DAGSTUHL
REPORTS

Volume 2, Issue 7, July 2012

Dagstuhl Reports, Vol. 2, Issue 7 **ISSN 2192-5283**

ISSN 2192-5283

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany.
Online available at http://www.dagstuhl.de/dagrep

Publication date
January, 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`.

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

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

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

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

Digital Object Identifier: 10.4230/DagRep.2.7.i

www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 12271

AI meets Formal Software Development

Edited by

Alan Bundy[1], Dieter Hutter[2], Cliff B. Jones[3], and J Strother Moore[4]

1 **University of Edinburgh, GB,** `a.bundy@ed.ac.uk`
2 **DFKI Saarbrücken, DE,** `hutter@dfki.de`
3 **Newcastle University, GB,** `cliff.jones@ncl.ac.uk`
4 **University of Texas at Austin, US,** `moore@cs.utexas.edu`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 12271 "AI meets Formal Software Development". This seminar brought together researchers from formal methods and AI. The participants addressed the issue of how AI can aid the formal software development process, including modelling and proof. There was a pleasing number of participants from industry and this made it possible to ground the discussions on industrial-scale problems.

Seminar 01.–06. July, 2012 – www.dagstuhl.de/12271
1998 ACM Subject Classification F.3.1 Specifying and Verifying and Reasoning about Programs, I.2 Artificial intelligence (specifically I.2.3, I.2.4, I.2.6)
Keywords and phrases Learning of proof processes and strategies, Theory development, Formal software development, Automated reasoning, Formal modelling, Industrial use of formal methods
Digital Object Identifier 10.4230/DagRep.2.7.1
Edited in cooperation with Andrius Velykis (Newcastle University, GB)

1 Executive Summary

Cliff B. Jones

This seminar brought together researchers from formal methods and AI. The participants addressed the issue of how AI can aid the formal software development process, including modelling and proof. There was a pleasing number of participants from industry and this made it possible to ground the discussions on industrial-scale problems.

Background

Industrial use of formal methods is certainly increasing but in order to make it more mainstream, the cost of applying formal methods, in terms of mathematical skill level and development time, must be reduced — and we believe that AI can help with these issues.

Rigorous software development using formal methods allows the construction of an accurate characterisation of a problem domain that is firmly based on mathematics; by applying standard mathematical analyses, these methods can be used to prove that systems satisfy formal specifications. A recent ACM computing survey [1] describes over sixty industrial projects and discusses the effect formal methods have on time, cost, and quality. It shows that with tools backed by mature theory, formal methods are becoming cost effective and their use is easier to justify, not as an academic exercise or legal requirement, but as part of a business case. Furthermore, the use of such formal methods is no longer confined to safety critical systems: the list of industrial partners in the DEPLOY project[1] is one indication of this broader use. Most methods tend to be posit-and-prove, where the user posits a development step (expressed in terms of specifications of yet-to-be-realised components) that has to be justified by proofs. The associated properties that must be verified are often called proof obligations (POs) or verification conditions. In most cases, such proofs require mechanical support by theorem provers.

One can distinguish between automatic and interactive provers, where the latter are generally more expressive but require user interaction. Examples of state-of-the-art interactive theorem provers are ACL2, Isabelle, HOL, Coq and PVS, while E, SPASS, Vampire and Z3 are examples of automatic provers.

AI has had a large impact on the development of provers. In fact, one of the first AI application was a theorem prover and all theorem provers now contain heuristics to reduce the search space that can be attributed to AI. Nevertheless, theorem proving research and (pure) AI research have diverged, and theorem proving is barely considered to be AI-related anymore.

There follows a list of background references.

References

1 J. Woodcock, P. G. Larsen, J. Bicarregui, and J. S. Fitzgerald. Formal methods: Practice and experience. *ACM Computing Surveys*, 41(4), 2009.

2 A. Bundy. *The Computer Modelling of Mathematical Reasoning.* Academic Press, 1983. (2nd edition 1986).

3 A. Bundy and A. Smaill. *A Catalogue of Artificial Intelligence Techniques.* Springer, 1984. (2nd edition 1986, 3rd edition 1990, 4th revised edition 1997).

4 A. Bundy, F. van Harmelen, J. Hesketh, and A. Smaill. Experiments with proof plans for induction. *J. Autom. Reasoning*, 7(3):303–324, 1991.

5 A. Bundy. A science of reasoning. In J.-L. Lassez and G. Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*, pages 178–198. MIT Press, 1991.

6 A. Bundy. The automation of proof by mathematical induction. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 847–911. Elsevier and MIT Press, 2001.

7 A. Bundy, D. Basin, D. Hutter, and A. Ireland. *Rippling: Meta-level Guidance for Mathematical Reasoning*, volume 56 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, June 2005.

8 D. Hutter and W. Stephan, editors. *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *LNCS*. Springer, 2005.

9 D. Hutter, W. Stephan, P. Traverso, and M. Ullmann, editors. *Proceedings of Current Trends in Applied Formal Methods (FM-Trends 98)*, volume 1641 of *LNCS*, Boppard, Germany, 1999. Springer.

10 C. B. Jones, K. D. Jones, P. A. Lindsay, and R. Moore. *mural: A Formal Development Support System.* Springer, 1991.

[1] DEPLOY was an EU-funded “IP” led by Newcastle University; a four year project with a budget of about 18M Euros; the industrial collaborators include Siemens Transport, Bosch and SAP.

11 R. S. Boyer and J. S. Moore. *A Computational Logic Handbook.* Formal Methods Series. Academic Press, second edition, 1997.

12 M. Kaufmann, P. Manolios, and J. S. Moore. *Computer-Aided Reasoning: An Approach*, volume 3 of *Advances in Formal Methods.* Kluwer Academic Publishers, 2000.

13 M. Kaufmann and J. S. Moore. Some key research problems in automated theorem proving for hardware and software verification. *Revista de la Real Academia de Ciencias (RAC-SAM)*, 98(1):181–196, 2004.

Organisation of the seminar

It might be useful to organisers of future seminars to record some organisational issues. We asked participants to prepare only short talks that introduced topics and –just as we wished– a number of the talks were actually prepared at the seminar location and with the benefit of having heard other talks. This free format worked well for our exchange of ideas and in most regards we were pleased that we started with only the Monday morning actually scheduled. Perhaps the biggest casualty of the fluid organisation (coupled with so many interesting participants) was that there was no time left for Panel Sessions. However, the differing lengths of discussions (and liberal use of breaks and a "hike" for people to establish new links) led to intensive interaction.

Notes on nearly all of the talks are contained in Section 3 of the current document.

It is a pleasure to extend our thanks to everyone involved in the Dagstuhl organisation: they provide a supportive and friendly context in which such fruitful scientific exchanges can develop unhindered by distraction.

Results

It is possible to address the results under the phases of the development cycle. Requirements capture is traditionally a pre-formal exercise and is the phase where one would expect least impact from formal ideas. There is certainly scope here for the use of ontologies and some hope for help in detecting inconsistencies in requirements but little time was spent in the seminar on these topics.

Once development moves to the creation of a specification, the scope for formalism increases and with it the hope for a greater contribution from AI. Essentially, a formal specification is a model. Formal proof can be used to establish internal consistency properties or to prove that properties match expectations about the required system. Model checking approaches are often the most efficient way of detecting inconsistencies.

Steps of development (in the posit and prove approaches) essentially introduce further models which should relate in precise ways to each other. The technical details vary between development methods but the overall implications for the use of proof and the contribution of AI are similar. It is perhaps worth reemphasising here that the seminar was trying to address problems of an industrial scale.

An interesting dichotomy was explored at the seminar concerning POs that fail to discharge. One school of thought is to interpose extra models in order to cause the generation of simpler POs; the alternative is to take the POs as fixed and develop "theories" (collections of auxiliary functions and lemmas) to complete the proof process. Suffice it to say here that AI was seen to have a role in both approaches.

More generally, the whole task of refactoring models and reusing libraries of established material is another area seen as being in need of help from AI thinking.

Turning to the richest area of collaboration –that of proof itself– a prominent theme was on the ways in which machine learning can help. There are many facets of this question including analogy with previous proofs, data mining of proofs (and failures) and proof strategy languages.

One particularly important aspect of the cost of proof in an industrial setting is proof maintenance. In practical settings, many things change and it is unlikely to be acceptable to have to repeat the whole proof process after each change.

Another area that led to useful interactions between participants was the subject of failure analysis and repair. It was observed that it is useful to have strong expectations as to how proofs were meant to succeed.

In conclusion many points of contact can be seen in the material presented below. Unsurprisingly, the material ranges from hopes for future research to mature results that can be readily applied. It is not only a hope that the links between ideas and researchers made at the seminar will persist — we already have clear proof of collaborative work.

The four organisers are extremely grateful to Andrius Velykis who took on the whole of the task of collecting and tidying the input in Section 3 representing the contributions of the speakers.

2 Table of Contents

Executive Summary

Overviews of Talks

3 Overviews of Talks

3.1 Applying Formal Methods In Industry

Rob Arthan (Lemma 1 Ltd. – Reading, GB)

Joint work of Arthan, Rob; Jones, Roger; O'Halloran, Colin

The talk was an overview of the speaker's experience of industrial applications of formal methods mostly involving the Z Notation and the ProofPower tools for specification and verification. This included a description of the CLawZ toolset that combines ProofPower and other tools into a system for verifying code that is automatically generated from Simulink specifications. Developed by Colin O'Halloran and others of DRisQ Ltd (http://drisq.com), CLawZ offers an independent verification path allowing the use of an untrusted code generator in the development of safety-critical systems, such as avionics control systems.

The talk concluded with some discussion of software engineering generally and offered a challenge for AI and formal methods: software developers often need to "work around" problems in software in the field that arise as a result of errors in the development process or of change in operating environments. What help can AI and formal methods offer to engineers who have to reason about systems that include flawed components?

3.2 Structure Formation in Formal Theories

Serge Autexier (DFKI Bremen, DE)

It has been long recognized that the modularity of specifications is an indispensable prerequisite for an efficient reasoning in complex domains. Algebraic specification techniques provide appropriate frameworks for structuring complex specifications and the notion of development graphs has been introduced as a technical means to work with and reason about such structured specifications. In this work we are concerned with assisting the process of structuring specifications in order to make intrinsic structures that are hidden explicit. Based on development graphs, we present an initial methodology and a formal calculus to transform unstructured specifications into structured ones. The calculus rules operate on development graphs allowing one to separate specifications coalesced in one theory into a structured graph. The calculus can both be used to structure a flat specification into sensible modules or to restructure existing structurings. We present an initial methodology to support the process of structure formations in large unstructured specifications.

3.3 Automated Reasoning and Formal Methods

Alan Bundy (University of Edinburgh, GB)

We review progress in automated reasoning in the last four decades and ask whether our provers are now sufficiently mature to support formal methods in mainstream ICT system development. We look at improvements due to Moore's Law and improvements in decision procedures, automatic provers, inductive provers and interactive provers. We ask what more AI can offer to further improve the situation. In particular, we speculate about the role of machine learning, e.g., to data-mine interactive proofs to extract proof tactics to be used to increase automation.

The slides for this talk are available on the AI4FM project website (http://www.ai4fm.org).

3.4 Explicit vs Implicit Search Guidance

Alan Bundy (University of Edinburgh, GB)

Suppose we want to analyse an interactively produced proof of a proof obligation, extract the proof strategy underlying this source proof and then apply it to guide the target automatic proofs of similar proof obligations. What form should such strategies take? We compare and contrast two alternatives, which we characterise as explicit and implicit. Explicit strategies are hierarchies of proof tactics, such as those described by Gudmund Grov in his talk at this Seminar. Implicit strategies are schemas that abstract the additional lemmas that are introduced in the source proof. We can then use theory exploration tools, such as Omar Montano Rivas's IsaScheme, to generate similar lemmas for the target proof by instantiating these schemas and proving the resulting conjectures. The implicit strategies trade off a loss of fine control for increased flexibility. The hypothesis to be evaluated is whether our provers are now sufficiently autonomous to find the right way to use the newly instantiated lemmas in the target proofs.

The slides for this talk are available on the AI4FM project website (http://www.ai4fm.org).

3.5 Reflections on AI meets Formal Software Development

Alan Bundy (University of Edinburgh, GB)

These are my wrap-up slides at the end of the Seminar. I tried to itemise all the points of contact that had emerged during the meeting. I grouped these under the headings of: requirements capture, modelling, proof, and failure analysis and repair.

The slides for this talk are available on the AI4FM project website (http://www.ai4fm.org).

3.6 Can IsaScheme be Used for Recursive Predicate Invention?

Alan Bundy (University of Edinburgh, GB)

URL https://dream.inf.ed.ac.uk/protected/Bluenote

At the Dagstuhl seminar "AI meets Formal Software Development" in July 2012, several people remarked on the importance of lemmas with conditions and the creativity involved in inventing these conditions. For instance, one sometimes wants to define new (recursive) predicates to provide these conditions. It occurred to me that IsaScheme might be used to generate these conditions and invent new recursive predicates. Blue book note 1763 at https://dream.inf.ed.ac.uk/protected/Bluenote explains the idea. If you don't have access to this protected area, contact me at a.bundy@ed.ac.uk for a copy.

3.7 Automated Theorem Proving in Perfect Developer and Escher C Verifier

David Crocker (Escher Technologies – Aldershot, GB)

We have two formal tools intended for development of high integrity software. Perfect Developer uses the specify-refine-generate paradigm, while Escher C Verifier is for formal verification of annotated hand-written MISRA-C code. Both have been used in developing industrial critical systems, and both use the same non-interactive theorem prover. I discuss different approaches to logics and theorem proving for software verification, drawing a comparison with the RISC/CISC processor wars of the 1990s, and outline the approach used by our prover. Finally, I discuss some areas where I think AI might be applied to improve the automation of our product.

3.8 Synthesizing Domain-specific Annotations

Ewen W. Denney (NASA – Moffett Field, US)

Joint work of Denney, Ewen; Fischer, Bernd

Verifying interesting properties on code typically requires logical annotations. If these annotations need to be added manually, this presents a significant bottleneck to automated verification. Here, we describe a language for encoding the domain knowledge needed to automatically generate such annotations for a class of mathematical properties. Interestingly, the problem of generating annotations turns out to essentially be a form of code generation.

3.9 Capturing and Inferring the Proof Process (Part 1: Case Studies)

Leo Freitas (Newcastle University, GB)

We report current work on inferring the proof process of an expert by wire-tapping various theorem proving environments (e.g. Isabelle/HOL, Z/EVES, etc). The idea is to have enough (meta-)proof information (i.e. user intent, lemmas used, points of failure and ways of recovery, various proof attempts [sub-]trees, etc.), in order to be able to do meta-level reasoning about proofs, in particular for proof reuse, but also for proof maintenance and transferability to non-expert users. For that we have worked on a number of case studies, large (e.g. EMV smart cards, Xenon security hypervisor), medium (e.g. Tokeneer ID station, Federated Cloud workflows properties, etc.), and small (e.g. Transitive closure lemma library, Union/Find Fisher/Galler problem characterisation, etc.). We are currently working on a paper describing this work to appear soon. Please visit http://www.ai4fm.org for more information.

3.10 Learning Component Abstractions

Dimitra Giannakopoulou (NASA – Moffett Field, US)

Automata learning algorithms are used increasingly in the research community to generate component abstractions or models. In this talk, we presented two of the learning-based frameworks that we have developed over the years, which use the automata learning algorithm L*. In the first framework, L* interacts with model checking to generate abstractions that are used as assumptions for automated compositional verification [1]. The second framework combines L* with symbolic execution to generate component interfaces [2]; the interfaces are three-valued, capturing whether a sequence of method invocations is safe, unsafe, or its effect on the component state is unresolved by the symbolic execution engine. The latter framework is available as the `jpf-psyco` project within the JavaPathfinder open source model checker for Java bytecode.

We then discussed other uses of learning in software engineering, including model and specification mining for black box or library components, potentially based on existing code that uses the components and is available on the Web. We believe that cross-fertilization with heuristic search used in AI applications will also be beneficial because exhaustive techniques often hit scalability issues. Finally, we believe that ultimately engineers should take verification into account when designing systems, and it is possible that machine learning could help us in the detection of good design or specification patterns.

References

1 J. M. Cobleigh, D. Giannakopoulou, and C. S. Pasareanu. Learning assumptions for compositional verification. In H. Garavel and J. Hatcliff, editors, *9th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2003)*, volume 2619 of *LNCS*, pages 331–346. Springer, 2003.

2 D. Giannakopoulou, Z. Rakamaric, and V. Raman. Symbolic learning of component interfaces. In A. Miné and D. Schmidt, editors, *19th Static Analysis Symposium (SAS 2012)*, volume 7460 of *LNCS*, pages 248–264. Springer, 2012.

