacper Bąk, Krzysztof Czarnecki, and Andrzej Wąsowski. Feature and Meta-Models in Clafer: Mixed, Specialized, and Coupled. In *Proc. of the 3rd Int'l Conf. on Software Language Engineering (SLE'10)*, LNCS, Vol. 6563, pp. 102–122, Springer, 2011. DOI: 10.1007/978-3-642-19440-5_7.
- 2 Benjamin Delaware, William R. Cook, and Don S. Batory. Product lines of theorems. In *Proc. of the 2011 ACM Int'l Conf. on Object-oriented Programming Systems, Languages, and Applications (OOPSLA'11)*, pp. 595–608, ACM, 2011. DOI: 10.1145/2048066.2048113.
- 3 Martin Erwig and Eric Walkingshaw. The Choice Calculus: A Representation for Software Variation. *ACM Trans. Softw. Eng. Methodol.*, Vol. 21, Issue 1, pp. 6:1–6:27, 2011. DOI: 10.1145/2063239.2063245.
- 4 Thomas Thüm, Sven Apel, Christian Kästner, Martin Kuhlemann, Ina Schaefer, and Gunter Saake. Analysis Strategies for Software Product Lines. Technical Report FIN-004-2012, School of Computer Science, University of Magdeburg, April 2012. http://www.cs.uni-magdeburg.de/inf_media/downloads/forschung/technical_reports_und_preprints/2012/04_2012.pdf.

2 Table of Contents

Executive Summary

Paulo Borba, Myra B. Cohen, Axel Legay, and Andrzej Wąsowski 144

Overview of Talks


Family and Sampling-based Reliability Analysis in Dynamic Software Product Line: the Body Area Network Case <i>Vander Alves</i>	150
Classifying and Unifying Product-Line Analyses <i>Sven Apel</i>	150
Software Product Lines in Clafer <i>Kacper Bąk</i>	151
Analyzing Software Product Lines in Minutes instead of Years <i>Eric Bodden</i>	151
State of The Art in Analysis of Programs with Variability <i>Eric Bodden</i>	152
Intraprocedural Dataflow Analysis for Software Product Lines <i>Claus Brabrand</i>	152
If not interfaces, then views <i>Dave Clarke</i>	153
On a Feature-Oriented Characterization of Exception Flows in Software Product Lines <i>Roberta Coelho</i>	153
ProVeLines: a Product Line of Model Checkers for Software Product Lines and some of the Theory behind it <i>Maxime Cordy</i>	154
Toward the Systematic Derivation of Variational Program Analyses <i>Martin Erwig</i>	154
Topologically configurable systems as product families <i>Alessandro Fantechi</i>	155
Patterns in Configuration Dependence <i>Brady J. Garvin</i>	155
Making Software Product Line Evolution Safer <i>Rohit Gheyi</i>	156
Reuse of Formal Verification Proofs By Abstract Method Calls <i>Reiner Hähnle</i>	156
Complete and Reproducible Applications of SPL Testing <i>Martin Fagereng Johansen</i>	157
Analyzing the #ifdef hell with TypeChef – Or the quest for realistic subjects in product-line analysis <i>Christian Kästner</i>	157

Is medical underwriting knowledge a field for verification in the presence of variability?	
<i>Kim Lauenroth</i>	161
Scientific Workflows: Eternal Components, Changing Interfaces, Varying Compositions	
<i>Tiziana Margaria</i>	161
Compositional Verification of Software Product Lines	
<i>Jean-Vivien Millo</i>	162
Feature Maintenance with Emergent Interfaces	
<i>Marcio Ribeiro</i>	162
Delta-oriented Regression-based Testing of Software Product Lines	
<i>Ina Schaefer</i>	163
Model-Based Testing of Software Product Lines	
<i>Holger Schlingloff</i>	163
Reuse of Test Cases for Model-Based Development of Software Product Lines	
<i>Holger Schlingloff</i>	164
Test Case Prioritization Criteria for Software Product Lines	
<i>Sergio Segura</i>	164
Composing Variable Components Considering Interaction Behavior – A Problem Statement	
<i>Vanessa Stricker</i>	164
Safe Evolution of Software Product Lines and Communities	
<i>Leopoldo Teixeira</i>	165
Product-Line Verification with Contracts	
<i>Thomas Thüm</i>	166
Evaluating Dataflow Analysis for Software Product Lines	
<i>Tarsis Tolêdo</i>	167
Inferring Variational Types for Variational Programs	
<i>Eric Walkingshaw</i>	167
Combinatorial Interaction Testing	
<i>Cemal Yilmaz</i>	168
Effective Test Execution for Software Product Lines	
<i>Sabrina de Figueirêdo Souto</i>	168
Modelling, analysing and verifying variability by means of Modal Transition Systems	
<i>Maurice H. ter Beek</i>	169
Participants	170

3 Overview of Talks

3.1 Family and Sampling-based Reliability Analysis in Dynamic Software Product Line: the Body Area Network Case

Vander Alves (University of Brasilia, BR)

License  Creative Commons BY 3.0 Unported license
© Vander Alves

Demographic and social changes have increased the number of elderly people living alone. Many of these need continuous medical assistance, yet it is not sustainable to have dedicated medical professional for each of them. As a result, automated support has been proposed, in particular, Body Area Network, in which a person goes about his or her daily activities at home or outdoors, but wears sensors monitoring vital signs and providing emergency detection and prevention. Such systems often have to reconfigure themselves based on some context change such as the persons' medical situation to meet a new and more suitable quality goal for that new situation. However, current approaches provide limited support for reliability-aware dynamic adaptation. Accordingly, we explore how family- and sampling-based analysis in Dynamic Software Product Line (DSPL) support reliability-aware dynamic adaptation. First, we present a domain reliability model relying on a state machine whose transitions are medical events (e.g., fall, stroke) and states are target reliability goals, prompting a reconfiguration to meet them. Second, the reliability of any given configuration is measured by a single function over the features of the DSPL. This function is derived from a family-based analysis leveraging a parametric discrete time Markov chain model representing the reliability of the DSPL. Lastly, the configuration space is searched in a bounded product analysis to find suitable configurations meeting reliability goals.

3.2 Classifying and Unifying Product-Line Analyses

Sven Apel (Universität Passau, DE)

License  Creative Commons BY 3.0 Unported license
© Sven Apel

Joint work of von Rhein, Alexander; Apel, Sven; Kästner, Christian; Thüm, Thomas; Schaefer, Ina
Main reference A. von Rhein, S. Apel, C. Kästner, Thomas Thüm, and Ina Schaefer, "The PLA Model: On the Combination of Product-Line Analyses," in Proc. of the 7th Int'l Workshop on Variability Modelling of Software-intensive Systems (VaMoS'13), pp. 73–80, ACM, 2013.
URL <http://dx.doi.org/10.1145/2430502.2430522>
URL <http://www.infosun.fim.uni-passau.de/publications/docs/RAK+13vamos.pdf>

Product-line analysis has received considerable attention in the last decade. As it is often infeasible to analyze each product of a product line individually, researchers have developed analyses, called variability-aware analyses, that consider and exploit variability manifested in a code base. Variability-aware analyses are often significantly more efficient than traditional analyses, but each of them has certain weaknesses regarding applicability or scalability. We present the Product-Line-Analysis model, a model for the classification and comparison of existing analyses, including traditional and variability-aware analyses, and lay a foundation for formulating and exploring further, combined analyses. This talk is based on a number of previous publications [1, 2, 3].

References

- 1 Alexander von Rhein, Sven Apel, Christian Kästner, Thomas Thüm, and Ina Schaefer. The PLA Model: On the Combination of Product-Line Analyses. In *Proceedings of the International Workshop on Variability Modelling of Software-intensive Systems (VaMoS)*, pages 73–80. ACM, January 2013.
- 2 Christian Kästner and Sven Apel. Feature-Oriented Software Development. In *Generative and Transformational Techniques in Software Engineering IV*, volume 7680 of *Lecture Notes in Computer Science*, pages 346–382. Springer-Verlag, January 2013.
- 3 Thomas Thüm, Sven Apel, Christian Kästner, Martin Kuhlemann, Ina Schaefer, and Gunter Saake. Analysis Strategies for Software Product Lines. Technical Report FIN-004-2012, School of Computer Science, University of Magdeburg, April 2012.