3.11 A strategy language to facilitate proof re-use

Gudmund Grov (University of Edinburgh, GB)

Joint work of Grov, Gudmund; Bundy, Alan; Komandantskaya, Katya; Dixon, Lucas

Within refinement-based formal methods, such as B, Event-B, VDM and Z, many proofs follow a similar pattern. The ability to re-use expert-provided proofs to automatically discharge proof within the same family could therefore greatly improve automation of such proofs—which is currently a bottleneck for industrial application beyond niche markets. A similar phenomenon is also observed in mathematics through proof by analogy.

An observation we have made is that in order to capture sufficiently generic strategies, both goal and proof technique properties must be captured, which is not supported by current tactic languages. Here, we introduced work-in-progress on a graph based strategy language, where nodes are proof techniques and edges contains goal properties. Evaluation is then achieved by sending actual goals down the edges of the graph, and updating the proof by applying the technique to the input goal, and sending new goals to the output edges. We outlined some properties about this language, and briefly discussed a combination of stochastic learning methods (for pattern discovery) and logic-based learning methods (for strategy extraction/generalisation) in order to learn new proof strategies represented in this language.

3.12 Glassbox vs Blackbox Software Analysis

Reiner Haehnle (TU Darmstadt, DE)

Many approaches to software analysis/verification/synthesis/testing can be classified as either “glassbox” or “blackbox”. Both kinds have specific advantages and disadvantages, the latter preventing wider industrial usage. We argue that there is considerable potential in a systematic combination of both and sketch the outlines of a possible way to do this.

3.13 Beyond pure verification: AI for software development

Dieter Hutter (DFKI Bremen, DE)

Traditionally, verification is a process separated from standard development processes. We have our own methodologies and tools. Hence often Formal Methods (in particular theorem proving) is applied post mortem, i.e. after the actual development has been completed to ensure that everything went right at the end. From the viewpoint of a standard software engineer, Formal Methods are just a nuisance in the development process.

However, in the last ten years we have also seen the upcoming of semantic-based approaches to ease the software development process: model driven architecture, domain specific languages, a variety of web services and their (dynamic) composition, etc. We at DFKI have spent a lot of time in developing a change management for software development that maintain the relations between documents using semantic knowledge about them. Analysing informal documents in depth requires NL-understanding and thus a detailed ontology of the domain.

Thus, formally specified application domains are not only needed to verify the implementation but are also required to guide the overall design process, to automate (parts of) refinement process and to realize a powerful change (or development) management system. Combining these efforts in an integrated design process would multiply the benefits of applying formal methods but also ease the formal specification process becoming now incremental (starting at a simple ontology analogous to a standard glossary and ending in full fledged formal specs).

3.14 Reasoned Modelling: Exploiting the Synergy between Reasoning and Modelling

Andrew Ireland (Heriot-Watt University Edinburgh, GB)

Joint work of Ireland, Andrew; Grov, Gudmund; Llano, Maria Teresa; Pease, Alison

While the rigour of building formal models brings significant benefits, formal reasoning remains a major barrier to the wider acceptance of formalism within the development of software intensive systems. In our work we abstract away from the complexities of low-level proof obligations, providing high-level modelling guidance. We have achieved this through a combination of techniques from Artificial Intelligence, i.e. planning, proof-failure analysis, automated theory formation, and Formal Methods, i.e. formal modelling, proof and simulation. Our aim is to increase the accessibility and productivity of Formal Methods—allowing smart designers to make better use of their time.

3.15 HipSpec: Theory Exploration for Automating Inductive Proofs

Moa Johansson (Chalmers UT – Göteborg, SE)

Joint work of Claessen, Koen; Johansson, Moa; Rosén, Dan; Smallbone, Nicholas
Main reference K. Claessen, M. Johansson, D. Rosén, and N. Smallbone, "HipSpec: Automating inductive proofs of program properties," in *IJCAR Workshop on Automated Theory eXploration (ATX 2012)*, Manchester, UK, July 2012.
URL http://web.student.chalmers.se/~danr/hipspec-atx.pdf

HipSpec is an inductive theorem prover for proving properties about Haskell programs [1]. It implements a novel bottom-up approach to lemma discovery, where theory exploration is used to first derive a background theory consisting of potentially useful lemmas about available functions and datatypes.

HipSpec consists of several sub-systems: Hip is an inductive theorem prover. It translates Haskell function definitions to first order logic and applies induction to given conjectures.

Resulting proof obligations are passed to an off the shelf prover (for instance E or Z3). QuickSpec is responsible for generating candidate lemmas about available functions and datatypes. It generates terms which are divided up into equivalence classes using counter-example testing. From these equivalence classes, equations can be derived. These are passed to Hip for proof. Those that are proved are added to the background theory and may be used in subsequent proofs.

Initial results are encouraging, HipSpec performs very well compared to other automated inductive theorem provers such as IsaPlanner, Zeno and Dafny.

HipSpec is available online from: http://github.com/danr/hipspec.

References

1 K. Claessen, M. Johansson, D. Rosén, and N. Smallbone. HipSpec: Automating inductive proofs of program properties. In *IJCAR Workshop on Automated Theory eXploration (ATX 2012)*, Manchester, UK, July 2012.

3.16 Formalism: pitfalls and overcoming them (with AI?)

Cliff B. Jones (Newcastle University, GB)

This was a general, opening, talk. Based partly on recent experience in the EU-funded DEPLOY project, I described typical industrial figures where decent heuristics might automatically discharge over 90 percent of the required proof obligations (POs) but that this can still leave a large enough collection of proof tasks needing hand-assisted proofs that industrial engineers find it a disincentive to use formal tools. The issue is of course not going to go away: any set of heuristics will have their limitations.

The good news is that such collections of undischarged POs appear to fall into families and that a single idea will be the key to discharging many POs. The "ideas" are sometimes expressible as high level strategies or can be captured as lemmas (or shapes thereof). A useful approach is therefore to try to capture these key ideas whilst an expert is doing one proof from the family and to use this to then obtain automatic proofs of the remaining tasks in that family. The discovery and replay of these ideas looks like an interesting AI challenge.

There is an additional payoff when, as so often happens, specifications change and POs are regenerated.

These ideas are behind (and are evolving in) the UK research project AI4FM (see http://www.ai4fm.org).

3.17 Languages and States: another view of "Why"

Cliff B. Jones (Newcastle University, GB)

Like the view in my introductory talk, this also relates to ongoing work in the AI4FM project.

If we are to capture the key ideas in the interaction between a user and a theorem proving system, we essentially have to have a "language" for high-level strategies. The talk put the point of view that, if one wants to design a language, the best way to start is to think about the "state" that the language constructs can manipulate. In AI4FM we talk about trying to capture the "Why" of what the user is doing. (The talk by Andrius Velykis showed an architecture for snooping on the interactions between user and theorem prover.) I sketched some points about a state for "Models of Why" but the details are less important (and are anyway still changing) than the general idea of starting thoughts about the design of our future system to learn from experts by discussing its state.

Since Ursula Martin had made kind references to the "mural" system build in the 1980s I took the opportunity to dig out some screen shots. This was an experiment in the design of of an interaction style. As [1] shows, this system was also designed from its formal specification (the cited book is now out of print but freely available on-line at http://homepages.cs.ncl.ac.uk/cliff.jones/ftp-stuff/mural.pdf).

References

1 C. B. Jones, K. D. Jones, P. A. Lindsay, and R. Moore. *mural: A Formal Development Support System.* Springer, 1991.

3.18 Machine Learning for the Working Logician

Ekaterina Komendantskaya (University of Dundee, GB)

Joint work of Komendantskaya, Ekaterina; Grov, Gudmund; Bundy, Alan
Main reference G. Grov, E. Komendantskaya, and A. Bundy, "A statistical relational learning challenge – extracting proof strategies from exemplar proofs," in *ICML'12 Workshop on Statistical Relational Learning (SRL-2012)*, Edinburgh, UK, July 2012.
URL http://www.computing.dundee.ac.uk/staff/katya/srl12.pdf

The talk consisted of two parts: Part 1 discussed the motivation for using Statistical Machine Learning in Automated Theorem Proving (ATP). In particular, the following research question was chosen for discussion: how can we identify application areas within automated theorem proving where machine-learning will be both genuinely needed and trusted by the ATP users?

Part 2 addressed this question by showing results of some recent experiments on data-mining first-order proofs. Coinductive proof trees for first-order logic programs were data-mined using Neural Networks and SVMs. This proof data-mining method allowed to solve five types of proof-classification problems: recognition of well-formed proofs, proofs belonging to the same proof-family, well-typed proofs, as well as proofs from a success family and a well-typed family of proofs.

Discovery of various proof families was identified as the most promising of these. This provided my tentative answer to the research question posed in the beginning.

References

1 G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data.* Neural Information Processing Series. MIT Press, 2007.

2 A. S. D. A. Garcez, K. Broda, and D. M. Gabbay. *Neural-Symbolic Learning Systems: Foundations and Applications.* Perspectives in Neural Computing. Springer, 2002.

3 G. Grov, E. Komendantskaya, and A. Bundy. A statistical relational learning challenge – extracting proof strategies from exemplar proofs. In *ICML'12 Workshop on Statistical Relational Learning (SRL-2012)*, Edinburgh, UK, July 2012.

4 J. Denzinger, M. Fuchs, C. Goller, and S. Schulz. Learning from previous proof experience: A survey. Technical report, Fachbereich Informatik, 1999.

5 J. Denzinger and S. Schulz. Automatic acquisition of search control knowledge from multiple proof attempts. *Inf. Comput.*, 162(1-2):59–79, 2000.

6 H. Duncan. *The Use of Data-Mining for the Automatic Formation of Tactics.* PhD thesis, School of Informatics, University of Edinburgh, 2007.

7 L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning.* Adaptive Computation and Machine Learning. MIT Press, 2007.

8 E. Komendantskaya, R. Almaghairbe, and K. Lichota. ML-CAP: the manual, software and experimental data sets. http://www.computing.dundee.ac.uk/staff/katya/MLCAP-man, 2012.

9 J. W. Lloyd. *Logic for Learning: Learning Comprehensible Theories from Structured Data.* Cognitive Technologies. Springer, 2003.

10 E. Tsivtsivadze, J. Urban, H. Geuvers, and T. Heskes. Semantic graph kernels for automated reasoning. In *11th SIAM International Conference on Data Mining (SDM 2011)*, pages 795–803. SIAM / Omnipress, 2011.

11 J. Urban, G. Sutcliffe, P. Pudlák, and J. Vyskočil. MaLARea SG1 – machine learner for automated reasoning with semantic guidance. In A. Armando, P. Baumgartner, and G. Dowek, editors, *4th International Joint Conference on Automated Reasoning (IJCAR 2008)*, volume 5195 of *LNCS*, pages 441–456. Springer, 2008.

3.19 Is there a human to save the model, proof?

Thierry Lecomte (ClearSy – Aix-en-Provence, FR)

Checking a model against properties is a demanding process, as it requires to cope with state-of-the-art demonstrators (theorem provers, tableaux-method, model checkers). In several cases, the demonstrator is not able to complete the demonstration and the human operator is in charge of finding a way to help the tool efficiently. As of today, if the demonstrator is not able to complete the proof, most of the time all the proof mechanisms have to be disabled, leaving the human operator to use this tools just as a sophisticated calculator (built-in mechanisms automatically going to a wrong direction).

However there is room for improvement at input level:

- requirements are usually not abstract enough,
- models need to be adapted to proof tools,
- the human operator needs abstraction skills to deal properly with modelling.

The exception is for data validation where ProB model-checker is strong enough to handle 100k Excel cells and 200 rules, without any human intervention.

References

1 T. Lecomte, L. Burdy, and M. Leuschel. Formally checking large data sets in the railways. In *Workshop on the experience of and advances in developing dependable systems in Event-B (DS-Event-B-2012)*, Kyoto, Japan, November 2012.

3.20 Induction and coinduction in an automatic program verifier

K. Rustan M. Leino (Microsoft Research – Redmond, US)

Main reference K. R. M. Leino. "Dafny: An automatic program verifier for functional correctness," in E. M. Clarke and A. Voronkov, editors, *16th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-16)*, volume 6355 of *LNCS*, pp. 348–370. Springer, 2010.
URL http://research.microsoft.com/en-us/projects/dafny/

Program verifiers are good tools for verifying programs. Theorem provers are good tools for proving theorems. But the line between the two kinds of tools often gets blurry. Theorem provers can also be used to verify programs and program verifiers can also be used to prove theorems.

In this talk, I show a number of programs whose verification also requires some lemmas. The programs and lemmas are both proved using the program verifier Dafny [1]. I also demonstrate how to manually write inductive proofs in Dafny and show Dafny's automatic induction tactic [2]. Moreover, I show some experimental features for dealing with co-induction.

I expect that some AI techniques developed in the theorem proving and AI communities could be useful in program verifiers like Dafny.

References

1 K. R. M. Leino. Dafny: An automatic program verifier for functional correctness. In E. M. Clarke and A. Voronkov, editors, *16th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-16)*, volume 6355 of *LNCS*, pages 348–370. Springer, 2010.

2 K. R. M. Leino. Automating induction with an SMT solver. In V. Kuncak and A. Rybalchenko, editors, *13th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2012)*, volume 7148 of *LNCS*, pages 315–331. Springer, 2012.

3.21 ProB Tool Demonstration and Thoughts on Using Artificial Intelligence for Formal Methods

Michael Leuschel (Universität Düsseldorf, DE)

In this talk I gave a demonstration of the validation tool ProB [1, 2]. In particular, I concentrate on model checking and the constraint solving kernel and how this links with proof and intelligent search. In particular, I believe that proof, model checking and constraint solving should go hand-in-hand, and that tackling high-level (higher-order) formalisms such as B is extremely challenging, but provides more potential for intelligent search.

References

1 M. Leuschel and M. J. Butler. ProB: A model checker for B. In K. Araki, S. Gnesi, and D. Mandrioli, editors, *FME 2003: Formal Methods*, volume 2805 of *LNCS*, pages 855–874. Springer, 2003.

2 M. Leuschel, J. Falampin, F. Fritz, and D. Plagge. Automated property verification for large scale B models with ProB. *Formal Asp. Comput.*, 23(6):683–709, 2011.

3.22 The Use of Rippling to Automate Event-B Invariant Preservation Proofs

Yuhui Lin (University of Edinburgh, GB)

Main reference Y. Lin, A. Bundy, G. Grov, "The use of rippling to automate Event-B invariant preservation proofs," in A. Goodloe and S. Person, editors, *NASA Formal Methods*, volume 7226 of *LNCS*, pages 231–236. Springer, 2012.
URL http://dx.doi.org/10.1007/978-3-642-28891-3_23

Proof automation is a common bottleneck for industrial adoption of formal methods. In Event-B a significant proportion of proof obligations which requires human interaction falls into a family called invariant preservation. In this talk we show that rippling can increase the automation of proof in this family, and extend this technique by combining two existing approaches.

3.23 Discovery of Invariants through Automated Theory Formation

Maria Teresa Llano Rodriguez (Heriot-Watt University Edinburgh, GB)

Joint work of Llano Rodriguez, Maria Teresa; Ireland, Andrew; Pease, Alison
Main reference M. T. Llano, A. Ireland, A. Pease, "Discovery of invariants through automated theory formation," in *15th International Refinement Workshop (Refine 2011)*, volume 55 of *EPTCS*, pp. 1–19, Limerick, Ireland, June 2011.
URL http://dx.doi.org/10.4204/EPTCS.55.1

Refinement is a powerful mechanism for mastering the complexities that arise when formally modelling systems. Refinement also brings with it additional proof obligations – requiring a developer to discover properties relating to their design decisions. With the goal of reducing this burden, we have investigated how a general purpose theory formation tool, HR, can be used to automate the discovery of such properties within the context of Event-B. This gave rise to an integrated approach to automated invariant discovery. In addition to formal modelling and automated theory formation, our approach relies upon the simulation of system models as a key input to the invariant discovery process as well as automated proof-failure analysis.

3.24 Abstraction in Formal Verification and Development

Christoph Lueth (DFKI Bremen, DE)

Besides correctness, another aspect of formal verification is that it turns the development into a formal object which we can reason over, and which we can manipulate. One possibility is abstraction, i.e. the process of making a given proof or development applicable in different situations. We talk about three different kind of abstractions here, datatype abstraction, development abstraction, and structure abstraction, to highlight potential avenues of research and their uses.

3.25 A study of cooperative online math

Ursula Martin (Queen Mary University of London, GB)

Joint work of Martin, Ursula; Pease, Alison
Main reference A. Pease, U. Martin, "Seventy four minutes of mathematics: An analysis of the third Mini-Polymath project," in *AISB/IACAP 2012, Symposium on Mathematical Practice and Cognition II*, pp. 19–29, Birmingham, UK, July 2012.
URL http://homepages.inf.ed.ac.uk/apease/papers/seventy-four.pdf

Blogs, question answering systems and "crowdsourced" proofs provide effective new ways for groups of people, who may be unknown to each other, to use the internet to conduct mathematical research. They also provide a rich resource to shed light on mathematical practice and how mathematics advances, with the internet making visible and codified matters which have heretofore been ephemeral to study.