3.3 Software Product Lines in Clafer

Kacper Bąk (University of Waterloo, CA)

License  Creative Commons BY 3.0 Unported license
© Kacper Bąk

Joint work of Bąk, Kacper; Czarnecki, Krzysztof; Wąsowski, Andrzej; Antkiewicz, Michal; Diskin, Zinovy; Liang, Jimmy; Olaechea, Rafael; Murashkin, Alexandr

Main reference K. Bąk, K. Czarnecki, A. Wąsowski, “Feature and Meta-Models in Clafer: Mixed, Specialized, and Coupled”, in Proc. of the 3rd Int’l Conf. on Software Language Engineering (SLE’10), LNCS, Vol. 6563, pp. 102–122, Springer, 2011.

URL http://dx.doi.org/10.1007/978-3-642-19440-5_7

URL <http://clafer.org>

Clafer is a lightweight modeling language for modeling and analysis of software product lines. The talk presents Clafer and showcases its features, such as feature modeling, the constraint language, staged configuration, partial instances, and multi-objective optimization. It also discusses two controversial design choices: concept unification and type-instance unification.

3.4 Analyzing Software Product Lines in Minutes instead of Years

Eric Bodden (TU Darmstadt, DE)

License  Creative Commons BY 3.0 Unported license
© Eric Bodden

Joint work of Bodden, Eric; Toledo, Tarsis; Ribeiro, Marcio; Brabrand, Claus; Borba, Paulo; Mezini, Mira

Main reference E. Bodden, T. Tolédo, M. Ribeiro, C. Brabrand, P. Borba, M. Mezini, “SPL^{LIFT} – Statically Analyzing Software Product Lines in Minutes Instead of Years”, in Proc. of the ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI’13), ACM, 2013.

URL <http://dx.doi.org/10.1145/2491956.2491976>

URL <http://www.bodden.de/pubs/bmb+13spllift.pdf>


URL <http://www.bodden.de/2013/02/18/pldi-spllift/>

A software product line (SPL) encodes a potentially large variety of software products as variants of some common code base. Up until now, re-using traditional static analyses for SPLs was virtually intractable, as it required programmers to generate and analyze all products individually. In this work, however, we show how an important class of existing inter-procedural static analyses can be transparently lifted to SPLs. Without requiring programmers to change a single line of code, our approach SPL^{LIFT} automatically converts any analysis formulated for traditional programs within the popular IFDS framework for inter-procedural, finite, distributive, subset problems to an SPL-aware analysis formulated in the IDE framework, a well-known extension to IFDS. Using a full implementation based on

Soot, CIDE and JavaBDD, we show that with SPL^{LIFT} one can reuse IFDS-based analyses without changing a single line of code. Through experiments using three static analyses applied to four Java-based product lines, we were able to show that our approach produces correct results and outperforms the traditional approach by several orders of magnitude.

3.5 State of The Art in Analysis of Programs with Variability

Eric Bodden (TU Darmstadt, DE)

License  Creative Commons BY 3.0 Unported license
© Eric Bodden

In this talk I will explain general principles behind static data-flow analysis, how data-flow analysis has been used to decide interesting properties about software product lines, and recent efforts on trying to lift existing static program analysis to software-product-line analyses.

3.6 Intraprocedural Dataflow Analysis for Software Product Lines

Claus Brabrand (IT University of Copenhagen, DK)

License  Creative Commons BY 3.0 Unported license
© Claus Brabrand

Joint work of Brabrand, Claus; Ribeiro, Márcio; Tolêdo, Társis; Winther, Johnni; Borba, Paulo
Main reference C. Brabrand, M. Ribeiro, T. Tolêdo, J. Winther, P. Borba, “Intraprocedural Dataflow Analysis for Software Product Lines”, *Transactions on Aspect-Oriented Software Development X*, Vol. 10, pp. 73–108, LNCS, Vol. 7800, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-36964-3_3

Software product lines (SPLs) developed using annotative approaches such as conditional compilation come with an inherent risk of constructing erroneous products. For this reason, it is essential to be able to analyze such SPLs. However, as dataflow analysis techniques are not able to deal with SPLs, developers must generate and analyze all valid products individually, which is expensive for non-trivial SPLs.

In this talk, we demonstrate how to take any standard intraprocedural dataflow analysis and automatically turn it into a feature-sensitive dataflow analysis in five different ways where the last is a combination of the other four. All analyses are capable of analyzing all valid products of an SPL without having to generate all of them explicitly.


We have implemented all analyses using SOOT’s intraprocedural dataflow analysis framework and experimentally evaluated four of them according to their performance and memory characteristics on five qualitatively different SPLs. On our benchmarks, the combined analysis strategy is up to almost eight times faster than the brute-force approach.

References

- 1 Claus Brabrand, Márcio Ribeiro, Társis Tolêdo, and Paulo Borba. Intraprocedural Dataflow Analysis for Software Product Lines. In *Proceedings of the 11th International Conference on Aspect-oriented Software Development (AOSD 2012)*, pages 13–24. ACM, Potsdam, Germany, 2012.
- 2 Claus Brabrand, Márcio Ribeiro, Társis Tolêdo, Johnni Winther, and Paulo Borba. Intraprocedural Dataflow Analysis for Software Product Lines. In *Transactions on Aspect-Oriented Software Development X*, vol. 10, pages 73–108, Springer, 2013.

3.7 If not interfaces, then views


Dave Clarke (KU Leuven, BE)

License  Creative Commons BY 3.0 Unported license
© Dave Clarke

Features tend to cross-cut a software product line's code base, and thus providing an interface for them is difficult, if not impossible. For many applications, such as design and modelling of SPLs and some analysis tasks, an alternative is makes more sense. The idea is to provide a view of a software product line by abstracting away irrelevant details. Numerous formalisms for SPLs are based on so-called super-imposed variants, wherein all products of the SPL are captured within the same semantic model. Notions of abstraction for these models should be used to produce views of features or feature combinations. This talk advocated a comprehensive study of views, where no sensible notion of interface exists, in order to simplify reasoning about SPLs.

3.8 On a Feature-Oriented Characterization of Exception Flows in Software Product Lines

Roberta Coelho (Federal University of Rio Grande do Norte, BR)

License  Creative Commons BY 3.0 Unported license
© Roberta Coelho

Joint work of Coelho, Roberta; Melo, Hugo; Kulesza, Uira

Main reference H. Melo, R. Coelho, U. Kulesza, "On a Feature-Oriented Characterization of Exception Flows in Software Product Lines," in Proc. of the 26th Brazilian Symp. on Software Engineering (SBES'12), IEEE, 2012.

URL <http://dx.doi.org/10.1109/SBES.2012.15>


The Exception Handling (EH) is a widely used mechanism for building robust systems. In Software Product Line (SPL) context it is not different. As EH mechanisms are embedded in most of mainstream programming languages, we can find exception signalers and handlers spread over code assets associated to common and variable SPL features. When exception signalers and handlers are added to an SPL in an unplanned way, one of the possible consequences is the generation of faulty products (i.e., products on which common or variable features signal exceptions that are mistakenly caught inside the system). This talk reports a first systematic study, based on manual inspection and static code analysis, in order to (i) categorize the possible ways exceptions flow in SPLs, and (ii) analyze its consequences. Fault-prone exception flows were consistently detected during this study; such as flows on which a variable feature signaled an exception a different and unrelated variable feature handled it. The talk is based on some previous publications [1, 2, 3].

References

- 1 Hugo Melo, Roberta Coelho, Uirá Kulesza On a Feature-Oriented Characterization of Exception Flows in Software Product Lines. 26th Brazilian Symposium on Software Engineering (SBES), August 2012.
- 2 Roberta Coelho, Awais Rashid, Uirá Kulesza, Arndt and von Staa, Carlos Lucena Unveiling and taming liabilities of aspects in the presence of exceptions: A static analysis based approach. In *Information Sciences*, 181, no. 13, pages 2700–2720. 2011
- 3 Roberta Coelho, Awais Rashid, Alessandro Garcia, Fabiano Ferrari, Nélio Cacho, Uirá Kulesza, Arndt and von Staa, Carlos Lucena Assessing the impact of aspects on exception flows: An exploratory study. In *Proceedings of ECOOP 2008– European Conference on Object-Oriented Programming*, pages 207–234. 2008.