We discuss the first steps in such a research programme, looking at two examples to see what we can learn about mathematics as practiced on the internet. Does it differ from non-internet practice, and does it support or refute traditional theories of how mathematics is made, and who makes it, in particular those of Lakatos, or does it suggest new ones.

Polymath supports "crowdsourced proofs" and provides a structured way for a number of people to work on a proof simultaneously, capturing not only the final result, but also the discussion, missteps, informal arguments and social mechanisms in use along the way. Mathoverflow supports asking and answering research level mathematical questions and provides 25 thousand mathematical conversations for analysis, again providing a record of the informal mathematical activity that goes into answering them, and the social processes underlying production, acceptance or rejection of "answers". We look at a sample of questions about algebra, and provide a typology of the kinds of questions asked, and consider the features of the discussions and answers they generate.

We hope that this work will lead to collaboration with the formal methods community, in understanding proof or in researching proof archives to compare and contrast with the maths community.

3.26 TLA+ Proofs

Stephan Merz (INRIA – Nancy, FR)

Joint work of Cousineau, Denis; Doligez, Damien; Lamport, Leslie; Merz, Stephan; Ricketts, Daniel; Vanzetto, Hernán
Main reference D. Cousineau, D. Doligez, L. Lamport, S. Merz, D. Ricketts, H. Vanzetto, "TLA$^+$ proofs," in D. Giannakopoulou and D. Méry, editors, *18th International Symposium on Formal Methods (FM 2012)*, volume 7436 of *LNCS*, pp. 147–154. Springer, 2012.
URL http://arxiv.org/abs/1208.5933

The TLA$^+$ Proof System (TLAPS) is mainly intended for the deductive verification of (models of) distributed algorithms. At the heart of TLAPS lies a hierarchical and explicit proof language. Leaf proof steps are discharged by different automatic backend provers. In order to ensure the overall coherence of a proof, backend provers may provide a detailed proof that can be checked within Isabelle/TLA$^+$, an encoding of TLA$^+$ as an object logic in the logical framework Isabelle.

The system has been designed for developing large interactive proofs. In particular, the GUI provides commands for reading and writing hierarchical proofs by letting the user focus on part of a proof. TLAPS uses a fingerprinting mechanism to store proof obligations and their status. It thus avoids reproving previously proved obligations, even after a model or a proof has been restructured, and it facilitates the analysis of what parts of a proof are affected by changes in the model.

The paper is a longer version of an article published at FM 2012.

3.27 Case Based Specifications – reusing specifications, programs and proofs

Rosemary Monahan (Nat. University of Ireland, IE)

Joint work of Monahan, Rosemary; O'Donoghue, Diarmuid
URL http://www.cs.nuim.ie/staff/rosemary
URL http://www.cs.nuim.ie/staff/dod

Many software verification tools use the design-by-contract approach to annotate programs with assertions so that tools, such as compilers, can generate the proof obligations required to verify that a program satisfies its specification. Theorem provers and SMT solvers are then used to, often automatically, discharge the proof obligations that have been generated.

While verification tools are becoming more powerful and more popular, the major difficulties facing their users concern learning how to interact efficiently with these tools. These issues include learning how to write good assertions so that the specification expresses what the program must achieve and writing good implementations so that the program verification is easily achieved [4, 5]. In this presentation we discuss guiding the user in these aspects by making use of verifications from previously written programs. That is by finding a similar or analogous program to the one under development, we can apply the same implementation and specification approaches. Our strategy is to use a graph-based representation of a program and its specification as the basis for identifying similar programs.

Graph-matching was identified as the key to elucidating analogical comparisons in the seminal work on Structure Mapping Theory [1]. By representing two sets of information as relational graphs, structure mapping allows us to generate the detailed comparison between the two concepts involved. So given two graphs we can identify the detailed comparison using graph matching algorithms. For one application of graph matching to process geographic and spatial data see [3]. However, we may not always have an identified "source" to apply to our given problem. Thus, more recent work has taken a problem description, searching through a number of potentially analogous descriptions, to identify the most similar past solution to that problem [2].

Our work will develop a graph matching framework for program verification. The associated tools will operate on a collection of previously verified programs, identifying specifications that are similar to those under development. The program associated with this "matching specification" will guide the programmer to construct a program that can be verified as correct with respect to the given specification. Likewise, the strategy can be applied when the starting point is a program for which we need to construct a correct specification.

The core matching process can be thought of as a K+J colored graph-matching algorithm, which coupled with analogical transfer will re-apply the old solution to a new problem. Graphs can be flow graphs, UML diagrams, parse trees or another representation of a specification. Therefore, an iterative implementation of a sigma function (say) using tail recursion will be analogous to another recursive implementation—possibly using head-recursion. Similarly, iterative calculations of the same function using while and for loops will be more analogous to one another. Identical graph matching (isomorphism) will identify exact matches, given the representation, while non-identical (homomorphic) matches will identify the best available solutions.

In summary, our work will help to make software specification and verification more accessible to programmers by guiding users with knowledge of previously verified programs. A graphical representation of the specification, coupled with graph matching algorithms, is used as the basis of an analogical approach to support reuse of specification strategies.

References

1 D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.

2 D. P. O'Donoghue and M. T. Keane. A creative analogy machine: Results and challenges. In M. L. Maher, K. Hammond, A. Pease, R. Pérez y Pérez, D. Ventura, and G. Wiggins, editors, *4th International Conference on Computational Creativity (ICCC 2012)*, pages 17–24, 2012.

3 D. P. O'Donoghue, A. J. Bohan, and M. T. Keane. Seeing things: Inventive reasoning with geometric analogies and topographic maps. *New Generation Comput.*, 24(3):267–288, 2006.

4 K. R. M. Leino and R. Monahan. Automatic verification of textbook programs that use comprehensions. In *9th Workshop on Formal Techniques for Java-like Programs (FTfJP 2007), ECOOP 2007 Workshop*, Berlin, Germany, July 2007.

5 K. R. M. Leino and R. Monahan. Dafny meets the verification benchmarks challenge. In G. T. Leavens, P. W. O'Hearn, and S. K. Rajamani, editors, *Verified Software: Theories, Tools, Experiments (VSTTE 2010)*, volume 6217 of *LNCS*, pages 112–126. Springer, 2010.

3.28 Can AI Help ACL2?

J Strother Moore (University of Texas at Austin, US)

Joint work of Moore, J Strother; Kaufmann, Matt; Boyer, Robert
Main reference M. Kaufmann, P. Manolios, J. S. Moore, "*Computer-Aided Reasoning: An Approach*," volume 3 of *Advances in Formal Methods.* Kluwer Academic Publishers, 2000.
URL http://www.cs.utexas.edu/users/moore/acl2

ACL2 stands for "A Computational Logic for Applicative Common Lisp," and is a fully integrated verification environment for functional Common Lisp. I briefly mentioned some of its industrial applications, primarily in microprocessor design, especially floating-point unit design, and security. ACL2 is used to prove functional correctness of industrial designs. I then demonstrated an ACL2 model of the Java Virtual Machine highlighting (a) the size and scale of the formal model, (b) the fact that it was executable and thus was a JVM engine, and (c) ACL2 can be configured so that code proofs are often automatic. I then turned to how AI could help the ACL2 user, including: facilitating proof maintenance in the face of continued evolution of designs; facilitating team interaction in team based proofs (e.g., automatically informing team member A that team member B has already proved a lemma that seems to be related to the one A is trying to formulate); intelligent, semantic-based search of the data bases of participating users exploiting the common, formal language used to express concepts; concept and lemma formation; and inductive generalization. I concluded with the lament that many people attracted to modern AI seem to be put off by formality. This is not to say they are imprecise, only that they seem more interested in studying less rigid systems than formal proof systems. I see theorem proving as a game, like chess: there are fixed rules that every game must follow, but there is plenty of room for statistical learning, probabilistic methods, and intelligent/heuristic strategies as long as they ultimately result in the recommendation of helpful legal moves. The title of this talk suggests there is a question of whether AI could help ACL2. But in fact, there is no question that it could, if the right sort of expertise is brought to bear on the problem.

3.29 Boogie Verification Debugger

Michał Moskal (Microsoft Research – Redmond, US)

Joint work of Moskal, Michał; Leino, K. Rustan M.; Le Goues, Claire
Main reference C. L. Goues, K. R. M. Leino, M. Moskal, "The Boogie Verification Debugger (tool paper)," in G. Barthe, A. Pardo, and G. Schneider, editors, *9th International Conference on Software Engineering and Formal Methods (SEFM 2011)*, volume 7041 of *LNCS*, pp. 407–414. Springer, 2011.
URL http://research.microsoft.com/en-us/um/people/moskal/pdf/bvd.pdf

The Boogie Verification Debugger (BVD) is a tool that lets users explore the potential program errors reported by a deductive program verifier. The user interface is like that of a dynamic debugger, but the debugging happens statically without executing the program. BVD integrates with the program verification engine Boogie. Just as Boogie supports multiple language front-ends, BVD can work with those front-ends through a plug-in architecture. BVD plugins have been implemented for two state-of-the-art verifiers, VCC and Dafny.

3.30 Formal software verification in avionics

Yannick Moy (AdaCore, Paris, FR)

Joint work of Comar, Cyrille; Kanig, Johannes; Moy, Yannick
Main reference C. Comar, J. Kanig, Y. Moy, "Integrating formal program verification with testing," in J. Sifakis and J. Botti, editors, *Embedded Real Time Software and Systems (ERTS^2 2012)*, 2012.
URL http://www.open-do.org/wp-content/uploads/2011/12/hi-lite-erts2012.pdf

Certification of civilian avionics software is a very costly process, partly due to the pervasive use of testing. To reduce these costs, the avionics industry is now looking at using formal methods instead of testing, which is allowed by the new version of the certification standard (DO-178C). Based on previous experience with formal verification of avionics software, we propose a methodology and tools based on formal methods to address the DO-178C objective of verifying that code correctly implements low-level requirements. Our approach simplifies the adoption of formal verification by using the executable semantics of assertions familiar to engineers, by relying on a combination of automatic proof and testing, and by having tools that support the development of specifications. We take advantage of the latest version (2012) of the Ada language, which includes many specification features like pre- and postconditions for subprograms.

3.31 What I've Learned from the VFS Challenge thus far

José Nuno Oliveira (Universidade de Minho – Braga, PT)

Joint work of Oliveira, José Nuno; Ferreira, Miguel Alexandre
Main reference J. N. Oliveira, M. A. Ferreira, "Alloy meets the algebra of programming: A case study," *IEEE Transactions on Software Engineering*, 99(PrePrints), 2012.
URL http://dx.doi.org/10.1109/TSE.2012.15

The question "Is Abstraction the Key to Computing?" (Jeff Kramer) is central to formal methods. The "yes!" answer is widely accepted, but perhaps there are other unanswered questions, for instance: "Are we using the right notation, formal language?"

In this talk I described how the experience in handling a concrete case study—the Verified File System (VFS) challenge in computing put forward by Joshi and Holzmann—changed my way of doing FMs, starting from a grand scale tool-chain involving several notations and tools (model checkers, animators, theorem-provers) to a minimalist approach, relying on quantifier-free relational notation and the Alloy model checker only.

Another question (central to the seminar) is "How can AI help?" The same experience points towards two fields where this may well happen in the future: requirements "engineering" based on semantics-rich boilerplates and ontologies of invariant theories specified around what in the talk was referred to as "the 4-relation invariant pattern".

3.32 The Need for AI in Software Engineering – A Message from the Trenches

Stephan Schulz (TU München, DE)

I discuss existing structured software development processes as used by a medium sized system integrator in the field of Air Traffic Control, highlighting the variability of the processes driven by both customer capability and market demands. There often are large gaps between current practice and the prerequisites for large-scale application of formal methods.

I suggest how AI can help to bridge some of these gaps and can improve industrial software development processes as used by SMEs. One particular hotspot which could profit from AI techniques is requirements engineering. In requirements capture and engineering, AI techniques can help structuring the requirements, guiding the refinement process, and point out where specifications appear to be unclear or contradictory.

Another significant problem is the large amount of existing legacy code, often embodying decades worth of domain knowledge and testing, but usually not written to today's standards. The ability to re-engineer existing code bases, documenting dependencies, side effects, pre- and postconditions, and invariants, would make existing libraries much more amendable to be combined with more rigorously developed new code.

3.33 Finding and Responding to Failure in Large Formal Verifications

Mark Staples (NICTA, AU)

The L4.verified project has completed the formal verification, to the level of C source code, of the full functional correctness of the seL4 microkernel [2]. The project proceeded in several phases, with internal iterations, and ongoing maintenance [1]. The team can now claim there are zero bugs in the microkernel, subject to assumptions and conditions. The scale and detail in the project raise challenges of potential interest for Artificial Intelligence (AI) and Formal Methods (FM). I discuss two of these.

Firstly, changes to the specification, design, code, and invariants are almost inevitable in large formal verification projects, because of bug fixes or enhancements. Changes mean the system must be re-verified, and all lemmas must be re-proved. The automated re-proof of some lemmas may fail, because they are no longer true and must be reworked, or because the proof scripts are too fragile and must be reworked. Reverification is a management and technical challenge for which AI techniques may be relevant. Specific challenges include: In what order should lemmas be reworked? How can proof scripts be made more robust to minor changes to lemmas?

Secondly, the existence of extra-logical "gaps" between formal models and the real world (actual requirements and implementations) is well known in the FM community. Formal methods are empirical too, because the properties proved about software are claims about how it will behave and satisfy requirements in the world. Nonetheless, it is less well known how to address these gaps. The L4.verified team identified some extra-logical assumptions, including that assembly-code, C compiler, and hardware are correct. What other key assumptions might there be, and how can we identify them? Can heuristic search techniques help to identify or test such assumptions?

References

1 J. Andronick, D. R. Jeffery, G. Klein, R. Kolanski, M. Staples, H. Zhang, and L. Zhu. Large-scale formal verification in practice: A process perspective. In M. Glinz, G. C. Murphy, and M. Pezzè, editors, *34th International Conference on Software Engineering, (ICSE 2012)*, pages 1002–1011. IEEE, 2012.

2 G. Klein, J. Andronick, K. Elphinstone, G. Heiser, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Winwood. seL4: formal verification of an operating-system kernel. *Commun. ACM*, 53(6):107–115, 2010.

3.34 Formal Verification of QVT Transformations

Kurt Stenzel (Universität Augsburg, DE)

Joint work of Stenzel, Kurt; Moebius, Nina; Reif, Wolfgang
Main reference K. Stenzel, N. Moebius, W. Reif, "Formal verification of QVT transformations for code generation," in J. Whittle, T. Clark, and T. Kühne, editors, *14th International Conference on Model Driven Engineering Languages and Systems (MODELS 2011)*, volume 6981 of *LNCS*, pp. 533–547. Springer, 2011.
URL http://dx.doi.org/10.1007/978-3-642-24485-8_39

We present a formal calculus for operational QVT. The calculus is implemented in the interactive theorem prover KIV and allows to prove properties of QVT transformations for arbitrary meta models.

Additionally we present a framework for provably correct Java code generation. The framework uses a meta model for a Java abstract syntax tree as the target of QVT transformations. This meta model is mapped to a formal Java semantics in KIV. This makes it possible to formally prove with the QVT calculus that a transformation always generates a Java model (i.e. a program) that is type correct and has certain semantical properties. The Java model can be used to generate source code by a model-to-text transformation or byte code directly.

Finally, we report on experiences with the development of the new calculus.

3.35 Modeling and Proving

Werner Stephan (DFKI – Saarbrücken, DE)

Joint work of Stephan, Werner; Cheikhrouhou, Lassaad; Langenstein, Bruno; Nonnengart, Andreas

To turn Formal Methods into an engineering discipline (beyond use in the academic community) modeling has to be taken seriously: attack the real problems, make explicit (hidden) assumptions, follow certain guidelines, provide (informal) interpretations for the interaction with non-formal parts, and allow for a third party assessment. To that end we still need modeling experts. Although their expertise might be domain specific they will be able to perform proofs in state-of-the-art interactive proof systems. However, typically they will not be able to develop complete (proof) strategies. Interactive systems offer the flexibility that is often needed for adequate models. Considerable progress has been made with respect to partial automation, user interaction, and the engineering of interactive proofs. For example our proof strategy for protocol verification selects around 85% of the steps automatically. In simple standard case it gets close to 100%. Unfortunately sophisticated real world problems in many cases tend to be 'out of range' of a given fixed strategy (to a varying degree). Adapting strategies still requires (sometimes a lot of) paper work and is completely out of range for users. The challenge (dream) therefore is to support the development of strategies within the system in an interactive way such that at least for simpler cases even users are able to perform these modifications themselves.

3.36 Overview of CSP||B for railway modelling

Helen Treharne (University of Surrey, GB)

Joint work of Moller, Faron; Nguyen, Hoang Nga; Roggenbach, Markus; Schneider, Steve; Treharne, Helen

CSP||B is a formal approach that has been developed at the University of Surrey over a number of years; it combines two well-established formal methods: CSP and B. At the heart of the method is a compositional verification framework. Our recent work has been using CSP||B in the verification of railway systems in collaboration with the University of Swansea. Our motivation is to develop a modelling and verification approach accessible to railway engineers: it is vital that they can validate the models and verification conditions, and—in the case of design errors—obtain comprehensible feedback. In this talk, we presented an overview of the style of formalization that we have adopted. It is aligned with the way engineers think about railway systems, and we have involved our industrial partner in detailed discussions at every stage. The railway models can become large when complex track plans are being modelled. We also discussed our ongoing work which is focusing on identifying abstractions of track plans so that model checking complex track plans is possible.

References

1 F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, and H. Treharne. Combining event-based and state-based modelling for railway verification. Technical Report CS-12-02, University of Surrey, 2012.

2 F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, and H. Treharne. Railway modelling in CSP||B: the double junction case study. In *12th International Workshop on Automated Verification of Critical Systems (AVOCS 2012)*, Bamberg, Germany, September 2012. (to appear).

3.37 AI via/for Large Mathematical Knowledge Bases

Josef Urban (Radboud University Nijmegen, NL)

Joint work of Urban, Josef; Vyskočil, Jiří
Main reference J. Urban, J. Vyskočil, "Theorem Proving in Large Formal Mathematics as an Emerging AI Field," Dagstuhl Preprint Archive, arXiv:1209.3914 [cs.AI], 2012.
URL http://arxiv.org/abs/1209.3914

The talk introduces the recently appeared large mathematical knowledge bases as a suitable repository for combining deductive and inductive AI methods. Several examples of ATP/AI systems working in this setting are given, including the Machine Learner for Automated Reasoning (MaLARea) and the Machine Learning Connection Prover (MaLeCoP). Some lessons learned from working on such systems are discussed and some future work topics are mentioned.

3.38 Capturing and Inferring the Proof Process (Part 2: Architecture)

Andrius Velykis (Newcastle University, GB)

Main reference A. Velykis, "Inferring the proof process," in C. Choppy, D. Delayahe, and K. Klaï, editors, *FM2012 Doctoral Symposium*, Paris, France, August 2012.
URL http://www.cs.ncl.ac.uk/publications/trs/papers/1344.pdf

Interactive theorem proving can be used to verify formal models and specifications as well as justify their development process. A large portion of the proof can be automated using general heuristics available in state-of-the-art automatic theorem provers, but significant manual work still gets left for human experts.

In this talk we ask how enough information can be collected from an interactive formal proof to capture an expert's ideas as a high-level proof process. Such information would then serve for extracting proof strategies to facilitate automation of similar proofs. We explore the question: what describes a proof process? The talk presents our take: we need structural information (e.g. proof granularity / multiple attempts) as well as proof meta-information (e.g. proof features). Furthermore, we present the architecture of a new ProofProcess framework, which is developed to support collecting and inferring the proof process (automatically, or by asking the expert). It aims to provide a generic way to capture proof process information from different theorem provers.

3.39 More Abstractions

Laurent Voisin (SYSTEREL – Aix-en-Provence, FR)

Based on 10+ years of formal modelling in industry, I advocate the use of domain theories and modelling patterns in system modelling.

When modelling a system, one has to somehow encode some domain data structures into some mathematical notation (e.g., set-theory for Event-B). This encoding is not trivial, except for very simple case studies. Inlining this encoding within a model makes it difficult to read and consequently difficult to prove. It is much better to separate the encoding in a separate file (i.e., a theory) which will describe the data structure, provide operators for updating it together with proof rules for reasoning about them. The model is then free from clutter and can be expressed at the same level of discourse as domain experts.

I think that AI could provide significant benefits by detecting when a model is not at the correct level of discourse and contains too much encoding. This could be detected by inspecting the model and assessing its intrinsic complexity. This would be particularly useful for beginners who usually have difficulty to separate concerns.

Another use of AI is to implement refinement plans (see paper by Grov, Ireland and Llano presented at ABZ 2012). In this setting, a failing proof is analysed with respect to some refinement patterns and the tool suggests amendment to the model that would allow to fix its proof. I think that it would be much more valuable if the refinement patterns would be in the form of generic models. The tool would then propose to instantiate the generic pattern and suggest ways to instantiate it (e.g., provide actual parameters). This would reuse not only the pattern but also its associated proof. The user would only have to prove that the actual parameters fulfill the pattern pre-conditions.

More generally, AI could be used to mine existing models to extract generic patterns from them. This would allow to build a library of recurring patterns. As for refinement plans, AI could also be used for guiding users within the library and help them select the appropriate patterns with respect to their modelling needs.

In conclusion, using both theories and generic model patterns makes models more easy to develop, read and prove, by allowing better reuse. AI could be of great help in assisting users for making the better use of these tools.

3.40 Finding Counterexamples through Heuristic Search

Martin Wehrle (Universität Basel, CH)

AI planning considers the task of automatically finding a sequence of actions such that applying these actions leads to a state that satisfies a given goal condition. A popular approach to solve planning tasks is based on heuristic search.

From an abstract point of view, AI planning is related to finding counterexamples in model checking. This talk shows the basic idea of computing distance heuristics automatically based on a general problem description, and shows how heuristic search techniques can be applied to finding counterexamples in model checking.

Participants

- Rob Arthan
Lemma 1 Ltd. – Reading, GB
- Serge Autexier
DFKI Bremen, DE
- Alan Bundy
University of Edinburgh, GB
- Simon Colton
Imperial College London, GB
- David Crocker
Escher Technologies – Aldershot, GB
- Jorge Cuellar
Siemens – München, DE
- Ewen W. Denney
NASA – Moffett Field, US
- Leo Freitas
Newcastle University, GB
- Dimitra Giannakopoulou
NASA – Moffett Field, US
- Gudmund Grov
University of Edinburgh, GB
- Reiner Hähnle
TU Darmstadt, DE
- Dieter Hutter
DFKI Bremen, DE
- Andrew Ireland
Heriot-Watt University Edinburgh, GB
- Moa Johansson
Chalmers UT – Göteborg, SE
- Cliff B. Jones
Newcastle University, GB
- Ekaterina Komendantskaya
University of Dundee, GB
- Thierry Lecomte
ClearSy – Aix-en-Provence, FR
- K. Rustan M. Leino
Microsoft – Redmond, US
- Michael Leuschel
Universität Düsseldorf, DE
- Yuhui Lin
University of Edinburgh, GB
- Maria Teresa Llano Rodriguez
Heriot–Watt University Edinburgh, GB
- Christoph Lüth
DFKI Bremen, DE
- Ursula Martin
Queen Mary University of London, GB
- Stephan Merz
INRIA – Nancy, FR
- Rosemary Monahan
Nat. University of Ireland, IE
- J Strother Moore
University of Texas at Austin, US
- Michał Moskal
Microsoft – Redmond, US
- Yannick Moy
AdaCore – Paris, FR
- José Nuno Oliveira
Univ. de Minho – Braga, PT
- Thomas Santen
European Microsoft Innovation Center – Aachen, DE
- Stephan Schulz
TU München, DE
- Volker Sorge
University of Birmingham, GB
- Mark Staples
NICTA, AU
- Kurt Stenzel
Universität Augsburg, DE
- Werner Stephan
DFKI – Saarbrücken, DE
- Helen Treharne
University of Surrey, GB
- Josef Urban
Radboud Univ. Nijmegen, NL
- Andrius Velykis
Newcastle University, GB
- Laurent Voisin
SYSTEREL – Aix-en-Provence, FR
- Martin Wehrle
Universität Basel, CH

Report from Dagstuhl Seminar 12272

Architecture-Driven Semantic Analysis of Embedded Systems

Edited by

Peter Feiler[1], Jérôme Hugues[2], and Oleg Sokolsky[3]

1 **Carnegie Mellon University – Pittsburgh, US,** `phf@sei.cmu.edu`
2 **ISAE – Toulouse, FR,** `jerome.hugues@isae.fr`
3 **University of Pennsylvania, Philadelphia, US,** `sokolsky@cis.upenn.edu`

Abstract

Architectural modeling of complex embedded systems is gaining prominence in recent years, both in academia and in industry. An architectural model represents components in a distributed system as boxes with well-defined interfaces, connections between ports on component interfaces, and specifies component properties that can be used in analytical reasoning about the model. Models are hierarchically organized, so that each box can contain another system inside, with its own set of boxes and connections between them.

The goal of Dagstuhl Seminar 12272 "Architecture-Driven Semantic Analysis of Embedded Systems" is to bring together researchers who are interested in defining precise semantics of an architecture description language and using it for building tools that generate analytical models from architectural ones, as well as generate code and configuration scripts for the system.

This report documents the program and the outcomes of the presentations and working groups held during the seminar.

Seminar 01.–06. July, 2012 – www.dagstuhl.de/12272

1998 ACM Subject Classification D.2.2 Design Tools and Techniques, D.2.4 Software/Program Verification, D.2.11 Software Architectures

Keywords and phrases Architectu Description Language, AADL, EAST-ADL, MARTE, Verification, Validation, Analysis, Embedded Systems, Model-Driven techniques

Digital Object Identifier 10.4230/DagRep.2.7.30

1 Executive Summary

Peter Feiler
Jérôme Hugues
Oleg Sokolsky

Architectural modeling of complex embedded systems is gaining prominence in recent years, both in academia and in industry. An architectural model represents components in a distributed system as boxes with well-defined interfaces, connections between ports on component interfaces, and specifies component properties that can be used in analytical reasoning about the model. Models are hierarchically organized, so that each box can contain another system inside, with its own set of boxes and connections between them. An architecture description language for embedded systems, for which timing and resource availability form an important part of the requirements, must describe resources of the system

Architecture-Driven Semantic Analysis of Embedded Systems, *Dagstuhl Reports*, Vol. 2, Issue 7, pp. 30–55
Editors: Peter Feiler, Jérôme Hugues, and Oleg Sokolsky
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

platform, such as processors, memories, communication links, etc. Several architectural modeling languages for embedded systems have emerged in recent years, including AADL, SysML, EAST-ADL, and the MARTE profile for UML.

In the context of model-based engineering (MBE) architectural modeling serves several important purposes:

An architectural model allows us to break the system into manageable parts and establish clear interfaces between these parts. In this way, we can manage complexity of the system by hiding the details that are unimportant at a given level of consideration; Clear interfaces between the components allow us to avoid integration problems at the implementation phase. Connections between components, which specify how components affect each other, help propagate the effects of change in one component to the affected components. Most importantly, an architectural model can be seen as a repository of the knowledge about the system, represented as requirements, design, and implementation artifacts, held together by the architecture. Such a repository enables automatic generation of analytical models for different aspects of the system, such as timing, reliability, security, performance, etc. Since all the models are generated from the same source, ensuring consistency of assumptions and abstractions used in different analyses becomes easier. The first two uses of architectural modeling have been studied in the research literature for a number of years. However, the coordination role of architectural modeling in MBE is just currently emerging. We expect this role to gain importance in the coming years. It is clear that realizing this vision of "single-source" MBE with an architectural model at its core is impossible without having first a clear semantics of the architecture description language.

The goal of the seminar is to bring together researchers who are interested in defining precise semantics of an architecture description language and using it for building tools that generate analytical models from architectural ones, as well as generate code and configuration scripts for the system. Despite recent research activity in this area to use semantic interpretation of architectural models for analytical model generation, we observe a significant gap between current state of the art and the practical need to handle complex models. In practice, most approaches cover a limited subset of the language and target a small number of modeling patterns. A more general approach would most likely require an interpretation of the semantics of the language by the tool, instead of hard-coding of the semantics and patterns into the model generator.

2 Table of Contents

Executive Summary

Overview of Talks

Working Groups

3 Overview of Talks

3.1 EAST-ADL – An Architecture-centric Approach to the Design, Analysis, Verification and Validation of Complex Embedded Systems

De-Jiu Chen (KTH – Stockholm, SE)

EAST-ADL is a domain specific Architecture Description Language (ADL) for safety-critical and software-intensive embedded systems. The language enables a formalized and traceable description of a wide range of engineering concerns throughout the entire lifecycle of systems. This makes it possible to fully utilize the leverage of state-of-the-art methods and tools for the development of correct-by-construction system functions and components in a seamless and cost efficient way.

This talk focuses on the recent advances of EAST-ADL in supporting the description, analysis, verification&validation of complex embedded systems for the purposes of requirements engineering, application design, and safety engineering. The approach is architecture centric as all behavior descriptions are formalized and connected to a set of standardized design artifacts existing at multiple levels of abstraction. This talk presents the language design, its theoretical underpinning and tool implementation. From a bigger perspective, the contribution makes it possible for embedded system and software developers to maintain various engineering concerns coherently, while exploiting mature state-of-the-art technologies from computer science and other related domains for a model-based design.

3.2 Model-Checking Support for AADL

Silvano Dal Zilio (LAAS – Toulouse, FR)

We present recent work on the extension of the Fiacre language with real-time constructs and real-time verification patterns. We will show how these enhancements have been used to implement a new version of a model-checking toolchain for AADL.

Fiacre is a formal language with support for expressing concurrency and timing constraints; its goal is to act as an intermediate format for the formal verification of high-level modeling language, such as UML profiles for system modeling. Essentially, Fiacre is designed both as the target of model transformation engines, as well as the source language of compilers into verification toolboxes, namely Tina and CADP.

Our motivations for extending Fiacre are to reduce the semantic gap between Fiacre and high-level description languages and to streamline our verification process. We will take the example of a transformation from AADL (and its behavioral annex) to Fiacre to explain the benefit of this new toolchain

3.3 Model-Based/ Platform-Based/Architecture-Driven Design of Cyber-Physical Systems

Patricia Derler (University of California – Berkeley, US)

This presentation focuses on recent efforts of including architectural properties into executable models in Ptolemy II. A programming model that includes some architecture information such as Sensors, Actuators, Platforms, Execution Time, Memory is Ptides, which is also implemented in Ptolemy. The talk describes the Ptides execution semantics and its implementation in Ptolemy. Evaluation of architectural properties and constraints is done via simulation.

3.4 On the mechanization of AADL subsets

Mamoun Filali-Amine (Paul Sabatier University – Toulouse, FR)

AADL (Architecture Analysis & Design Language) is a real-time specification language that focuses on the early analysis of the dynamic architecture of a system and the correctness of resource allocation described by the soft- ware / hardware mapping. Although, these aspects are precisely described in the standard, they need a formal expression in order to be able to verify their properties formally.

With respect to formal verification, it is interesting to consider two kind of properties:

- applicative properties like schedulability, absence of buffer overflows, deadlocks, starvation and more generally safety and liveness properties. The first ones are application independent while the last ones depend on the intended behavior of the application.
- properties related to the semantics of the language constructs, e.g., determinism, correctness of some model transformations (flattening, application of distribution strategies, ...).

While for applicative properties, model checking techniques have been widely and successfully applied, proof based techniques are still necessary to address properties which in general do not concern a fixed finite domain.

In this talk, we will present the different proof based attempts that we have conducted in order to mechanize some aspects of AADL.

- Semantics of basic AADL protocols related to threads and communication.
- Semantics of a basic AADL computation model.
- Mechanization of the basic semantics of the FIACRE language and proof support for a parameterized version.

3.5 Extended Literate Programming. Introducing the ⊔ (SquareCup) Language

Laurent Fournier (Rockwell Collins France, FR)

We introduce a universal typed sparse graph language called "⊔" (the *square cup* character, Unicode #2294, or `\sqcup` in TEX.). ⊔ extends the established *Literate Programming* (LP) approach with Model Driven Engineering (MDE) paradigm. Unlike *Literate Modeling* that basically produces documented diagrams reports, our *Extended Literate Programming* (ELP) proposal focus on satisfying the two original LP goals; full **automation** from the highest level declarative description to build a runnable machine code, and better **understanding** to produce high quality shared documentation, all in a literate form and within the same capture tool.

⊔ graph nodes support the useful notion of *port* and the ⊔ simple text syntax – concision and readability to compare for instance with XMI serialization format – can represent any kind of nested graphs. Supported formalisms vary but not limited to UML, *SysML*, *AADL graph*, *Simulink/Scicos/Scade blocs*, *State machine*, *Markov chain*, *Petri Nets*, *EntityRelation graph*, KAOS…For rendering, diagram positioning/routing attributes are excluded to rely only on automatic layout algorithms. All the semantics of a ⊔ graph is defined in nodes and arc *types* by a mapping to a particular generated code construction. Those *types* allow to build some DSL and Domain Specific Libraries of components for easy *Product Line Development.* ELP facilitates Requirement Engineering tasks like traceability, impact analysis, and transformations to design phase. Because each ⊔*-tool* instance natively provides a ⊔ models library and a *types* library available as a remote service, the designer can access to a set of cooperating/competing code generators, building a simple *Semantic Web* for software engineering. For any ⊔ graph, the same techniques apply to generate SVG code for web browsing, *TikZ* code for document embedded diagrams, than for generating compilable code. Furthermore, the node/arc content is a free *Unicode* string parsed as a *Python* interpreter so all downstream transformations works on the *Python Abstract Syntax Tree* (PAST). A tool framework is under development, using a web based editor (*CodeMirror*), optimized PDF rendering, a *Git* database on the cloud and *Python3* as glue language. Code generation will focus on AADL interpertation first. Unlike WYSIWYG (*What You See Is What you Get*) frameworks, the text nature of ⊔ models makes save, search, diff or merge operations easy and long term resistant.