3.9 ProVeLines: a Product Line of Model Checkers for Software Product Lines and some of the Theory behind it

Maxime Cordy (University of Namur, BE)


License  Creative Commons BY 3.0 Unported license
© Maxime Cordy

Joint work of Cordy, Maxime; Classen, Andreas; Heymans, Patrick; Schobbens, Pierre-Yves; Legay, Axel
Main reference A. Classen, M. Cordy, P.-Y. Schobbens, P. Heymans, A. Legay, J.-F. Raskin, “Featured Transition Systems: Foundations for Verifying Variability-Intensive Systems and their Application to LTL Model Checking,” IEEE Transactions on Software Engineering, 2013.
URL <http://dx.doi.org/10.1109/TSE.2012.86>
URL <http://info.fundp.ac.be/fts>

Software Product Lines (SPLs) are families of similar software products built from a common set of features. As the number of products of an SPL is potentially exponential in the number of its features, the model checking problem is harder than for single software. A practical way to face this exponential blow-up is to reuse common behaviour between products. We previously introduced Featured Transition Systems, a mathematical model from which serves as a basis for efficient SPL model checking techniques. Here, we present ProVeLines, a product line of verifiers for SPLs that incorporates the results of over three years of research on formal verification of SPLs. Being itself a product line, our tool is flexible and extensible, and offers a wide range of solutions for SPL modelling and verification.

3.10 Toward the Systematic Derivation of Variational Program Analyses

Martin Erwig (Oregon State University, US)

License  Creative Commons BY 3.0 Unported license
© Martin Erwig

Joint work of Erwig, Martin; Walkingshaw, Eric
Main reference M. Erwig, E. Walkingshaw, “Variation Programming with the Choice Calculus”, in Proc. of the Int’l Summer School on Generative and Transformational Techniques in Software Engineering IV (GTTSE’11), LNCS, Vol. 7680, pp. 55–100, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-35992-7_2
URL <http://eecs.oregonstate.edu/~erwig/papers/abstracts.html#GTTSE12>

In this presentation I will illustrate some basic principles for the systematic derivation of variational program analyses from non-variational ones. The approach is based on the view of an analysis as a function f that maps programs from an object language L to elements of some type T that represents the possible results of the analysis. The method proceeds in a number of small steps. First, the syntax of L is extended to a language VL that can represent variational programs. Then, in a similar way, the result type T is extended to a type VT to represent variational results. The language extension is realized through the choice calculus, which provides a generic representation for variation in (tree-structured) software artifacts. Because of its generic structure, the choice calculus can essentially be used as a variational type constructor V , and this makes the first two language extension steps systematic. Finally, f is lifted into a variational analysis by employing a number of simple transformations. Here it is the fact that the variational type constructor V is a monad that makes much of the lifting systematic, because the complexity involved in lifting f can be captured in many cases through an application of the monadic bind operation.

3.11 Topologically configurable systems as product families

Alessandro Fantechi (University of Firenze, IT)

License © Creative Commons BY 3.0 Unported license
© Alessandro Fantechi

Main reference A. Fantechi, “Topologically configurable systems as product families”, to appear in the Proc. of the 17th Int’l Software Product Line Conf., Tokyo, August 2013.

We address a category of systems whose deployment requires a configuration according to topological information. Actually, this study is inspired by the case of railway interlocking systems, but gives a general definition of topologically configurable control systems. We consider the application of product line engineering principles to the development of these systems, e.g. by discussing the adoption of different approaches to achieve a flexible configuration of products, that allow factorising most of the design effort, as typical in a product line approach.

Things become more complex when there is a need of analysing the behaviour of such systems, either by testing or by formal verification: the intricate relations between the actual topology controlled by a product and its functional requirements may prevent any attempt to factorise analysis activities.

Indeed, in the cited case of the interlocking systems, the heavy verification and certification activities required by the safety regulations make a large part of the software development costs. Every newly configured product needs to undergo such certification activities, and little is saved from certifications made for previously deployed systems, with significant costs for each new installation.

We will discuss how a product line approach can help, with special focus on formal verification, showing that several research issues are still to be investigated in this direction.

3.12 Patterns in Configuration Dependence

Brady J. Garvin (University of Nebraska – Lincoln, US)

License © Creative Commons BY 3.0 Unported license
© Brady J. Garvin

Joint work of Garvin, Brady J.; Cohen, Myra B.; Dwyer, Matthew B.


For configuration-aware testing and analysis techniques to effectively exploit whitebox knowledge, it is essential that the mapping of configurability to source code be precise. Unfortunately, as systems grow, less and less of the mapping is syntactically explicit; configuration information may flow through non-configuration data and control dependencies, ultimately rendering code dead in seemingly unrelated places. These dependency chains can grow quite long, and the heavyweight analyses needed to track the relevant properties may not scale to the whole-program level. I will give some examples in my talk.

Using a combination of static and dynamic analysis, we have established the configuration dependence in some small subjects exactly, in order to look for common patterns. Following earlier work, we express reachability of a block either as a CNF formula over atoms that represent configuration options, or else the complement of such a formula, and, thus far, the clauses tend to be extremely short (usually one literal, or, very rarely, two), with unique clauses across all basic blocks being few. In addition to some illustrative data, I will also overview results from previous studies that corroborate these findings.

If these observations hold up in other settings, and we can reasonably assume these patterns in larger systems, then we can drastically shrink the family of possible configuration dependence formulae and therefore the analysis effort needed to identify the formula for a particular basic block.

3.13 Making Software Product Line Evolution Safer

Rohit Gheyi (*Federal University of Campina Grande, BR*)

License  Creative Commons BY 3.0 Unported license
© Rohit Gheyi

Developers evolve software product lines (SPLs) manually or using typical program refactoring tools. However, when evolving a product line to introduce new features or to improve its design, it is important to make sure that the behavior of existing products is not affected. Typical program refactorings cannot guarantee that because the SPL context goes beyond code and other kinds of core assets, and involves additional artifacts such as feature models and configuration knowledge. Besides that, in a SPL we typically have to deal with a set of possibly alternative assets that do not constitute a well-formed program. As a result, manual changes and existing program refactoring tools may introduce behavioral changes or invalidate existing product configurations. To avoid that, we propose approaches and implement tools for making product line evolution safer; these tools check whether SPL transformations are refinements in the sense that they preserve the behavior of the original SPL products. This talk is based on some previous publications [1, 2].

References

- 1 Paulo Borba, Leopoldo Teixeira, and Rohit Gheyi. A theory of software product line refinement. *Theoretical Computer Science*, 455:2 – 30, 2012.
- 2 Gustavo Soares, Rohit Gheyi, Tiago Massoni. Automated behavioral testing of refactoring engines. *IEEE Transactions on Software Engineering*, 39: 147–162, 2013.

3.14 Reuse of Formal Verification Proofs By Abstract Method Calls

Reiner Hähnle (*TU Darmstadt, DE*)

License  Creative Commons BY 3.0 Unported license
© Reiner Hähnle

Joint work of Hähnle, Reiner; Schaefer, Ina; Bubel, Richard

Main reference R. Hähnle, I. Schaefer, R. Bubel, “Reuse in Software Verification by Abstract Method Calls,” in Proc. of the 24th Conf. on Automated Deduction (CADE’13), LNCS, Vol. 7898, pp. 300–314, Springer, 2013.

URL http://dx.doi.org/10.1007/978-3-642-38574-2_21

Modern software tends to undergo frequent requirement changes and typically is deployed in many different scenarios. This poses significant challenges to formal software verification, because it is not feasible to verify a software product from scratch after each change. It is essential to perform verification in a modular fashion instead. The goal must be to reuse not merely software artifacts, but also specification and verification effort.

In our setting code reuse is realized by delta-oriented programming, an approach where a core program is gradually transformed by code “deltas” each of which corresponds to a product feature. The delta-oriented paradigm is then extended to contract-based formal

specifications and to verification proofs. As a next step towards modular verification we transpose Liskov's behavioural subtyping principle to the delta world. Finally, based on the resulting theory, we perform a syntactic analysis of contract deltas that permits to automatically factor out those parts of a verification proof that stays valid after applying a code delta. This is achieved by a novel verification paradigm called "abstract verification".

3.15 Complete and Reproducible Applications of SPL Testing

Martin Fagereng Johansen (University of Oslo, NO)

License © Creative Commons BY 3.0 Unported license
© Martin Fagereng Johansen

Joint work of Johansen, Martin Fagereng; Haugen, Øystein; Fleurey, Franck; Carlson, Erik; Endresen, Jan; Wien, Tormod

Main reference M.F. Johansen, Ø. Haugen, F. Fleurey, E. Carlson, J. Endresen, T. Wien, "A Technique for Agile and Automatic Interaction Testing for Product Lines", in Proc. of the 24th IFIP WG 6.1 Int'l Conf. on Testing Software and Systems (ICTSS'12), LNCS, Vol. 7641, pp. 39–54, Springer, 2012.

URL http://dx.doi.org/10.1007/978-3-642-34691-0_5

Product line engineering is an inherently large scale effort; thus, most product lines are closed source, proprietary and not free. Understanding and verifying the performance of product line testing techniques benefits from a complete application of it that can be reproduced freely and easily. As resource material for a recent paper, we did provide two such complete and reproducible applications, available freely; one small application using CVL and model driven engineering, and a second large application with the Eclipse IDEs using the Eclipse plug-in system and ordinary textual programming. A quick overview of these two examples will be given with views on the benefits of the availability of complete and reproducible applications of SPL testing.

3.16 Analyzing the #ifdef hell with TypeChef – Or the quest for realistic subjects in product-line analysis

Christian Kästner (Carnegie Mellon University – Pittsburgh, US)

License © Creative Commons BY 3.0 Unported license
© Christian Kästner

Joint work of Kästner, Christian; Apel, Sven; Berger, Thorsten; Erdweg, Sebastian; Giarrusso, Paolo; Liebig, Joerg; Ostermann, Klaus; von Rhein, Alexander; Rendel, Tillmann

In recent years, work on analysis of configurable systems has exploded. I and many others have investigated how we can make analysis of entire product lines faster, for example type check all configurations of a program annotated with features. The community has come up with many approaches to speed up analyses (type checking, model checking, static analysis, parsing, and others) by orders of magnitude compared to a brute-force approach [20].

We have claimed that analyses of entire product lines are necessary, because there is an exponentially exploding number of configurations. We have claimed that one could not feasibly check every configuration in isolation. We have claimed that industrial product lines typically have hundreds or thousands of configuration options and more configurations than there are atoms in the universe. We have claimed that analysis is critical, because otherwise users, who configure the systems, run into problems late in the development process when problems are expensive to fix.

When we proposed analysis mechanisms, our evaluations did not really align with our claims. For example, in our work on type checking in CIDE [7], we implemented the checks in a research environment and checked three systems with less than 15 configuration options and one system with 42 configuration options—far away from “more than atoms in the universe”. Given the fact that many of our arguments are empirical (SAT solvers, sharing in typical applications), results may not immediately generalize. In fact, analyzable subject systems are rare. The few systems we had, we shared. The graph product line [15] has served the community well as a canonical example, but it’s tiny. MobileMedia [6] is a common candidate but also rather limited in size and generalizability. The collected applications in CIDE, FeatureHouse [1], and the collected feature models in SPLOT [16] are a great start, but all relatively small and mostly stemming from student projects. We knew that there are lot’s of large industrial product lines, but we could not get our hands on them.

With envy, we have been looking at evaluation subjects from the testing community. Real-world large-scale systems, such as MySQL [23] and GCC [4, 22] with hundreds of configuration options tested and bugs found. For feature modeling, the Kconfig model of the Linux kernel turned out to be a great source for insights [17, 3].

To demonstrate that our analyses can scale to real-world problems and find real-world bugs in actual product lines, we searched for larger subject systems. Even though potentially not really product lines, we found that many C systems are highly configurable at compile time through their use of `#ifdef` directives and the C preprocessor. And the good news was that there are many openly available C systems with active developer communities, who might care about our results.

In initial studies, we found that most open-source systems that we looked at contain a massive amount of variability through the preprocessor [12]. Unfortunately, the way that the preprocessor is used (not always but often enough) makes it hard to parse the code without preprocessing it [13]. That meant, if we wanted to analyze that code in a precise way, and after all we intended to perform at least type checking without running into many false negatives, we would need to build our own infrastructure.

We decided to go after the Linux kernel, which was previously already investigated in the community regarding its feature model and dead code [17, 3, 19, 18]. That was the birth of the TypeChef project (initially short for “type checking `#ifdef` code”). Over the following years, we developed a lexer and parser that could actually process unpreprocessed C code in a sound and complete way [9]. On top, we have built various analyses, which by now form an interesting ecosystem. As of writing, TypeChef includes parsers for GNU C and Java, a type system for C, linker checks, control-flow graphs, and intra-procedural data-flow analysis, first steps toward refactoring engines and interpreters and sampling algorithms; and several more are currently planned and part of ongoing work.

TypeChef is now able to handle the x86 architecture of the Linux kernel (9 million lines of code, 6000 configuration options), Busybox (250 thousand lines of code, 800 configuration options) and we are currently looking into several other systems including OpenSSL, openVPN, BerkeleyDB, Apache, ChibiOS, and vim. We have found and reported several configuration-specific type errors already. We share the necessary scaffolding for these projects as well, which makes it easy to use and extend TypeChef and build other analyses.

While ecosystem and community around TypeChef is thriving, TypeChef is not without limitations. As any analysis of C code it struggles with C dialects and extensions. The parser is unnecessarily slow for mere technical reasons and we do not offer unparsing yet. Setting up analysis for a new system is difficult, because build paths, build system, and feature model need to be reverse engineered—a task where the community has helped us a lot with

Linux [2, 17, 18] and a reason why we publish all our scaffolding to setups easier for others.

Overall, I have seen a great spirit in this community that has frequently shared subject systems such as the graph product line [15], MobileMedia [6], and ArgoUML [5], and I have tried to contribute my own with CIDE and now TypeChef. As a community, we should foster and expand this sharing. Project collecting subject systems, such as first SPLOT for feature models and now SPL2go for product-line implementations are great. Currently, I'd be interested in such a repository for product lines with specifications or test cases...

Acknowledgments. TypeChef was only possible through great collaborations across several research groups, including direct contributors to the project and many colleagues who have helped with the infrastructure around it. I deeply appreciate the help of Sven Apel, Thorsten Berger, Sebastian Erdweg, Paolo G. Giarrusso, Steffen Haase, Andy Kenner, Joerg Liebig, Sarah Nadi, Klaus Ostermann, Alexander von Rhein, Tillmann Rendel, and Reinhard Tartler.

Further Reading on TypeChef. An in-depth discussion of the parsing approach and our experience with parsing Linux was published at OOPSLA 2011 [9]. In the context of a variability-aware module system, we discussed type checking and linker checks on the example of Busybox, published at OOPSLA 2012 [10]. A more detailed discussion of the variability-aware lexer (or partial preprocessor) was presented at VaMoS 2011 [8]. More information on the performance of our type system and data-flow analysis can be found in a technical report [14]. A simple variability-aware interpreter for executing test cases was build on top of TypeChef and published at FOSD 2012 [11]. For an overview of variability-aware analysis in general, please refer to the corresponding reports [20, 21] on the following webpage <http://fbsd.net/spl-strategies>.