The ⊔ project is currently hosted at https://github.com/pelinquin/u.

3.6 Software Component Architecture Model Analysis and Executable Generation using Semantic Language Layering

Serban Gheorhe (Edgewater Computer Systems Inc. – Ottawa, CA)

Without executable code generation, models of software systems are inevitably relegated to the role of exploration of design alternatives and design documentation. They will inevitably represent an alternative and potentially stale "source of truth" of the real running software, usually implemented in programming languages accepted in the industry. In this talk, we focus on executable software applications described using the AADL software component model and compliant with the AADL run-time execution semantics. The AADL run time semantics accommodates multiple possible computation model choices (synchronous/asynchronous, preemptable/non-preemptable, etc.). Also, target execution platforms currently in use have a wide variety of possible execution semantics. It would be extremely costly to build trusted code generators for all possible combinations affecting the AADL run-time execution semantics that also preserve at run-rime the formal temporal properties expressed and proven by static analysis on the AADL model. Our approach is to select a small anchor subset of the AADL run-time semantics, called the RTEdge AADL Microkernel subset, and use it as a trusted lower level semantic layer, encoded in a run-time executive middleware (called RTExec) available as a library on multiple execution platforms. Corresponding to the RTExec semantics we define an AADL language subset, called the RTEdge modeling language subset, which is executable via code generation and linking with the RTExec executive.

Given an AADL software system, we use this semantic layering to generate software executables for any execution platform that runs the RTExec: firstly by translating the AADL source into the intermediate RTEdge subset, thus obtaining an RTEdge model with equivalent execution semantics, secondly by generating target specific executables from the RTEdge model. Formal temporal properties expressed at the AADL run-time semantic layer can be mapped into equivalent temporal properties expressed and checked at the lower and fix semantic layer of the RTEdge subset, guaranteeing consistency between the analysis assumptions and the real execution semantics of the generated executable.

3.7 Architecture Evaluation @ Run-time: Problems, Challenges and Solutions

Lars Grunske (TU Kaiserslautern, DE)

The majority of innovations in modern technical systems are driven by software. Based on the research results of the software engineering community over the past decades we are able to develop software systems with immense complexity. However, concomitant with these increases in complexity, the quality demands also appear to be ever-growing. Probabilistic properties defined in probabilistic temporal logics are commonly applied to specify these quality demands and are especially suitable for performance, reliability, safety, and availability requirements. This talk will present approaches but also problems and challenges for run-time architecture evaluation strategies of these probabilistic properties.

3.8 Embedded System Architecture for Software Health Management

Gabor Karsai (Vanderbilt University, US)

As software increasingly becomes the main source of functionality and the ultimate tool for system integration in cyber-physical systems there is an increasing chance that imperfections in the design and implementation of the software need to be detected and mitigated at run-time. Such needs can be addressed by borrowing metaphors and techniques from the area of 'systems health management', where the concepts and technologies of anomaly detection, fault source isolation, and fault mitigation have been developed. Similarly to physical systems, a software health management (SHM) approach necessitates an architectural foundation: a component framework that defines units for fault management and fault containment, with precisely specified and controlled interfaces and interactions. Based on this foundation a highly reusable software health management layer can be constructed that maintains system functionality even when software defects appear. This layer is model-based: it consists of generic building blocks and algorithms that are configured via models. Using an architectural framework, with precisely defined component interaction semantics enables not only the implementation but the formal verification and analysis of the entire system. The talk introduces a motivating example, presents a component framework and how it was extended to support software health management, and concludes with a realistic case study.

3.9 Experience of Using Architecture Models in Civil Aviation Domain

Alexey Khoroshilov (Russian Academy of Sciences – Moscow, RU)

The talk presents an experience of building an AADL-based toolset for integrated modular avionics (IMA) design and integration. Features and an architecture of the toolset are described and the role of the architecture description language is discussed.

3.10 Hierarchy is Good For Discrete Time: a Compositional Approach to Discrete Time Verification

Fabrice Kordon (UPMC – Paris, FR)

Introduction

Model checking is now widely used as an automatic and exhaustive way to verify complex systems. However, this approach suffers from an intrinsic combinatorial explosion, due to both a high number of synchronized components and a high level of expressivity in these components.

With respect to the expressivity issue, we consider the particular problem of introducing explicit time constraints in the components of a system. In this modeling step, the choice of a time domain is important, impacting on the size of the resulting model, the class of properties which can be verified and the performances of the verification.

In this presentation, we show that hierarchical encoding of elementary components encapsulating labeled transition systems (LTS), synchronized by means of public transitions, is an efficient way to encode discrete time.

Instantiable Transition Systems

Instantiable Transition Systems (ITS) are a framework designed to exploit the hierarchical characteristics of SDD [5]. This structure is used to encode the state space, for the description of component based systems. ITS were introduced in [11]. Here are the main principles ITS rely on:

- ITS types (elementary) represent a LTS and export some public transitions that can be synchronized with other ITS types,
- composite ITS gather several ITS (composite or elementary) and propose a new interface that can be connected to some of the synchronized public actions of enclosed ITS,
- instanciation allows to create a number of entities having the same behavior. This emphasizes the description of regularities in distributed systems.

Encoding discrete time with ITS

The basic idea of using ITS to model discrete time is to propose an extra interface dedicated to time elapse [10]. This interface interacts with local clocks. When time elapse the same way all over the system, the elapse interfaces must be synchronized together. It is also possible to have local synchronization of clocks to model several timelines.

This presentation shows the main principles of this mechanism and illustrates it on a simple example from the literature. Then, we emphasize its interest in a small medical case study: the Body Area Network. Here, ITS are generated from a high-level language dedicated to the description of Wireless Sensor Networks: Verisensor [4].

3.11 Formal Semantics of AADL Component Behavior to Prove Conformance to Specification

Brian Larson (Multitude Corp., US)

Software can be more dependable than hardware. This requires programs, their specifications, and their executions are mathematical objects, so that conformance to specification of every execution can be formally verified. This talk presents an AADL annex sublanguage, Behavioral Languages for Embedded Systems with Software (BLESS) to formally define component behavior, based on the Behavior Annex (BA) language. BLESS adds Assertions to form proof outlines that can be transformed into complete formal proofs semi-automatically.

3.12 Software Architecture Modeling by Reuse, Composition and Customization

Ivano Malavolta (Univ. degli Studi di L'Aquila, IT)

While developing a complex system, it is of paramount importance to correctly and clearly specify its software architecture. Architecture Description Languages (ADLs) are the means to define the software architecture of a system. ADLs are strongly related to stakeholder concerns: they must capture all design decisions fundamental for systems stakeholders. From the earliest work in software architecture, the usefulness of expressing software architectures in terms of multiple views is well recognized. Architecture views represent distinct aspects of the system of interest and are governed by viewpoints which define the conventions for their construction, interpretation and use to frame specific system concerns. Most practicing software architects operate within an architecture framework which is a coordinated set of viewpoints, models and notations prescribed for them. As a matter of fact, stakeholders concerns vary tremendously, depending on the project nature, on the domain of the system to be realized, etc. So, even if current architecture frameworks are defined to varying degrees of rigor and offer varying levels of tool support, finding the right architecture framework that allows to address the various system concerns is both a risky and difficult activity. Therefore, an effective way to define and combine architectural elements into a suitable framework for effectively create architecture descriptions is still missing.

In this presentation, I propose an infrastructure for modeling the architecture of a software system by adapting existing architectural languages, viewpoints and frameworks to domain- and organization-specific features. Under this perspective, the proposed infrastructure allows architects to set up customized architectural frameworks by: (i) defining and choosing a set of viewpoints that adequately fit with the domain and features of the system being developed, (ii) automatically adapting existing architecture description languages to project-specific concerns, (iii) keeping architectural views within the framework synchronized, (iv) enabling consistency and completeness checks based on defined correspondences and rules among architectural elements. The proposed approach builds upon the conceptual foundations of ISO/IEC/IEEE 42010 for architecture description and it is generic with respect to the used architectural elements (i.e., views, viewpoints, languages, stakeholder's concerns, etc.).

The impact of the proposed approach is three-fold: (i) a novel approach is presented for architecting by reusing, composing and customizing existent architectural elements, (ii) a new composition mechanism is presented for extending architectural languages in a controlled fashion, (iii) a new mechanism for keeping architectural views in a consistent state is provided.

The proposed approach is realized through a combination of model transformations, weaving, and megamodeling techniques. The approach has been put in practice in different scenarios and has been evaluated in the context of a real complex system.

3.13 Architecture-Driven analysis with MARTE/CCSL

Frédéric Mallet (INRIA Sophia Antipolis, FR)

The UML Profile for Modeling and Analysis of Real-Time and Embedded systems promises a general modeling framework to design and analyze systems. Lots of works have been published on the modeling capabilities offered by MARTE, much less on verification techniques supported. The Clock Constraint Specification Language (CCSL), first introduced as a companion language for MARTE, was devised to offer a formal support to conduct causal and temporal analysis on MARTE models.

This presentation focuses on the analysis capabilities of MARTE/CCSL and describes a process where the logical description of the application is progressively refined to take into account the execution platforms (software and hardware architectures) and the environment constraints.

The approach is illustrated on two very simple examples where the architecture plays an important role. During the presentation, issues are raised on the expressiveness of CCSL, on the nature of properties that can be analyzed and on possible extensions.

3.14 Approximating physics in the design of technical systems

Pieter J. Mosterman (The MathWorks Inc. – Natick, US)

In the design of Cyber-Physical Systems, physics plays a crucial role. Models of physics at a macroscopic level often comprise differential and algebraic equations. These equations typically require computational approaches to derive solutions. Approximations introduced by the solvers that derive these solutions to a large extent determine the meaning of the models, in particular when discontinuities are included. In reasoning about models that are solved computationally it is therefore imperative to also model the solvers. This presentation shows how performance of a cyber- physical system may be affected by physics and conceptualizes the modeling of computational solvers. Opportunities that derive from the availability of solver models are presented and a control synthesis approach for stiff hybrid dynamic systems based on model checking is outlined.

3.15 Satellite Platform Case Study with SLIM and COMPASS

Viet Yen Nguyen (RWTH Aachen, DE)

This talk is a continuation of Thomas Noll's talk on SLIM, a formalized dialect of AADL. We report on the use of the COMPASS toolset on a satellite platform in development at the European Space Agency. These efforts were carried out in parallel with the conventional software development of the satellite. The nominal behavior of the satellite platform model, expressed in SLIM, comprises nearly 50 million states. This multiplies manifold upon the injection of failures. We show that verification and validation artifacts, typically constructed manually in a space system engineering process, can be automatically generated by the COMPASS toolset. The model's size pushed the computational tractability of the algorithms underlying the formal analyses, and revealed bottlenecks for future theoretical research. Additionally, the effort led to newly learned practices from which subsequent formal modeling and analysis efforts shall benefit, especially when they are injected in the conventional software development lifecycle. The case demonstrates the feasibility of fully capturing a system-level design as a single comprehensive formal model and analyze it automatically using a formal methods toolset based on (probabilistic) model checkers.

3.16 Correctness, Safety and Fault Tolerance in Aerospace Systems: The ESA COMPASS Project

Thomas Noll (RWTH Aachen, DE)

Building modern aerospace systems is highly demanding. They should be extremely dependable, offering service without failures for a very long time – typically years or decades. The need for an integrated system- software co-engineering framework to support the design of such systems is therefore pressing. However, current tools and formalisms tend to be tailored to specific analysis techniques and do not sufficiently cover the full spectrum of required system aspects such as safety, dependability and performability. Additionally, they cannot properly handle the intertwining of hardware and software operation. As such, current engineering practice lacks integration and coherence.

This talk gives an overview of the COMPASS project that was initiated by the European Space Agency to overcome this problem. It supports system- software co-engineering of real-time embedded systems by following a coherent and multidisciplinary approach. We show how such systems and their possible failures can be modeled in (a variant of) AADL, how their behavior can be formalized, and how to analyze them by means of model checking and related techniques. Practical experiences obtained in a larger case study will be described in a subsequent presentation by Viet Yen Nguyen.

3.17 Synchronous AADL: From Single-Rate to Multirate

Peter Csaba Ölveczky (University of Illinois – Urbana, US)

Distributed Real-Time Systems (DRTSs), such as avionics systems and distributed control systems in motor vehicles, are very hard to design because of asynchronous communication, network delays, and clock skews. Furthermore, their model checking typically becomes unfeasible in practice due to the large state spaces caused by the interleavings. Based on the observation that many automotive and avionics systems should be *virtually synchronous*—that is, conceptually, there is a logical period during which all components perform a transition and send messages to each other—that must be realized in a distributed environment with network delays, skewed local clocks, etc., we have proposed the PALS transformation [8, 9]. The key idea of PALS ("physically asynchronous logically synchronous") is that one can model and verify the much simpler synchronous design, and PALS then provides a correct-by-construction distributed asynchronous model.

To make the PALS modeling and verification methodology available to the modeler, we have defined an annotated sublanguage of AADL, called *Synchronous AADL*, that can be used to specify synchronous PALS designs in AADL [2]. We have defined the formal semantics of Synchronous AADL in Real-Time Maude, and have used this semantics to develop an OSATE plug-in, called *SynchAADL2Maude*, that provides simulation and temporal logic model checking for synchronous designs modeled in Synchronous AADL *within* OSATE [3]. This enables a model-engineering process for important classes of distributed real-time systems that combines the convenience of AADL modeling, the complexity reduction of PALS, and formal verification in Real-Time Maude. We have used SynchAADL2Maude on a virtually synchronous avionics system whose distributed asynchronous version (even in very simple settings) has millions of reachable states and cannot be feasibly model checked, but where the Synchronous AADL model of the corresponding synchronous PALS design could be verified by the SynchAADL2Maude tool in less than a second.

However, a number of DRTSs are *multirate* systems whose components have different periods. For example, the controller for the ailerons on an airplane may operate with a period of 15 ms, whereas the rudder controller operates with period 20 ms. These different components need to synchronize when turning the airplane. We have therefore extended PALS to multirate virtually synchronous systems, and are working on extending the SynchAADL2Maude tool to specify and verify such systems. That work could be based on the support for modeling multirate systems in AADL that has recently been developed by colleagues at UIUC and Rockwell Collins [1].

This presentation is based on joint work with José Meseguer, Kyungmin Bae, Lui Sha, Abdullah Al-Nayeem, Steven P. Miller, and Darren D. Cofer.

3.18 Semantic anchoring of industrial architectural description languages

Paul Pettersson (Mälardalen University – Västerås, SE

In some recent work, we have focused on defining semantics to industrial architecture description languages, such as Procom, AADL and EAST-ADL. Semantic anchoring of such languages has served different purposes. A main goal has been to enable analysis of the languages in analysis tools such as simulators and model-checkers, including UPPAAL and UPPAAL PORT. Current work is focusing on addressing dynamically reconfiguring systems and model-based testing using ADLs.

3.19 Integration of AADL models into the TTEthernet toolchain; Towards a model-driven analysis of TTEthernet networks

Ramon Serna Oliver (TTTech Computertechnik – Wien, AT)

As networked cyber-physical embedded systems become more and more populated, models to enable semantic analysis are a key factor to reduce complexity. We introduce Time-Triggered Ethernet (TTEthernet) and the TTE Tool-chain and explore the advantages of a complete system representation by means of the Architecture Analysis and Design Language (AADL).

The presentation elaborates on the use of AADL at different levels. Namely: providing a manageable representation of a (potentially complex) network; introducing a comprehensible interface to and within TTE-Tools; building a repository of system components, properties, constraints, and requirements; a means to network analysis and property verification; and, an open door to complex analysis (through existing tools, annexes, etc...).

3.20 Architecture Modeling and Analysis for Automotive Control System Development

Shin'ichi Shiraishi (TOYOTA InfoTechnology Center USA Inc., US)

Architecture modeling languages, e.g., AADL, SysML, and MARTE are well known languages and used among several different domains. This talk explains modeling steps based on these languages through a real-world automotive system example. On the other hand, this talk also explains our experience of architecture analysis from the real-time automotive system viewpoint.

In the end, the relation and gap between the architecture model and architecture analysis are discussed.

3.21 About architecture description languages and scheduling analysis

Frank Singhoff (University of Brest, FR)

The talk deals with performance verifications of architecture models. We focus on real-time embedded systems and their verification with the real-time scheduling theory.

Many industrial projects do not perform performance analysis with this theory even if the demand for the use of it is large. To perform verifications with the real-time scheduling theory, the architecture designers must check that their models are compliant with the assumptions of this theory. Unfortunately, this task is difficult since it requires that designers have a deep understanding of the real-time scheduling theory. In this presentation, we show how to help designers to check that an architecture model is compliant with the real-time scheduling theory assumptions.

We focus on schedulability tests. We show how to explicitly model the relationships between an AADL architectural model and schedulability tests. From these models, we apply a model- based engineering process to generate tools which are able to check compliance of architecture models with schedulability tests assumptions.

3.22 Co-modeling, simulation and validation of embedded software architectures using Polychrony

Jean-Pierre Talpin (INRIA – Rennes, FR)

The design of embedded systems from multiple views and heterogeneous models is ubiquitous in avionics as, in particular, different high-level modeling standards are adopted for specifying the structure, hardware and software components of a system. The system-level simulation of such composite models is necessary but difficult task, allowing to validate global design choices as early as possible in the system design flow. This paper presents an approach to the issue of composing, integrating and simulating heterogeneous models in a system co-design flow. First, the functional behavior of an application is modeled with synchronous data-flow and statechart diagrams using Simulink/Gene-Auto. The system architecture is modeled in the AADL standard. These high-level, synchronous and asynchronous, models are then translated into a common model, based on a polychronous model of computation, allowing for a Globally Asynchronous Locally Synchronous (GALS) interpretation of the composed models. This compositional translation is implemented as an automatic model transformation within Polychrony, a toolkit for embedded systems design supporting simulation, verification, controller synthesis, sequential and distributed code-generation. An avionic case study, consisting of a simplified doors and slides control system, is presented to illustrate our approach.

3.23 Compositional Analysis of Architecture models

Michael W. Whalen (University of Minnesota, US)

This presentation describes work towards a design flow and supporting tools to improve design and verification of complex cyber-physical systems. We focus on system architecture models composed from libraries of components and complexity-reducing design patterns having formally verified properties. This allows new system designs to be developed rapidly using patterns that have been shown to reduce unnecessary complexity and coupling between components. Components and patterns are annotated with formal contracts describing their guaranteed behaviors and the contextual assumptions that must be satisfied for their correct operation. We describe the compositional reasoning framework that we have developed for proving the correctness of a system design, and provide a proof of the soundness of our compositional reasoning approach. An example based on an aircraft flight control system is provided to illustrate the method and supporting analysis tools.

4 Working Groups

During the seminar, the group discussed possible topics for break-out sessions, and form different working groups. We placed two afternoon sessions dedicated to brainstorming session on Tuesday and Thursday afternoon.

The topics were the following, on Tuesday afternoon:

- Attaching semantics to a modeling framework (section 4.1)
- How much expressive power is needed for architectural models (section 4.2)
- Analysis of architectural systems (section 4.3)

on Thursday afternoon:

- Multi-point view analysis and combination of analysis result (section 4.4)
- Run-time Architectural Analysis (section 4.5)
- Notion of Time: Physical vs. Real-Time vs Discrete vs Logical (section 4.6)
- Patterns for (de)composing and analysing systems (section 4.7).

4.1 Attaching semantics to a modeling framework

Motivation. The group acknowledged the fact that semantics is a key enabler to perform further analysis. Actually, most analysis require hypothesis on the behavior of the system, the interconnection between elements to follow some particular semantics: typing system, execution semantics, propagation of information/events.

Also, the group recognized that this area is usually not well developed in many tools: semantics is usually part of the analysis tool itself, except for some consistency checking performed during model analysis and transformation. Actually, the interpretation of semantics as done by a tool is usually not explicit and remains hidden.

The group contemplated different options:

- **at tool-level**: this creates a strong link to a particular tool, and can be perceived as a vendor lock-in strategy.
- **at model-level**: the model can not only store user artifacts, but also the underlying semantics. This could enable a wider sharing and understanding of how to interpret a model.

  Furthermore, analysis tools can use this additional information. Yet this poses the question of the mechanisms to encode this semantics.
- **as part of the transformation process**: yet, this still creates a strong link between the tools, mostly the editing and processing parts, e.g. ECLIPSE.

Open questions and discussions. From the different options raised, the key question is *"What is expected from the tool?"*. There might be several expectations from the users (design teams, project management, tool builder, ...).

Building consistent semantics is a complex work, usually performed using operational semantics or equivalent frameworks. Attaching semantics, even if it is a desirable effort may create an artifact that is, in the end, too-complex to be manipulated. A corollary to the previous question is *"Is formal too formal?"*.

A good compromise seems to opt for the following strategy: 1) define a DSL to define the model, with an implicit semantics easy to understand thanks to well-chosen concepts, 2) define separately the formal semantics, in a readable format but only for people interested in the core details. Such strategy would enable meeting certification requirements. Also, defining separately the semantics would enable an adaptation on a per tool basis: scheduling and fault analysis could be defined based on two separate yet compatible description of the semantics of the system.

4.2 Expressive Power of Architectural Models

Motivation. A well-known trade-off in modeling is that, while increasing expressive power of a modeling language makes construction of models easier, it also makes analysis of models more difficult. Therefore, for an architectural modeling and analysis framework, it is important to identify the right level of expressiveness. Architectural analysis is intended to cover large-scale systems and systems of systems. This seems to imply that expressive power of architectural models and the level of detail in models cannot be too high. At the same time, an architectural modeling framework should not be viewed as a single modeling language, but rather as a collection of complementary languages that concentrate on different aspects of the system design, so that a model of a particular aspect can be simple and does not have to require much expressive power.

Generative techniques in architectural analysis. The key to an architecture-centric modeling framework is its reliance of generative techniques and model transformations. At the core of the framework, an architectural model serves as a repository of knowledge about the system. To perform analysis of a certain aspect of the system, an analysis model is generated from the architectural model that contains only the details needed for the selected analysis, and the expressive power of the language for the generated model is similarly targeted to, and limited by, the needs of the analysis. On the other hand, expressive power of the architectural description affects complexity of generation algorithms.

Generation of analysis models is enabled by the semantic core of the architectural framework that helps ensure that generated analysis models are consistent with each other and provides the basis for proving correctness of the generation. The semantic core is built on a semantic domain (e.g., sets of event sequences). Elements of the language establish relationships between elements of the domain. The logic for expressing these relationships can be general and expressive. We can use domain-specific languages to limit expressive power as needed.

An important aspect of a framework such as AADL is the ability to extend models through custom properties and annexes. Such an extension mechanism allows us to use different formalisms for different aspects of the system. The challenge is to use the extension mechanism is a way that keeps extensions compatible. Semantics of the extension mechanism itself become important here. In particular, when the extension is done by means of a new property set, a suitable semantics for properties may be by means of equations or constraints that relate values of properties in the set to each other as well as to concepts from the semantic core.

Architectural vs. behavioral modeling. Many architecture-level analysis techniques are concerns with operational aspects of the system. For example, timing and schedulability analysis relies on high-level thread execution semantics. Other analysis techniques can refine these high-level semantics with additional details of application behavior. There is much discussion in the community about the distinction between behavioral and architectural modeling and analysis, and whether the addition of behavioral details to an architectural model turns it into a behavioral model, defeating the promise of architecture-driven, system-level analysis. If this is indeed a valid concern, a question to be addressed by the research community is to decide whether the modeling language should enforce the acceptable level of behavioral detail.

It has been pointed out that the distinction between behavior and architecture may lie in the kind of analysis that is applied and not in the formalism *per se*. For example, a system of differential equations can be simulated, which involves computing the flows in the system. This is a clear example of behavioral analysis. On the other hand, the same system of equations can be used for structural analysis, which is, essentially, an architecture-level technique.

4.3 Analysis of architectural systems

Motivation. The breakout session started with a presentation of an example by Gabor Karsai of a multi-mission versatile constellation of satellites: the Fractionated Space System Architecture. Each satellite carries different instruments. The combination of satellites allows for complex missions. There are multi-scale architectural issues in this system: satellite-level, constellation-level to dimension network, scheduling, fault management logic or energy. But also mission-level challenges, e.g. to select the best configuration, but also the satellites most suitable for a particular mission.

The group noted that the architecture of the system is central to the analysis. Given the complexity of the whole constellation, one needs to perform careful and clever separation of concerns along the various dimensions. One needs to distinguish:

- Architectural level at which a property can be assessed
- Property determination as part of lifecycle: permanent, instance-specific
- V&V techniques: model checking, test, proof, validation, runtime monitoring, etc
- Time available for V&V effort to be bounded

Open questions and discussions. From these considerations, we note there are several issues related to the analysis strategy to be deployed:

We first note that the analysis strategy, as part of the design or the V&V qualification effort, is costly. A first element of choice is therefore the running-time assumptions one may attach to particular activities (e.g. model checking vs. static analysis of a pattern) Composability of analysis across architectural layers could also reduce this effort.

A first solution would be to properly document analysis in the form of contract passed to architecture elements. Pre-conditions are expected patterns, properties, post-conditions are new elements deduced and propagated to the architecture. Although the group agrees on general list of V&V techniques and objectives, we acknowledged such document is lacking to the community.

Another issues reported by the group is a "chicken/egg" problem among analysis. Some analysis are cross-dependent, but more important is that the output of these analysis are of equal importance to the designer, for instance parameter configuration of the system. Therefore, one needs to apply some particular optimization or SAT problem solving strategies at the architectural level.

We note these two open questions pose a strong constraint on tooling support. We note that analysis tools are usually viewed as "extensions" plug-in to the modeling environment. Actually, we may need to view it the other way: the analysis framework is central, and considers the actual architecture descriptions as "plug-in" from which it extracts relevant information and support the designer activities.

4.4 Multi-point view analysis and combination of analysis result

Motivation. Analyzing a complex embedded systems involve combining a model of the system under consideration (architecture, functional and non-functional properties) and analysis tools. We note several issues in this domain.

First, "properties" is a fuzzy term used to define either

1. what to describe: the way the system is from a set of simple classifiers, e.g. a task priority;
2. or what to assess: more complex elements deduced from the interacting componenents, answering an elaborate question "is the system schedulable? safe, etc"?

Besides, properties are either defined by the designer, or output of a particular analysis. A typical example being priorities applied to thread that can be either enforced or deduced from other elements. If we continue in the field of scheduling, it could be Rate Monotonic Analysis for deciding schedulability of a system based on actual priority value, or computing a priority assignment from Deadline Monotonic Analysis, etc.

Finally, we note, from the variety of Architecture Description Languages presented (MARTE, EAST-ADL, AADL, but also Simulink) that common properties may differ in names (WCET, Compute_Execution_Time), semantics (fixed value, range, ...), but also mae assumption on units ("ticks" vs. actual time units). Finally, they can be used in different contexts, to represent an assumption, a budget, a computed, measured or refined value depending on the process and the maturity of the model.

Open questions and discussions. We note a strong interaction between models and analysis. Actually, the "final" system after the whole design effort is actually an iterative fix-point where properties and model are no longer evolving. This raises some questions on how to combine analysis in a way that eases convergence to this fix point.

The group note this is highly related to the modeling process in place, and that some elements of solutions already exist. For instance, EAST-ADL has been defined so that project manager knows elements to be modeled for each steps of its design, how to relate those steps to analysis, and how to build new properties at step N+1 from analysis performed at step N. Such process is highly specific to the automotive domain covered by EAST-ADL that constrains the system dimension, it is not available for generic ADL like MARTE or AADL.

Another open question is about the role of analytical model in the whole design process. In MARTE or AADL, this analytical model is a by-product, that is not kept. Yet, some analysis are time consuming (e.g. simulations, model checking), knowing whether they should be redone is a critical part to help converging to the final solution. An option is to determine when a model has evolved in a way that impact the analytical model.

The group discussed possible options to tackle this issue: one may consider an analytical model to be: an actual result (yes/no, numerical values, etc.) and a contract set on the model, defining invariants to be preserved so that the result is preserved, e.g. architectural patterns, properties to be maintained over time. If one of the invariant is broken, the analytical model should be rebuilt. Such concept, although appealing to the mind needs to be further refined and applied to the underlying meta-modeling framework, e.g. EMF.

The group concluded that it would be a good topic for further collaborations.

4.5 Run-time Architectural Analysis

Motivation. Discussions at the working group were concerned with using architectural knowledge in dynamic analysis that is performed during the execution of the system. The need for run-time analysis is motivated by the two observations:

- On the one hand, the system is not always built according to the model. This is inevitable, since existing generative technologies cannot produce all aspects of the system implementation automatically. Manual implementation inevitably opens the possibility that developers deviate from the model, either by misinterpretation or by overzealous optimization.
- On the other hand, model-based process always realized on the assumptions made during modeling. These assumptions need to be validated at run time. A typical example of such an assumption is failure rates built into the error model of the system [7].

Design of run-time monitors. Based on the discussion above, runtime monitoring has two distinct functions:

- Enforce guarantees that have been offered by static architectural analysis done at run time. This includes validation of assumptions that were used to build the model, as well as validation of model parameters.
- Provide error detection and invocation of recovery operations.

Run-time analysis is based on monitoring of the system execution, which means that the system, in addition to the components introduced into the architectural model by the system designer, also implicitly contains components that implement the monitor. The choice of monitoring architecture, to some extent, is determined by the architecture of the system itself.

An important question is how to isolate the observer from the system and minimize, or at least account for, the monitoring overhead. The two common ways of implementing monitors are 1) build the monitor into the control path of the system or 2) run it in parallel with the system. For the first solution, monitoring can be built into the budget for each component during design. Concurrent observers are more powerful, as it is easier for the to maintain a global view of the execution of multiple components. However, their overhead is harder to quantify. Moreover, deployment of concurrent observers presents additional challenges, since an observer has to be synchronized to stable observation points of each component it monitors.

Open questions and discussions. In addition to the problem of monitoring of information related to the architectural model, a reverse question can be asked. Many systems already employ run-time observers; for example, for health management [6]. How can we use the available architectural information to improve efficiency of monitoring and reduce overhead. For example, system architecture can be utilized in deciding monitor placement.

The question of quantifying additional robustness is achieved in the system design through run-time monitoring remains an important question to be answered.

4.6 Notion of Time: Physical vs. Real-Time vs Discrete vs Logical time

Motivation. The notion of time is central not only in physics, but also in computer-centric systems, where several dimensions are to be combined:

- Semantics: discrete/dense time
- Uniformity: uniform/non-uniform time
- Linearity: linear/non-linear

- Representation: Timed Automata, Petri Nets, Tagged Systems, clocks
- Solving techniques
- Requirements and time

Actually, these dimensions reflect several use cases of time concepts for 1) modeling a system and its semantics then 2) to analyze it.

Hence, *discrete time* can be use to model *logical* time relationships between events (e.g. Lamport clocks, synchronous systems), *discrete events* systems or time scales for simulation. *Dense time* is relevant for physical system (ideal time for Newtonian systems) and is well represented by Timed automata.

Representing a system using a particular class of time (or clocks) is the first aspect. The challenging part is to define consistent solving techniques to assess time-based properties. A timed system is possibly infinite, one needs to map them to an equivalent problem that is finite. Several techniques have been defined: K-induction with monotonic real input, symbolic approaches (region graphs, etc.), explicit model checking (dealing with instants); representation of time using rational or floating-point abstractions.

Open questions and discussions. At first, the group questioned the necessity for multiple representations of time. Considering the family of embedded systems, we note that heterogeneous time representations are required to capture the time as seen by the hardware elements of the system, connected to the physical environment; but also logical time for pure software system as a logical abstraction of its behavior. Hence, we need to combine safely these representations, and "meet-in-the-middle" either in a top-down or bottom-up way.

Then, another question is how to map requirements onto this time system. Requirements to be fulfilled by the system are usually expressed in natural language, such as "The pacemaker shall pace the heart at least once per second". This requires variants of temporal logic (e.g. Timed-CTL) that are usually hard to master.

From these considerations, the group defined the following set of open questions:

- Which analyses are compatible ?
- How to combine different analysis techniques? In a practical way, but not in a purely engineering way !
- What is the role of architecture ? What can be reused/combined?

4.7 Patterns for (de)composing and analysing systems

Motivation. Patterns for composing and decomposing systems is an essential element for the engineering of complex system. The group reviewed typical strategies to address complexity in systems:

- "Divide and conquer", where a problem is separated into subproblems being resolved separately. The global problem being solved when all subproblems have a solution;
- "Separation of concerns", where concerns (e.g. safety and security) are addressed separately, with adapted techniques.

Open questions and discussions. One interesting question answered by the group is the relationship between well-known engineering practice and architectures. Architecture is about defining relationships between elements.

We note there are actually many elements to consolidate through an architecture.

"Architecture of structure" focuses on the system decomposition into subsystems, their interfaces, connections, etc. It is also concerned by extension points for later system evolution.

“Architecture of relations” focuses on the comprehension of a system intrinsic nature: functional, safety, reliability, physical, cost, etc.

Both architectures follow a similar “divide and conquer” pattern, the main distinction being that interfaces are more natural to separate than concern. Security can be seen as a particular view of the global system, while divide and conquer applies recursively to (sub)systems elements in an orderly way.

Hence, the challenge is to know how a particular concern can be mapped onto the architecture. Keeping the example of security, it is a global concept that emerges from the combination of all system blocks. The architecture details how basic elements cooperate to provide security.

The group made a convincing point that in initial stages of specification, the combination of concerns/components is usually explicit, but that it tends to get lost in later stages. In some sense, the engineering process evolves from a “correct by construction” paradigm to a “construct by correction” one.