References

- 1 S. Apel, C. Kästner, and C. Lengauer. Language-independent and automated software composition: The FeatureHouse experience. *IEEE Transactions on Software Engineering (TSE)*, 2012. in press.
- 2 T. Berger, S. She, K. Czarnecki, and A. Wąsowski. Feature-to-code mapping in two large product lines. In *Proc. Int'l Software Product Line Conference (SPLC)*, pages 498–499. Springer-Verlag, 2010.
- 3 T. Berger, S. She, R. Lotufo, A. Wąsowski, and K. Czarnecki. Variability modeling in the real: A perspective from the operating systems domain. In *Proc. Int'l Conf. Automated Software Engineering (ASE)*, pages 73–82. ACM Press, 2010.
- 4 M. B. Cohen, M. B. Dwyer, and J. Shi. Interaction testing of highly-configurable systems in the presence of constraints. In *Proc. Int'l Symp. Software Testing and Analysis (ISSTA)*, pages 129–139. ACM Press, 2007.
- 5 M. V. Couto, M. T. Valente, and E. Figueiredo. Extracting software product lines: A case study using conditional compilation. In *Proc. European Conf. on Software Maintenance and Reengineering (CSMR)*, pages 191–200. IEEE Computer Society, 2011.
- 6 E. Figueiredo et al. Evolving software product lines with aspects: An empirical study on design stability. In *Proc. Int'l Conf. Software Engineering (ICSE)*, pages 261–270. ACM Press, 2008.
- 7 C. Kästner, S. Apel, T. Thüm, and G. Saake. Type checking annotation-based product lines. *ACM Trans. Softw. Eng. Methodol. (TOSEM)*, 21(3):Article 14, 2012.
- 8 C. Kästner, P. G. Giarrusso, and K. Ostermann. Partial preprocessing of C code for variability analysis. In *Proc. Int'l Workshop on Variability Modelling of Software-intensive Systems (VaMoS)*, pages 137–140. ACM Press, 2011.
- 9 C. Kästner, P. G. Giarrusso, T. Rendel, S. Erdweg, K. Ostermann, and T. Berger. Variability-aware parsing in the presence of lexical macros and conditional compilation.

- In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, pages 805–824. ACM Press, Oct. 2011.
- 10 C. Kästner, K. Ostermann, and S. Erdweg. A variability-aware module system. In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*. ACM Press, 2012.
 - 11 C. Kästner, A. von Rhein, S. Erdweg, J. Pusch, S. Apel, T. Rendel, and K. Ostermann. Toward variability-aware testing. In *Proc. GPCE Workshop on Feature-Oriented Software Development (FOSD)*, pages 1–8, 2012.
 - 12 J. Liebig, S. Apel, C. Lengauer, C. Kästner, and M. Schulze. An analysis of the variability in forty preprocessor-based software product lines. In *Proc. Int'l Conf. Software Engineering (ICSE)*, pages 105–114. ACM Press, 2010.
 - 13 J. Liebig, C. Kästner, and S. Apel. Analyzing the discipline of preprocessor annotations in 30 million lines of C code. In *Proc. Int'l Conf. Aspect-Oriented Software Development (AOSD)*, pages 191–202. ACM Press, 2011.
 - 14 J. Liebig, A. von Rhein, C. Kästner, S. Apel, J. Dörre, and C. Lengauer. Large-scale variability-aware type checking and dataflow analysis. Technical report, Department of Informatics and Mathematics, University of Passau, 2012.
 - 15 R. Lopez-Herrejon and D. Batory. A standard problem for evaluating product-line methodologies. In *Proc. Int'l Conf. Generative and Component-Based Software Engineering (GCSE)*, volume 2186 of *Lecture Notes in Computer Science*, pages 10–24. Springer-Verlag, 2001.
 - 16 M. Mendonca, M. Branco, and D. Cowan. S.p.l.o.t.: Software product lines online tools. In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, pages 761–762. ACM Press, 2009.
 - 17 S. She, R. Lotufo, T. Berger, A. Wąsowski, and K. Czarnecki. The variability model of the Linux kernel. In *Proc. Int'l Workshop on Variability Modelling of Software-intensive Systems (VaMoS)*, pages 45–51. University of Duisburg-Essen, 2010.
 - 18 R. Tartler, D. Lohmann, J. Sincero, and W. Schröder-Preikschat. Feature consistency in compile-time-configurable system software: Facing the Linux 10,000 feature problem. In *Proc. European Conference on Computer Systems (EuroSys)*, pages 47–60. ACM Press, 2011.
 - 19 R. Tartler, J. Sincero, W. Schröder-Preikschat, and D. Lohmann. Dead or alive: Finding zombie features in the Linux kernel. In *Proc. GPCE Workshop on Feature-Oriented Software Development (FOSD)*, pages 81–86. ACM Press, 2009.
 - 20 T. Thüm, S. Apel, C. Kästner, M. Kuhlemann, I. Schaefer, and G. Saake. Analysis strategies for software product lines. Technical Report FIN-004-2012, School of Computer Science, University of Magdeburg, Apr. 2012.
 - 21 A. von Rhein, S. Apel, C. Kästner, T. Thüm, and I. Schaefer. The pla model: On the combination of product-line analyses. In *Proceedings of the 7th Int'l Workshop on Variability Modelling of Software-Intensive Systems (VaMoS)*, pages 14:1–14:8, 1 2013.
 - 22 X. Yang, Y. Chen, E. Eide, and J. Regehr. Finding and understanding bugs in C compilers. In *Proc. Conf. Programming Language Design and Implementation (PLDI)*, pages 283–294. ACM Press, 2011.
 - 23 C. Yilmaz. Test case-aware combinatorial interaction testing. *IEEE Trans. Softw. Eng. (TSE)*, PP(99):1–1, 2012.

3.17 Is medical underwriting knowledge a field for verification in the presence of variability?

Kim Lauenroth (adesso AG – Dortmund, DE)

License © Creative Commons BY 3.0 Unported license
© Kim Lauenroth

Medical underwriting is a part of the application process and refers to the assessment of an applicant for insurance coverage (e.g. life or health insurances). Many insurance companies use software systems to automate the medical underwriting process.

The necessary assessment knowledge for these systems is specified by:

- (a) a large set of medical variables including weight, height, body mass index (bmi), type of diabetes;
- (b) different questions to elicit medical variables, *e.g.* “What is your height (cm)?” or “What is your weight (kg)?”;
- (c) rules that trigger additional questions, *e.g.* if (height > 200) then ask “Do you suffer from back pain?”;
- (d) rules that lead to a risk decision (if (type of diabetes == 2 AND bmi >30) then reject application).

Variables, rules, and questions vary in several dimensions, for example, between different insurance products, and between countries or regions of the world. Quality assurance of this assessment knowledge is a constant challenge for insurance companies, since the knowledge is modified regularly because of new medical research results and new insurance products.

Insurance companies typically perform the quality assurance of their assessment knowledge based on different sample test cases but do not use automated techniques such as model checking. This talk has two goals:

1. Give an introduction into the field of medical underwriting and into the structure of the assessment knowledge;
2. Discuss possible strategies for the automated quality assurance of assessment knowledge with the seminar participants.

3.18 Scientific Workflows: Eternal Components, Changing Interfaces, Varying Compositions

Tiziana Margaria (Universität Potsdam, DE)

License © Creative Commons BY 3.0 Unported license
© Tiziana Margaria

Joint work of Margaria, Tiziana; Lamprecht, Anna-Lena

Main reference A.-L. Lamprecht, T. Margaria, “Scientific workflows: eternal components, changing interfaces, varying compositions,” in Proc. of the 5th Int’l Conf. on Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change – Part I (ISoLA’12), LNCS, Vol. 7609, pp. 47–63, Springer, 2012.

URL http://dx.doi.org/10.1007/978-3-642-34026-0_5

We describe how scientific application domains are characterized by the long-term availability of the basic computational components, and how software systems for managing the actual scientific workflows must deal with changing service interfaces and varying service compositions. In this light, we explain how rigorous technical and semantic abstraction, which is key

to dealing with huge and heterogeneous application domains in an “extreme model driven design” framework like the jABC, supports the management of workflow evolution. We illustrate the different aspects by means of examples and experiences from the application of the framework in different scientific application domains.