We concluded that system architecture but also architecture modeling framework should guide the designer back on track to follow only a “correct by construction” path.

5 Summary and Open Challenges

The working group and the concluding session were the occasion of a lively discussion about open problems and roadmap for our community. The diversity of speakers, themes discussed and existing contributions showed that architecture is a central artifact in many processes, tools and methodology.

We summarize some of the discussions we had, and open challenges outlined by the group:

- **Building an architecture** design activities have a goal that may differ in nature: either test an hypothesis, build a full system ready to be deployed. In between, V&V activities must be conducted. Design patterns helped structuring software activities, but fails short to guide architectural designs.

  Hot topic the building of architecture requires higher-level patterns to know how to combine elements. At first, combining interfaces/components, but also concerns (e.g. safety, security, …). The former is well mastered by the industry, the latter is more problematic. Yet, we note that it is the failure to combine concerns that delay most projects, in particular in the embedded domain.
- **Semantics of an architecture** an architecture interconnects element, and give meaning to this assembly. Defining its semantics is usually done in an informal way (natural language), or through the combination of small elements of semantics (timed automata, syntactic elements, etc).

  Hot topic we see the emergence of numerous “Model of Computation” (Ravenscar, synchronous, etc), but also semantics for key aspects like time, fault propagation, … They precisely describe one semantics, but separately. One needs a global view on how to combine them to give a precise meaning to the whole architecture.
- **Architecture analysis as a MDE topic** Architecture Description Language are bound to model-based technology. Hence, many projects around ADL focus on model transformation techniques, mapping one architecture onto numerous analysis tools.

  Hot topic the preservation of semantics between the model of a system, and its analytical model (in a scheduling analysis framework, model checker, etc.) is mandatory to demonstrate that the results is meaningful. Current technologies fail to demonstrate this mapping in a general case.

- **Analysis of systems** several talks and discussions focused on particular analysis techniques, either to address particular properties (time, fault), or technical limitations (combinatorial explosions). We also noted that analysis is usually performed by people who are experts in a particular problem space (avionics, medical) but not in a given design space (timed automata, Petri nets, . . .).
  Hot topic thanks to MDE, one can map architecture to particular analysis, and take advantage of advanced techniques. One needs to complete this mapping with "wizards" to determine when a model is "ready" for analysis, and how to correct it to reach this state. Another hot topic is to determine how to send back meaningful diagnosis to the architectural designer in case an analysis fails.
- **Coupling architecture/analysis** we notes that analysis rely on architecture artifacts, but could also enrich architectures. This has a complex impact that must be evaluated. As it could definitely helps building better architectures faster.
  Hot topic an analysis could be defined as a contract set on an architecture. There are several "concerns" associated to analysis techniques and tools. An important topic is to build a cartography of these analysis, and their requirement put on architecture. Such map would ease the definition of architecture that is "correct by construction".

These different topics are already studied by the different participants. A consensus emerged that all these topics must be addressed in uniform way: either vertically, following one concern and one family of analysis; or horizontally, by combining concerns.

Solution to these different problems will address actual shortcomings in many domains: system engineering, software engineering and model-based techniques that are required to address the complexity of embedded systems.

6 Bibliography

1 A. Al-Nayeem, L. Sha, D. D. Cofer, and S. M. Miller. Pattern-based composition and analysis of virtually synchronized real-time distributed systems. In *Proc. Cyber-Physical Systems (IEEE/ACM ICCPS'12)*, 2012.

2 K. Bae, P. C. Ölveczky, A. Al-Nayeem, and J. Meseguer. Synchronous AADL and its formal analysis in Real-Time Maude. In *Proc. ICFEM'11*, volume 6991 of *LNCS*. Springer, 2011.

3 K. Bae, P. C. Ölveczky, J. Meseguer, and A. Al-Nayeem. The SynchAADL2Maude tool. In *Proc. FASE'12*, volume 7212 of *LNCS*. Springer, 2012.

4 Y. Ben Maïssa, F. Kordon, S. Mouline, and Y. Thierry-Mieg. Modeling and Analyzing Wireless Sensor Networks with VeriSensor. In *Petri Net and Software Engineering (PNSE 2012)*, volume 851, pages 60–76, Hamburg, Germany, June 2012. CEUR.

5 J-M. Couvreur and Y. Thierry-Mieg. Hierarchical Decision Diagrams to Exploit Model Structure. In *Formal Techniques for Networked and Distributed Systems - FORTE*, volume 3731 of *LNCS*, pages 443–457. Springer, 2005.

6 N. Mahadevan, A. Dubey, and G. Karsai. Architecting health management into software component assemblies: Lessons learned from the arinc-653 component model. In *The 15th IEEE International Symposium on Object/component/service-oriented Real-time distributed computing*, April 2012.

7 I. Meedeniya, I. Moser, A. Aleti, and L. Grunske. Architecture-based reliability evaluation under uncertainty. In *Proceedings of the 7th International Conference on the Quality of Software Architectures (QoSA 2011)*, pages 85–94, June 2011.

8 J. Meseguer and P. C. Ölveczky. Formalization and correctness of the PALS architectural pattern for distributed real-time systems. *Theoretical Computer Science*, 2012. Article in press, http://dx.doi.org/10.1016/j.tcs.2012.05.040.

9 S. P. Miller, D. D. Cofer, L. Sha, J. Meseguer, and A. Al-Nayeem. Implementing logical synchrony in integrated modular avionics. In *Proc. DASC'09*. IEEE, 2009.

10 Y. Thierry-Mieg, B. Bérard, F. Kordon, D. Lime, and O. H. Roux. Compositional Analysis of Discrete Time Petri nets. In *1st workshop on Petri Nets Compositions (CompoNet 2011)*, volume 726, pages 17–31, Newcastle, UK, June 2011. CEUR.

11 Y. Thierry-Mieg, D. Poitrenaud, A. Hamez, and F. Kordon. Hierarchical set decision diagrams and regular models. In *Tools and Algorithms for the Construction and Analysis of Systems – TACAS*, volume 5505 of *LNCS*, pages 1–15. Springer, 2009.

Participants

- De-Jiu Chen
  KTH – Stockholm, SE
- Silvano Dal Zilio
  LAAS – Toulouse, FR
- Patricia Derler
  University of California – Berkeley, US
- Mamoun Filali-Amine
  Paul Sabatier University – Toulouse, FR
- Laurent Fournier
  Rockwell Collins France, FR
- Serban Gheorghe
  Edgewater Computer Systems Inc. – Ottawa, CA
- Lars Grunske
  TU Kaiserslautern, DE
- Jérôme Hugues
  ISAE – Toulouse, FR
- Naoki Ishihama
  JAXA – Ibaraki, JP
- Gabor Karsai
  Vanderbilt University, US
- Alexey Khoroshilov
  Russian Academy of Sciences – Moscow, RU
- Fabrice Kordon
  UPMC – Paris, FR
- Brian Larson
  Multitude Corp., US
- Bruce Lewis
  US Army AMRDEC, US
- Ivano Malavolta
  Univ. degli Studi di L'Aquila, IT
- Frédéric Mallet
  INRIA Sophia Antipolis, FR
- Pieter J. Mosterman
  The MathWorks Inc. – Natick, US
- Viet Yen Nguyen
  RWTH Aachen, DE
- Thomas Noll
  RWTH Aachen, DE
- Peter Csaba Ölveczky
  Univ. of Illinois – Urbana, US
- Paul Pettersson
  Mälardalen University – Västerås, SE
- Ramon Serna Oliver
  TTTech Computertechnik – Wien, AT
- Shin'ichi Shiraishi
  TOYOTA InfoTechnology Center USA Inc., US
- Frank Singhoff
  University of Brest, FR
- Oleg Sokolsky
  University of Pennsylvania, US
- Jean-Pierre Talpin
  INRIA – Rennes, FR
- Michael W. Whalen
  University of Minnesota, US

Report from Dagstuhl Seminar 12281

Security and Dependability for Federated Cloud Platforms

Edited by

Rüdiger Kapitza[1], Matthias Schunter[2], Marc Shapiro[3], Paulo Verissimo[4], and Michael Waidner[5]

1 **TU Braunschweig, DE,** kapitza@ibr.cs.tu-bs.de
2 **INTEL CRI-SC, Darmstadt, DE,** mts@schunter.org
3 **INRIA & LIP6, Paris, FR,** Marc.Shapiro@acm.org
4 **University of Lisboa, PT,** pjv@di.fc.ul.pt
5 **Fraunhofer SIT - Darmstadt, DE,** waidner@sit.fraunhofer.de

Abstract

From July 8-13, 2012, the Dagstuhl Seminar "Security and Dependability for Federated Cloud Platforms" was held in Schloss Dagstuhl – Leibniz Center for Informatics. During this seminar, participants presented their current research and discussed open problems in the fields of security and dependability of infrastructure clouds and their federation. The executive summary and abstracts of the talks given during the seminar are put together in this paper.

Seminar 8.–13. July, 2012 – http://www.dagstuhl.de/12281
1998 ACM Subject Classification K.6.5 Security and Protection
Keywords and phrases Security, Cloud computing, Virtual systems, Dependability, Grid computing, Systems management
Digital Object Identifier 10.4230/DagRep.2.7.56

1 Executive Summary

Rüdiger Kapitza
Matthias Schunter
Marc Shapiro
Paulo Verissimo
Michael Waidner

Computing services are increasingly pooled within global utility computing infrastructures offered by providers such as Amazon, Google and IBM. Infrastructure clouds provide virtual machines and resources. These infrastructure clouds are used to enable "platforms as a service" that simplify implementation of arbitrary scalable services.

The seminar targeted the management and protection of individual clouds and addressed the trend towards cloud federation by bringing together researchers from systems management, security, and dependability. The idea was that only such an integrated approach is able to guarantee security and dependability while preserving the essential cost and efficiency benefits of today's emerging solutions.

The challenge to address was how to provide secure and dependable services on such federated cloud platforms. Selected research questions were: How can clouds securely interoperate, how can service availability be guaranteed despite failures or attacks by individual clouds, how can existing algorithms be adjusted to provide scalable eventual consistency, and finally whether cloud-of-cloud infrastructures can provide such benefits at costs that are competitive with single cloud solutions. While these questions where addressed during the seminar it got also clear that dependability and security of single clouds is by far not solved and therefore was also discussed in depth.

2 Table of Contents

Executive Summary

Overview of Talks

3 Overview of Talks

3.1 Capacity Planning for Clouds: Problems, Solutions, Consequences

Artur Andrzejak (Universität Heidelberg, DE)

Joint work of Andrzejak, Artur; Kondo, Derrick; Yi, Sangho; Anderson, David P.
Main reference Artur Andrzejak, Derrick Kondo, and Sangho Yi: Decision Model for Cloud Computing under SLA Constraints, 18th Annual Meeting of the IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2010), Miami Beach, Florida - August 17-19, 20

A major promise of cloud computing is that a customer can change resource capacity on-demand. Combined with multiplexing demands of (uncorrelated) customers, this seems to solve the capacity (planning) problem for cloud providers and their clients. We argue that despite of this obvious progress, there are still significant research challenges and as well as opportunities on both sides.

First we analyse the provider's strategies to handle demand spikes which are likely to be caused by correlated demand. Among different options, we take a closer look at the approach taken by Amazon's EC2, namely to provision for the worst case but sell unused capacity at a discount (via Spot Instances offered at a fluctuating market price). We study how this market approach offers to users opportunity to trade various QoS aspects (e.g. duration of a batch computation, number of interruptions) against the total cost. Our (quite surprising) conclusion is: it is not worth to bid low but it is significant to select the right instance type.

In the further part of the talk we discuss a technique for increasing user's computational capacity under cost constraints. It works by mixing resources of different availability levels (e.g. voluntarily resources, Spot Instances, and (highly available) dedicated resources according) in proportions which guarantee both a certain availability level and a cap on the total costs. An important aspect here is that we request the user to replace traditional notion of availability (of individual machines) by collective availability, which only guarantees that some fraction of resources within a set will be available in a time interval.

Such "tweaking" of QoS notions is instrumental also in the third topic of the talk: cost-efficient backup storage via private clouds. The key idea is to use dedicated and non-dedicated institutional machines together, and lower the storage costs by trading these costs against QoS metrics such as time until data restore completes. Also this approach illustrate that a changed understanding of common QoS notions can provide substantial savings for users.

References

1 Artur Andrzejak, Derrick Kondo, and Sangho Yi, "Decision Model for Cloud Computing under SLA Constraints," in *18th Annual Meeting of the IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2010)*, Miami Beach, Florida - August 17-19, 2010.

2 S. Yi, D. Kondo, and A. Andrzejak, "Reducing costs of spot instances via checkpointing in the amazon elastic compute cloud," in *The 3rd International Conference on Cloud Computing (CLOUD'10)*, July 2010, pp. 236–243.

3 A. Andrzejak, D. Kondo, and D. P. Anderson, "Exploiting non-dedicated resources for cloud computing," in *12th IEEE/IFIP Network Operations & Management Symposium (NOMS 2010)*, Osaka, Japan, Apr 19–23 2010.

4 D. Kondo, A. Andrzejak, and D. P. Anderson, "On correlated availability in internet distributed systems," in *IEEE/ACM International Conference on Grid Computing (Grid)*, Tsukuba, Japan, 2008.

3.2 Towards a Cloud-of-Clouds File System

Alysson Neves Bessani (University of Lisboa, PT)

Joint work of Bessani, Alysson; Mendes, Ricardo; Oliveira, Tiago; Pasin, Marcelo

Recently it was shown that the use of multiple cloud providers for building cloud storage services can amend many concerns related to the lack of trust in storing critical data in the cloud. However, these services are still difficult to use when compared with local storage systems, e.g., a file system. In this talk we'll present the design of C2FS, a multi-client cloud-of-clouds file system that provides near- POSIX semantics (including strong consistency) even if a subset of the providers are subject to arbitrary faults. The design of C2FS leads to the development of new abstractions that may be useful for other cloud-backed files systems, or even distributed file systems in general.

3.3 Old and new security issues in the Cloud

Herbert Bos (VU - Amsterdam, NL)

The question we address is: to what extent are clouds different from other forms of computing?

We can loosely describe the cloud as a scalable solution where the more you pay, the more you get. Cycles, storage, bandwidth, availability – anything. It is interesting to note that the same description fits some of the modern botnets. We have compute botnets, DDoS botnets, botnets that steal information, and so on. The specialisation, flexibility and support provided by modern malware just about qualifies them to be called Malware as a Service (MaaS).

Besides noticing the remarkable similarities between clouds and botnets, the real research question in this presentation was whether or not (public) clouds are fundamentally different from local computation – from a security point of view. In other words, which security issues are fundamental to cloud computing and which are more or less similar? The talk proposed several such issues to serve as inputs for debate.

Positives

1. Clouds allow us to scale expert administration. On the one hand, the need for computation and storage keeps growing and so does the complexity of the systems that provides these services. On the other, we see no such growth in the number of expert administrators. There is a risk that expert administrators will become unaffordable in the near future for many organisations, with serious security implications. The cloud allows expert administrators to handle a much larger group of customers who can enjoy proper and secure administration of their systems.
2. Consolidation makes it easier to detect new attacks. The presence of many systems from many clients in the same data center makes it possible to detect anomalous behaviour earlier and more accurately.
3. Software monocultures *may* allow us to spread expensive security checks over multiple installations (collaborative security) and to spread cures quickly. A similar point was argued by Michael Locasto et al. in NDSS 2006 for the monoculture provided by MS Windows.

Negatives:

1. You make yourself a bigger target. For attackers, it may be more interesting to attack a system that hosts many applications from many different customers.
2. You become vulnerable to information loss/theft by the cloud provider (or perhaps a disgruntled employee) – neither of whom are under your control. Homomorphic encryption is too expensive to be a real solution in general in the foreseeable future. You may spread your trust over multiple clouds, but then cost equations change.
3. Software monocultures can also be a threat (and is frequently perceived as such). Exploits and malware spread faster.
4. You rely on a payment scheme to pay for the cloud resources. The payment scheme itself may be targeted by attackers. An example is the Zeus attack on the Canadian company Ceridian (where the attackers stole the credentials for the use of a cloud-based payroll system and created fake employees to have them paid).
5. Black hats may use legitimate clouds for criminal purposes. Clouds are interesting for attackers only. They offer a lot of temporary compute power to crack passwords, or launch DoS attacks, etc. Thus, we see that cybercriminals move to the cloud also.