3.19 Compositional Verification of Software Product Lines

Jean-Vivien Millo (INRIA Sophia Antipolis – Méditerranée, FR)

License © Creative Commons BY 3.0 Unported license
© Jean-Vivien Millo

Joint work of Millo, Jean-Vivien; Ramesh, S; Sankara Narayanan, Krishna; Narwane Kandhu, Ganesh
Main reference J.-V. Millo, S. Ramesh, K. Sankara Narayanan, G. Narwane Kandhu, “Compositional Verification of Software Product Lines,” in Proc. of the 10th Int’l Conf. on Integrated Formal Methods (IFM’13), LNCS, Vol. 7940, pp. 109–123, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-38613-8_8
URL <http://hal.archives-ouvertes.fr/hal-00747533/>

This work presents a novel approach to the design verification of Software Product Lines (SPL). The proposed approach assumes that the requirements and designs at the feature level are modeled as finite state machines with variability information. The variability information at the requirement and design levels are expressed differently and at different levels of abstraction. Also the proposed approach supports verification of SPL in which new features and variability may be added incrementally. Given the design and requirements of an SPL, the proposed design verification method ensures that every product at the design level behaviourally conforms to a product at the requirement level. The conformance procedure is compositional in the sense that the verification of an entire SPL consisting of multiple features is reduced to the verification of the individual features. The method has been implemented and demonstrated in a prototype tool SPLEnD (SPL Engine for Design Verification) on a couple of fairly large case studies.

3.20 Feature Maintenance with Emergent Interfaces

Marcio Ribeiro (Federal University of Pernambuco – Recife, BR)

License © Creative Commons BY 3.0 Unported license
© Marcio Ribeiro

Joint work of Ribeiro, Marcio; Pacheco, Humberto; Teixeira, Leopoldo; Borba, Paulo
Main reference M. Ribeiro, H. Pacheco, L. Teixeira, P. Borba. “Emergent Feature Modularization,” in In Onward!, affiliated with ACM SIGPLAN Int’l Conf. on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH’10), pp. 11–18, ACM, 2012.
URL <http://dx.doi.org/10.1145/1869542.1869545>

Hidden code dependencies are responsible for many complications in maintenance tasks. With the introduction of variable features in product lines, dependencies may even cross feature boundaries and related problems are prone to be detected late. Many current implementation techniques for product lines lack proper interfaces, which could make such dependencies explicit. As alternative to changing the implementation approach, we introduce a tool-based solution to support developers in recognizing and dealing with feature dependencies: emergent interfaces. Emergent interfaces are computed on demand, based on feature-sensitive interprocedural dataflow analysis. They emerge in the IDE and emulate benefits of modularity not available in the host language.

References

- 1 M. Ribeiro, H. Pacheco, L. Teixeira, and P. Borba. Emergent Feature Modularization. In *Onward!, affiliated with ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH)*, pages 11–18, New York, NY, USA, 2010. ACM.
- 2 M. Ribeiro, F. Queiroz, P. Borba, T. Tolêdo, C. Brabrand, and S. Soares. On the impact of feature dependencies when maintaining preprocessor-based software product lines. In *Proc. of the 10th ACM International Conference on Generative Programming and Component Engineering (GPCE)*, pages 23–32, Portland, Oregon, USA, 2011. ACM.
- 3 C. Brabrand, M. Ribeiro, T. Tolêdo, J. Winther, and P. Borba. Intraprocedural dataflow analysis for software product lines. *Transactions on Aspect-Oriented Software Development (TAOSD)*, 10:73–108, 2013.

3.21 Delta-oriented Regression-based Testing of Software Product Lines

Ina Schaefer (TU Braunschweig, DE)

License © Creative Commons BY 3.0 Unported license
© Ina Schaefer

Joint work of Schaefer, Ina; Malte Lochau; Sascha Lity

Main reference M. Lochau, I. Schaefer, J. Kamischke, S. Lity, “Incremental Model-Based Testing of Delta-Oriented Software Product Lines,” in *Proc. of the 6th Int’l Conf. on Tests and Proofs (TAP’12)*, LNCS, Vol. 7305, pp. 67–82, Springer, 2012.

URL http://dx.doi.org/10.1007/978-3-642-30473-6_7

Software architecture specifications are of growing importance for coping with the complexity of large-scale systems. They provide an abstract view on the high-level structural system entities together with their explicit dependencies and build the basis for ensuring behavioral conformance of component implementations and interactions, e.g., using model-based integration testing. The increasing inherent diversity of such large-scale variant-rich systems further complicates quality assurance. In this article, we present a combination of architecture-driven model-based testing principles and regression-inspired testing strategies for efficient, yet comprehensive variability-aware conformance testing of variant-rich systems. We propose an integrated delta-oriented architectural test modeling and testing approach for component as well as integration testing that allows the generation and reuse of test artifacts among different system variants. Furthermore, an automated derivation of retesting obligations based on accurate delta-oriented architectural change impact analysis is provided. Based on a formal conceptual framework that guarantees stable test coverage for every system variant, we present a sample implementation of our approach and an evaluation of the validity and efficiency by means of a case study from the automotive domain.

3.22 Model-Based Testing of Software Product Lines

Holger Schlingloff (HU Berlin, DE)

License © Creative Commons BY 3.0 Unported license
© Holger Schlingloff

We give an introduction to software product lines as defined in the literature, share our view on model-based testing, and survey some recent work in the model-based testing of software product lines.

3.23 Reuse of Test Cases for Model-Based Development of Software Product Lines

Holger Schlingloff (HU Berlin, DE)

License © Creative Commons BY 3.0 Unported license
© Holger Schlingloff

Main reference T. Kahsai, M. Roggenbach, B.-H. Schlingloff, “Specification-Based Testing for Software Product Lines,” in Proc. of the 6th IEEE Int’l Conf. on Software Engineering and Formal Methods (SEFM’08), pp. 149–158, IEEE CS, 2008.

URL <http://dx.doi.org/10.1109/SEFM.2008.38>

URL http://www2.informatik.hu-berlin.de/~hs/Publikationen/2008_SEFM_Kahsai-Roggenbach-Schlingloff_Specification-based-Testing-of-Product-Lines.pdf

In a model-based development process of a software product line, the feature model determines which functions are present in each instance. The functional model elaborates the features to be transformed into an implementation. Both feature model and functional model are constantly enhanced and extended to allow for the incorporation of new features and functionalities. A frequent problem in this process is to decide which test cases can be reused from one instance to the next one. New features can extend the existing ones, or be in conflict with them. Thus, some test cases remain valid for the new product, while others have to be reworked. In this talk, we formalize these notions and show how to decide which parts of a test suite can be reused in a controlled product line development.

3.24 Test Case Prioritization Criteria for Software Product Lines

Sergio Segura (University of Sevilla, ES)

License © Creative Commons BY 3.0 Unported license
© Sergio Segura

Testing all the products of a software product line is usually unfeasible, there are simply too many. To address this problem, contributions in the last years has focused on obtaining a representative subset of the products to be tested. But, once we have selected a set of products, is the order in which they are tested relevant? We think so. This is what test case prioritization techniques are about. For example, testers may wish to order their test cases in order to achieve code coverage at the fastest rate possible, employ features in order of expected frequency of use or increase the rate of fault detection of test cases. Our preliminary results show that the order in which test cases are run may have a significant impact in the rate of fault detection.

3.25 Composing Variable Components Considering Interaction Behavior – A Problem Statement

Vanessa Stricker (Universität Duisburg – Essen, DE)

License © Creative Commons BY 3.0 Unported license
© Vanessa Stricker

Current approaches for integrating variability into hierarchical component structures neglect the behavioral specification of the component hierarchy. Though, in single system development it is acknowledged that early verification, ensuring that a component composition behaves as intended, is crucial. Especially the interaction between the components might result in unexpected deviations and inconsistencies between the actual and intended behavior.

Detecting these inconsistencies in composed component behavior is even more important, but also more complicated, in the presence of variability. The analysis of composed behavior has to identify if variable components can safely be composed as well as how the components have to be configured (i.e. binding of variability) in a composition in order to behave as intended. This is in particular important since a component's behavior might be restricted by its relations to other components. The influence of variability onto these behavior restrictions needs to be analyzed carefully and cannot be considered in isolation — especially if the variability of the components is described decentralized in multiple variability models, e.g. for 3rd party components or supplier product lines. To enable a rigorous analysis and reasoning about the behavioral correctness of variable component compositions, existing component-oriented design and analysis methods need to be extended to be able to consider variability. Using existing approaches of the formal methods community in an engineering method for composing variable components has to address certain challenges: There is a large set of existing formal methods in single system development and software product line engineering, which focus on verifying specific properties of component behaviors while having certain assumptions and restrictions. Therefore, the question arises whether these approaches can be used in an engineering approach, how they might have to be adapted and combined and how the results can be interpreted and exploited.

3.26 Safe Evolution of Software Product Lines and Communities

Leopoldo Teixeira (Federal University of Pernambuco – Recife, BR)

License © Creative Commons BY 3.0 Unported license
© Leopoldo Teixeira

Joint work of Teixeira, Leopoldo; Borba, Paulo; Gheyi, Rohit

Main reference P. Borba, L. Teixeira, R. Gheyi, “A theory of software product line refinement,” *Theoretical Computer Science*, Vol. 455, pp. 2–30, 2010.

URL <http://dx.doi.org/10.1016/j.tcs.2012.01.031>

Software Product Line evolution can benefit from refactorings with formal basis, to ensure correctness by construction. In this talk, we present a language independent theory of product line refinement, establishing refinement properties that justify stepwise and compositional product line evolution [1]. Instead of dealing directly with the stronger notion of refactoring, we focus on refinement, which also captures behavior preservation but abstracts quality improvement. We take the broader view of refinement as a relation that preserves properties necessary to assure safe evolution. The theory also supports reasoning about sets of product lines that define communities. To illustrate one of the practical applications of the theory, we introduce and prove soundness of a number of refinement transformation templates [2], that range from evolving individual artifacts to the product line as a whole, and sets of product lines. These templates can be used as the basis to derive comprehensive product line refinement catalogues. We use the Prototype Verification System to encode and prove soundness of the theory and associated properties and templates.

References

- 1 Paulo Borba, Leopoldo Teixeira, and Rohit Gheyi. A theory of software product line refinement. *Theoretical Computer Science*, 455:2–30, 2012.
- 2 Laís Neves, Leopoldo Teixeira, Demóstenes Sena, Vander Alves, Uirá Kulezsa, and Paulo Borba. Investigating the safe evolution of software product lines. In *Proceedings of the 10th ACM International Conference on Generative Programming and Component Engineering (GPCE'11)*, pages 33–42, 2011.

3.27 Product-Line Verification with Contracts

Thomas Thüm (Universität Magdeburg, DE)

License © Creative Commons BY 3.0 Unported license
© Thomas Thüm

Main reference T. Thüm, “Verification of Software Product Lines Using Contracts,” in Doktorandentagung Magdeburger-Informatik-Tage (MIT), pp. 75–82, University of Magdeburg, 2012.

URL http://www.witi.cs.uni-magdeburg.de/iti_db/publikationen/ps/auto/Th:MIT12.pdf

Software product lines challenge existing specification and verification techniques known from single-system engineering. Specifying and verifying each product separately involves redundant effort and is usually not feasible. A promising technique is to specify and verify product lines based on design by contract. We apply generative programming to contracts, and generate contracts for each product based on domain artifacts. Furthermore, we analyze whether all products fulfill their contracts using model checking, static analysis, and deductive verification. Finally, we discuss pitfalls and benefits of design by contract for product-line verification.

References

- 1 S. Apel, A. von Rhein, T. Thüm, and C. Kästner. Feature-Interaction Detection based on Feature-Based Specifications. *Computer Networks*, 2013. To appear.
- 2 F. Benduhn. Contract-Aware Feature Composition. Bachelor’s thesis, University of Magdeburg, Germany, 2012.
- 3 W. Scholz, T. Thüm, S. Apel, and C. Lengauer. Automatic Detection of Feature Interactions using the Java Modeling Language: An Experience Report. In *Proc. Int’l Workshop Feature-Oriented Software Development (FOSD)*, pages 7:1–7:8. ACM, 2011.
- 4 T. Thüm. Verification of Software Product Lines Using Contracts. In *Doktorandentagung Magdeburger-Informatik-Tage (MIT)*, pages 75–82. University of Magdeburg, 2012.
- 5 T. Thüm, S. Apel, C. Kästner, M. Kuhlemann, I. Schaefer, and G. Saake. Analysis Strategies for Software Product Lines. Technical Report FIN-004-2012, School of Computer Science, University of Magdeburg, Germany, 2012.
- 6 T. Thüm, I. Schaefer, S. Apel, and M. Hentschel. Family-Based Deductive Verification of Software Product Lines. In *Proc. Int’l Conf. Generative Programming and Component Engineering (GPCE)*, pages 11–20. ACM, 2012.
- 7 T. Thüm, I. Schaefer, M. Kuhlemann, and S. Apel. Proof Composition for Deductive Verification of Software Product Lines. In *Proc. Int’l Workshop Variability-intensive Systems Testing, Validation and Verification (VAST)*, pages 270–277. IEEE, 2011.
- 8 T. Thüm, I. Schaefer, M. Kuhlemann, S. Apel, and G. Saake. Applying Design by Contract to Feature-Oriented Programming. In *Proc. Int’l Conf. Fundamental Approaches to Software Engineering (FASE)*, volume 7212 of *LNCS*, pages 255–269. Springer, 2012.

3.28 Evaluating Dataflow Analysis for Software Product Lines

Tarsis Tolêdo (*Federal University of Pernambuco – Recife, BR*)

License © Creative Commons BY 3.0 Unported license
© Tarsis Tolêdo

Joint work of Brabrand, Claus; Ribeiro, Márcio; Tolêdo, Tarsis; Borba, Paulo

Main reference C. Brabrand, M. Ribeiro, T. Tolêdo, P. Borba, “Intraprocedural dataflow analysis for software product lines,” in Proc. of the 11th Annual Int’l Conf. on Aspect-oriented Software Development (AOSD’12), pp. 13–24, ACM, 2012.

URL <http://dx.doi.org/10.1145/2162049.2162052>

Brabrand et al. [1] proposed four different approaches to perform feature-sensitive intraprocedural dataflow analysis, which avoids having to explicitly generate each variant of a method before analyzing it. We evaluated these four approaches on four different software product lines and learned that each approach behaves differently depending on different characteristics of each method, like method size, size of lattice domain, number of confluence points; and also on different characteristics of the variability constructs, such as number, size and position of the #ifdef statements. In this talk, we discuss the confounding factors and possible strategies for a better evaluation of the approaches.

References

- 1 Claus Brabrand, Márcio Ribeiro, Tarsis Tolêdo, and Paulo Borba. Intraprocedural Dataflow Analysis for Software Product Lines. In *Proceedings of the 11th International Conference on Aspect-oriented Software Development (AOSD 2012)*, pages 13–24. ACM, Potsdam, Germany, 2012.

3.29 Inferring Variational Types for Variational Programs

Eric Walkingshaw (*Oregon State University, US*)

License © Creative Commons BY 3.0 Unported license
© Eric Walkingshaw

Joint work of Chen, Sheng; Erwig, Martin; Walkingshaw, Eric

Main reference S. Chen, M. Erwig, E. Walkingshaw, “An Error-Tolerant Type System for Variational Lambda Calculus,” in Proc. of the ACM SIGPLAN Int’l Conf. on Functional Programming (ICFP’12), pp. 29–40, ACM, 2012.

URL <http://dx.doi.org/10.1145/2364527.2364535>

URL <http://eecs.oregonstate.edu/~erwig/ToSC/VLC-TypeSystem.pdf>

I presented a type system and type inference algorithm for variational lambda calculus (VLC). VLC extends the lambda calculus with a simple construct for representing variation points as choices between alternatives. Each choice is associated with a dimension and all choices in the same dimension are synchronized. The type of a VLC expression is a correspondingly variational type. VLC and variational types are two different instantiations of the choice calculus, making the extension of the Damas-Milner algorithm systematic, and giving us many components of the type system and inference algorithm “for free”.

References

- 1 S. Chen, M. Erwig, and E. Walkingshaw. Extending Type Inference to Variational Programs. (Journal paper, in minor revision). 2013.
- 2 S. Chen, M. Erwig, and E. Walkingshaw. An Error-Tolerant Type System for Variational Lambda Calculus. *ACM SIGPLAN Int. Conf. on Functional Programming*, pages 29–40, 2012.

3.30 Combinatorial Interaction Testing

Cemal Yilmaz (Sabanci University – Istanbul, TR)

License © Creative Commons BY 3.0 Unported license
© Cemal Yilmaz

Main reference C. Yilmaz, “Test Case-Aware Combinatorial Interaction Testing”, IEEE Transactions on Software Engineering, Vol. 39, No. 5, pp. 684–706, 2013.

URL <http://dx.doi.org/10.1109/TSE.2012.65>

The configuration spaces of modern software systems are too large to test exhaustively. Combinatorial interaction testing (CIT) approaches, such as covering arrays, systematically sample the configuration space and test only the selected configurations. Given a configuration space model, a t-way covering array is a set of configurations in which each valid combination of option settings for every combination of t options appears at least once. This talk has two parts. In the first part, I provide a brief survey of CIT approaches, including t-way covering arrays, system-wide inter-option constraints, seeding mechanisms, and variable strength covering arrays. In the second part, I introduce two novel combinatorial objects for testing, namely test case-aware covering arrays and cost-aware covering arrays. Traditional covering arrays, while taking system-wide inter-option constraints into account, do not provide a systematic way of handling test case-specific inter-option constraints. Thus they suffer from masking effects – unaccounted test case-specific constraints perturb program executions so as to prevent some option-related behaviors from being exercised. On the other hand, test case-aware covering arrays account for test case-specific constraints while constructing covering sets. A t-way test case-aware covering array is not just a set of configurations as is the case in traditional covering arrays, but a set of configurations, each of which is associated with a set of test cases, such that all system-wide and test case-specific constraints are satisfied and that, for each test case, each valid combination of option settings for every combination of t options appears at least once in the set of configurations that the test case is associated with. Another downside of traditional covering arrays is that they simply assume that every configuration costs the same. We, however, argue that this is often not the case in practice. Cost aware-covering arrays take actual cost of testing into account when computing covering sets. A t-way cost-aware covering array is a t-way covering array that minimizes a given cost function.

3.31 Effective Test Execution for Software Product Lines

Sabrina de Figueirêdo Souto (Federal University of Pernambuco – Recife, BR)

License © Creative Commons BY 3.0 Unported license
© Sabrina de Figueirêdo Souto

Joint work of de Figueirêdo Souto, Sabrina; d’Amorim, Marcelo

URL <http://www.cin.ufpe.br/~sfs/souto-dagstuhl2013-talk.pdf>

Software Product Lines (SPLs) have gained significant attention recently as an approach to improve productivity in development of software families. Considering that the number of valid products in one SPLs can be very high, one important practical problem is to speedup testing. This paper evaluates Symbolic Execution of Features (SEF) to address this problem. It is widely known that general symbolic execution is expensive compared to concrete execution. The efficiency of SEF comes from its ability to partition the state in two parts: one symbolic and one concrete. The symbolic part models the features which need to be enabled in the system along the analysis of one path while the concrete part models the

entire program state. In SEF, the symbolic data does not flow to the stores associated to the concrete parts of the state; hence, operations that elaborate the program state perform efficiently as they only manipulate concrete data. Feasibility checking of paths is achieved with efficient incremental sat solving of feature constraints.

3.32 Modelling, analysing and verifying variability by means of Modal Transition Systems

Maurice H. ter Beek (CNR – Pisa, IT)

License © Creative Commons BY 3.0 Unported license

© Maurice H. ter Beek

Joint work of ter Beek, Maurice H.; Fantechi, Alessandro; Gnesi, Stefania; Mazzanti, Franco; Asirelli, Patrizia
Main reference P. Asirelli, M.H. ter Beek, S. Gnesi, A. Fantechi, “Formal Description of Variability in Product Families,” in Proc. of the 15th Int’l Software Product Line Conference (SPLC’11), pp. 130–139, IEEE, 2011.

URL <http://dx.doi.org/10.1109/SPLC.2011.34>

Our aim is a unifying logical framework for modelling, analysing and verifying behavioural variability in product families, with tool support for formal verification. We use Modal Transition Systems (MTSs) for modelling the behaviour and v-CTL, a suitable action-based branching-time temporal logic interpreted over MTSs, for the necessary additional constraints. Our tool (VMC) accepts a product family specified as an MTS, possibly with additional variability constraints. Subsequently, VMC can be used for the derivation, exploration and analysis (from ‘family-based’ to ‘product-based’ analyses) of the family and its products, including the efficient verification of temporal logic properties through on-the-fly model checking. This talk is based on a number of previous publications [1, 2, 3].

References

- 1 Patrizia Asirelli, Maurice H. ter Beek, Alessandro Fantechi, and Stefania Gnesi. *A Logical Framework to Deal with Variability*. In Proceedings 8th International Conference on Integrated Formal Methods (IFM 2010). LNCS 6396, Springer, 2010, 43–58.
- 2 Patrizia Asirelli, Maurice H. ter Beek, Alessandro Fantechi, and Stefania Gnesi. *Formal Description of Variability in Product Families*. In Proceedings 15th International Software Product Line Conference (SPLC 2011). IEEE, 2011, 130–139.
- 3 Maurice H. ter Beek, Franco Mazzanti, and Aldi Sulova. *VMC: A Tool for Product Variability Analysis*. In Proceedings 18th International Symposium on Formal Methods (FM 2012). LNCS 7436, Springer, 2012, 450–454.

Participants

- Vander Alves
University of Brasilia, BR
- Sven Apel
Universität Passau, DE
- Joanne M. Atlee
University of Waterloo, CA
- Kacper Bał
University of Waterloo, CA
- Don Batory
University of Texas at Austin, US
- Thorsten Berger
IT Univ. of Copenhagen, DK
- Eric Bodden
TU Darmstadt, DE
- Paulo Borba
University of Pernambuco –
Recife, BR
- Claus Brabrand
IT Univ. of Copenhagen, DK
- Dave Clarke
KU Leuven, BE
- Andreas Classen
University of Namur, BE
- Roberta Coelho
Federal University of Rio Grande
do Norte, BR
- Myra Cohen
Univ. of Nebraska – Lincoln, US
- Maxime Cordy
Facultés Universitaires
Notre-Dame de la Paix, BE
- Krzysztof Czarnecki
University of Waterloo, CA
- Sabrina de Figueirêdo Souto
University of Pernambuco –
Recife, BR
- Martin Erwig
Oregon State University, US
- Alessandro Fantechi
University of Firenze, IT
- Brady J. Garvin
Univ. of Nebraska – Lincoln, US
- Rohit Gheyi
Federal University of Campina
Grande, BR
- Stefania Gnesi
CNR – Pisa, IT
- Reiner Hähnle
TU Darmstadt, DE
- Øystein Haugen
SINTEF – Oslo, NO
- Martin Fagereng Johansen
University of Oslo, NO
- Christian Kästner
Carnegie Mellon University –
Pittsburgh, US
- Shriram Krishnamurthi
Brown University, US
- Kim Lauenroth
adesso AG – Dortmund, DE
- Axel Legay
INRIA Bretagne Atlantique –
Rennes, FR
- Martin Leucker
Universität Lübeck, DE
- Tiziana Margaria
Universität Potsdam, DE
- Dusica Marijan
Simula Reseach Laboratory –
Lysaker, NO
- Jean-Vivien Millo
INRIA Sophia Antipolis –
Méditerranée, FR
- Gilles Perrouin
University of Namur, BE
- Márcio Ribeiro
University of Pernambuco –
Recife, BR
- Ina Schaefer
TU Braunschweig, DE
- Holger Schlingloff
HU Berlin, DE
- Sergio Segura
University of Sevilla, ES
- Vanessa Stricker
Univ. Duisburg–Essen, DE
- Leopoldo Teixeira
University of Pernambuco –
Recife, BR
- Maurice H. ter Beek
CNR – Pisa, IT
- Thomas Thüm
Universität Magdeburg, DE
- Társis Tolêdo
University of Pernambuco –
Recife, BR
- Salvador Trujillo
Ikerlan Research Centre –
Arrasate-Mondragón, ES
- Eric Walkingshaw
Oregon State University, US
- Andrzej Waśowski
IT Univ. of Copenhagen, DK
- Cemal Yilmaz
Sabanci Univ. – Istanbul, TR