3.4 On Dependability of Hadoop MapReduce in Cloud Environments

Sara Bouchenak (INRIA Rhône-Alpes, FR)

Main reference Amit Sangroya, Damian Serrano, Sara Bouchenak. Benchmarking Dependability of MapReduce Systems. In Proceedings of the 31st IEEE International Symposium on Reliable Distributed Systems (SRDS), Irvine, CA, October 2012.
URL http://sardes.inrialpes.fr/research/mrbs/

MapReduce is a popular programming model for distributed data processing. Extensive research has been conducted on the reliability of MapReduce, ranging from adaptive and on-demand fault-tolerance to new fault-tolerance models. However, realistic benchmarks are still missing to analyze and compare the effectiveness of these proposals. To date, most MapReduce fault-tolerance solutions have been evaluated using microbenchmarks in an ad-hoc and overly simplified setting, which may not be representative of real-world applications. This talk presents MRBS, a comprehensive benchmark suite for evaluating the dependability of MapReduce systems. MRBS includes five benchmarks covering several application domains and a wide range of execution scenarios such as data-intensive vs. compute-intensive applications, or batch applications vs. online interactive applications. MRBS allows to inject various types of faults at different rates. It also considers different application workloads and dataloads, and produces extensive reliability, availability and performance statistics. We illustrate the use of MRBS with Hadoop clusters running on Amazon EC2, and on a private cloud.

3.5 Decentralized Software Verification for Trusted Computing in Federated Clouds

Gregory Chockler (IBM - Haifa, IL)

One of the chief benefits of federated cloud environments is their potential to offer their users a rich ecosystem of diverse software services hosted within the confines of the individual federation members. To enable the interested parties to safely use these services in their applications, they should be able to gain assurance that the behavior of the services complies with their advertised specifications. We propose a new approach allowing the users to gain assurance in the service integrity in a scalable fashion without relying on either a centralized certification authority or access to the actual implementation code. Our approach relies on the existing techniques for black-box testing that generate test suites covering all interesting behaviors described by the specification. In our framework, the specification is described as a collection of safety and liveness properties, each of which is expressed as a temporal logic formula or a finite automaton. Coverage of the specification is achieved by ensuring that generated test cases exercise all paths through the automaton. Each individual test case is then executed large number of times with varying input parameters values thus reducing the provider's ability to forge the execution results without having access to correctly behaving software. Finally, our method gains efficiency by applying the techniques of property testing, which allow the verifier to estimate whether an execution satisfies the property with a high probability by sampling a constant number of bits from it. The selections are independent, hence the confidence in the result can be increased by increasing the size of the sample.

3.6 Byzantine Fault-Tolerant MapReduce in Clouds-of-Clouds

Miguel Pupo Correia (IST - TU of Lisbon, PT)

Main reference Pedro Costa, Marcelo Pasin, Alysson N. Bessani, and Miguel Correia. Byzantine Fault-Tolerant MapReduce: Faults are Not Just Crashes. In Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CLOUDCOM '11), 2011.

MapReduce is a popular framework for processing large data sets in cloud environments. Both the original MapReduce implementation and the open source Hadoop can tolerate some classes of faults: worker crashes and file corruptions. However, there is evidence that other, more pernicious, failure modes can affect MapReduce job executions. First, accidental arbitrary faults due to hardware faults may corrupt the results of a job execution. Second, malicious arbitrary faults, due to malicious insiders or intrusions, can affect the correctness and liveness of a job execution. Finally, many cloud outages have been reported, so it may be desirable to run jobs in a federation of clouds to tolerate such faults.

The talk is about a MapReduce runtime that tolerates arbitrary faults and runs in a set of clouds (a 'cloud-of-clouds') at a reasonable cost in terms of computation and execution time. The replication of MapReduce to tolerate accidental arbitrary faults has already been explored in a previous work (presented at CloudCom'11). The novel challenges were to deal with malicious faults and to avoid sending through the internet the huge amount of data that would normally be exchanged between map and reduce tasks.

3.7 Resource and service sharing in clouds of mobile devices

Alexandra Dmitrienko (Fraunhofer Inst. - Darmstadt, DE)

Mobile phones are personal devices that provide useful services to their users, such as telephony, SMS messages and the Internet connection. While traditionally these services are consumed by the phone owners, we consider application scenarios where these resources are shared among multiple users within a cloud of mobile devices. However, the ability to share services and resources with other potentially untrusted entities exposes mobile devices to attacks. We discuss one possible solution to this problem based on the adaptation of the trusted virtual domains (TVDs) concept known in the PC world. TVDs allow for isolating private and public workloads in different domains. However, distributed and ad-hoc nature of a mobile cloud makes the application of a typical TVD design very challenging and requires significant refinements of the TVD architecture. Thus, we are investigating possible TVD designs for ad-hoc TVDs in order to provide secure service sharing among mobile devices.

3.8 Cost-efficient Robust Atomic Storage

Dan Dobre (NEC Laboratories Europe - Heidelberg, DE)

We address the problem of robustly sharing data among a possibly unbounded number of clients by leveraging a set of untrusted cloud servers. Robustness here means providing wait-free and atomic read/write access to shared data in the face of asynchrony, concurrency and the largest possible number of malicious failures by servers and clients. In this work, we present a protocol that features optimal worst-case read/write latency of two and three communication round-trips respectively. In addition, our protocol exhibits communication efficient reads, rendering our solution particularly suitable for sharing large data objects. As far as we are aware, this is the first result showing that the optimal read latency of two round-trips can be attained without relying on expensive public-key cryptography. Furthermore, that in the Byzantine context, rather than writing back full values when reading, it is sufficient to store pointers to values already held in storage. For these purposes, we are using a novel technique enabling readers to provably determine the progress of a (possibly concurrent) write operation by inspecting only a fraction of the servers accessed by that write. We give two alternative implementations of our technique, relying on one- way functions and secret sharing respectively. While the former is cost-efficient and has direct practical applicability, the latter shows that the same properties can be attained even in the presence of a computationally unbounded adversary.

3.9 Dart - a new programming language for structured web programming

Nicolas Geoffray (Google - Aarhus, DK)

Dart is a new programming language for creating structured web applications. It has an unsurprising and familiar syntax and it has been designed from ground up with performance and ease-of-use in mind. The presentation targeted the story behind Dart, and an introduction to the language, with an emphasis on its security features.

3.10 On the insecurity of Cloud configurations

Gabriela Gheorghe (University of Luxembourg, LU)

Joint work of Gheorghe, Gabriela; Crispo, Bruno; Carbone, Roberto; Desmet, Lieven; Joosens, Wouter
Main reference Gheorghe, Gabriela and Crispo, Bruno and Carbone, Roberto and Desmet, Lieven and Joosen, Wouter; Deploy, adjust and readjust: supporting dynamic reconfiguration of policy enforcement; Proceedings of the 12th ACM/IFIP/USENIX international conference on Middleware; 2011; 350–369
URL http://dx.doi.org/10.1007/978-3-642-25821-3_18

For large distributed applications based on Cloud technologies, security and performance are two requirements often difficult to satisfy together. In the first part of this talk, I discuss the limits of Cloud abstractions when it comes to security: verification, cross-layer trust concerns, absence of semantics coupled with security policy languages, the need for security at the middleware layer. In the second part of the talk, I discuss the security implications of mishandling authorisation information, by subjecting it to performance-enhancing techniques like caching. I argue for the necessity of managing security data separately from application data, and in a way that adapts to the fluctuating tradeoff between the security and performance requirements of the Cloud application. This approach leaves open questions such as: what are the different constraints on handling application data? what are the ways to configure or misconfigure a Cloud from the point of view of management of security data? how to be sure that a distributed system is handling security data as it should?

3.11 Building specialized BFT protocols using Abortable abstractions

Nikola Knezevic (IBM Research - Zürich, CH)

Joint work of Guerraoui, Rachid; Knezevic, Nikola; Quema, Vivien; Vukolic, Mark
Main reference Rachid Guerraoui, Nikola Knezevic, Vivien Quéma, and Marko Vukolic;. 2010. The next 700 BFT protocols. In Proceedings of the 5th European conference on Computer systems (EuroSys '10). ACM
URL http://doi.acm.org/10.1145/1755913.1755950

Modern Byzantine fault-tolerant state machine replication (BFT) protocols are notoriously difficult to develop, test and prove, as their implementations span over 20000 lines of involved

C++ code, related to synchronization, network and cryptography. Furthermore, one-size-fits-all does not easily apply to BFT protocols, which need to tolerate all possible, even malicious, situations well, making their implementations even more complex. In order to remedy this situation, we propose a new abstraction to simplify the development and the analysis of BFT protocols. We treat a BFT protocol as a composition of instances of our abstraction, where each instance can *abort* if working conditions differ from its specification. Each instance is developed and analyzed independently. To illustrate our approach, we show how abortable BFT could be used to develop a highly specialized protocol — namely, one that achieves the highest possible throughput when there is no faults (the most common situation). To cover worst-case situations, we developed another protocol, but our abstraction allows for using any other, existing BFT protocol. Typically, a good choice is a classical one like PBFT which has been proved correct and widely tested. Finally, our specialized protocol, named Ring, achieves up to 27other protocols, requiring only about 6000 lines of C++ code.

3.12 Performance Dependability for Complex Cloud Applications

Guillaume Pierre (VU - Amsterdam, NL)

Joint work of Pierre, Guillaume; Dejun, Jiang; Wei, Zhou

Online cloud applications often receive widely varying workloads which make it hard to guarantee performance and financial hosting costs. One solution for this problem is resource provisioning, which aims at maintaining the end-to-end response time of a web application within a pre-defined range (Service Level Objective, or SLO).

Resource provisioning is hard even if we assume that applications and resources are fully homogeneous. However, both of these hypotheses are usually untrue. This presentation discusses how one may handle these two challenges.

First, online applications are usually not homogeneous but are rather composed of multiple services calling each other to provide the desired service. When the SLO is violated, a difficult decision is to choose which service(s) should be re-provisioned for optimal effect. We propose to assign an SLO only to the front-end service. Other services are not given any particular response time objectives. Services are autonomously responsible for their own provisioning operations and collaboratively negotiate performance objectives with each other to decide the provisioning service(s).

Second, cloud resources are usually not homogeneous. Even when creating multiple instances with the the exact same VM type, the performance one gets out of these VMs is extremely heterogeneous. This has important implications for resource provisioning: one now needs to benchmark each VM instance individually, in order to determine how each VM can be put to use for the best effect.

3.13 Towards Secure Cloud Applications using Information Flow Control

Peter R. Pietzuch (Imperial College London, GB)

Joint work of Pietzuch, Peter R.; Ioannis Papagiannis; Matteo Migliavacca, David M. Eyers, Brian Shand, David Evans, Petr Hosek, Jean Bacon
Main reference Matteo Migliavacca, Ioannis Papagiannis, David M. Eyers, Brian Shand, Jean Bacon, and Peter Pietzuch, "DEFCon: High-Performance Event Processing with Information Security", USENIX Annual Technical Conference (USENIX ATC'10), Boston, MA, USA, USENIX, 06/2010.
URL http://www.smartflow.org

Ensuring the confidentiality and integrity of data in cloud-deployed healthcare or financial applications is challenging. Developers may introduce unintended or deliberate security flaws in different parts of an application, which may lead to the disclosure of sensitive data, or there may be vulnerabilities in the cloud platform itself. While access control mechanisms and source code auditing are used in practice to avoid security flaws, security violations are nevertheless a frequent occurrence.

Instead of attempting to avoid all security flaws, we propose to provide a "safety net" to cloud-deployed distributed applications that prevents sensitive data disclosure from happening. Our approach is to use information flow control (IFC) to track the flow of data through a complex, heterogeneous distributed application and constrain undesirable flows that could violate data protection policy. Our DEFCon middleware demonstrates how an IFC model can be applied to Java applications by adding support for strong isolation between objects to the Java runtime system.

3.14 Diagnostics and Forensics in Clouds of Clouds

Hans Peter Reiser (Universität Passau, DE)

Loss of control is a problem commonly attributed to cloud computing. Users no longer have direct physical control over the resources they use. This problem not only rises the question of trustworthiness of the cloud-based infrastructure, but also complicates diagnosis and forensics targeted at the user's applications. With growing application complexity, applications are likely to be confronted with more frequent failures. If a user's application no longer works as expected, failure and root-cause analysis is essential. Similarily, criminal forensics also requires approaches to acquire detailled information about a target user's applications in the cloud. In a cloud-of-cloud situation, in which applications are replicated and/or migrated over multiple, independent, competing clouds, a comprehensive analysis is even more challenging.

3.15 Partial Replication in Multi-Clouds

Luis Rodrigues (Technical University - Lisboa, PT)

Joint work of Romano, Paolo; Carvalho, Nuno; Couceiro, Maria; Didona, Diego; Fernandes, Joao; Palmieri, Roberto; Peluso, Sebastiano; Quaglia, Francesco; Ruivo, Pedro; Bessani, Alysson

The first part of my talk makes an overview of the research that has been performed by the e Distributed Systems Group at INESC-ID on the topic Distributed Software Transactional Memories (DTM). Over the last 4 years we have published a number of papers based on prototypes that we have built to better understand the advantages and limitations of the DTM paradigm. These papers have been briefly introduced and put in the context of main research direction explored by our group in this domain.

The second part of the talk focused on one of these results, that we consider that can be potentially extended to run on multi-cloud environments. In particular, I have described GMU, a multiversion update-serializable protocol for genuine partial data replication. To the best of our knowledge, the first genuine, and hence highly scalable, multi-versioning protocol supporting invisible reads and wait-free read-only transactions, hence achieving excellent performance in read-dominated workloads, as typical of a wide range of real-world applications. Interestingly, achieving this result required introducing a slight relaxation of classic One Copy Serializability (1CS): GMU in fact guarantees update- serializability, a consistency criterion weaker than 1CS but still compliant with the ansi serializable isolation level.

Finally, the third part of the talk discussed how GMU may be extended to support partial replication in multi-clouds.

This work was partially supported by the FCT (INESC-ID multi annual funding through the PIDDAC Program fund grant, under project PEst-OE/ EEI/ LA0021/ 2011, and by the project PTDC/EIA-EIA/102212/2008) and also by project CloudTM (FP7-ICT 257784).

References

1 A. Adya. *Weak Consistency: A Generalized Theory and Optimistic Implementations for Distributed Transactions.* PhD thesis, MIT, 1999.

2 N. Carvalho, P. Romano, and L. Rodrigues. Asynchronous lease-based replication of software transactional memory. In *Middleware*, pages 376–396, 2010.

3 N. Carvalho, P. Romano, and L. Rodrigues. A generic framework for replicated software transactional memories. In *NCA*, 2011.

4 N. Carvalho, P. Romano, and L. Rodrigues. Scert: Speculative certification in replicated software transactional memories. In *SYSTOR*, 2011.

5 M. Couceiro, P. Romano, N. Carvalho, and L. Rodrigues. D2STM: dependable distributed software transactional memory. In *PRDC*, 2009.

6 M. Couceiro, P. Romano, and L. Rodrigues. A machine learning approach to performance prediction of total order broadcast protocols. In *SASO*, 2010.

7 M. Couceiro, P. Romano, and L. Rodrigues. Polycert: Polymorphic self-optimizing replication for in-memory transactional grids. In *Middleware*, 2011.

8 R. Palmieri, P. Romano, and F. Quaglia. Aggro: Boosting stm replication via aggressively optimistic transaction processing. In *NCA*, 2010.

9 A. Peluso, P. Ruivo, P. Romano, Quaglia F., and L. Rodrigues. When scalability meets consistency: Genuine multiversion update-serializable partial data replication. In *ICDCS*, 2012.

10 S. Peluso, J. Fernandes, P. Romano, F. Quaglia, and L. Rodrigues. SPECULA: speculative replication of software transactional memory. In *SRDS*, October 2012.

11 P. Romano, R. Palmieri, F. Quaglia, N. Carvalho, and L. Rodrigues. An optimal speculative transactional replication protocol. In *ISPA*, Taiwan, Taipei, 2010.

12 P. Romano and L. Rodrigues. An efficient weak mutual exclusion algorithm. In *ISPDC*, 2009.

13 P. Romano, L. Rodrigues, and N. Carvalho. The weak mutual exclusion problem. In *IPDPS*, 2009.

14 P. Ruivo, M. Couceiro, P. Romano, and L. Rodrigues. Exploiting total order multicast in weakly consistent transactional caches. In *PRDC*, 2011.

3.16 Policy-Sealed Data: A New Abstraction for Building Trusted Cloud Services

Nuno Santos (MPI für Softwaresysteme - Saarbrücken, DE)

Joint work of Santos, Nuno; Rodrigues, Rodrigo; Gummadi, Krishna; Saroiu, Stefan
Main reference Policy-Sealed Data: A New Abstraction for Building Trusted Cloud Services; Nuno Santos, Rodrigo Rodrigues, Krishna P. Gummadi, and Stefan Saroiu; Proceedings of USENIX Security, Bellevue, WS, USA, August, 2012
URL https://www.usenix.org/conference/usenixsecurity12/tech-schedule/technical-sessions

Accidental or intentional mismanagement of cloud software by administrators poses a serious threat to the integrity and confidentiality of customer data hosted by cloud services. Trusted computing provides an important foundation for designing cloud services that are more resilient to these threats. However, current trusted computing technology is ill-suited to the cloud as it exposes too many internal details of the cloud infrastructure, hinders fault tolerance and load-balancing flexibility, and performs poorly. We present Excalibur, a system that addresses these limitations by enabling the design of trusted cloud services. Excalibur provides a new trusted computing abstraction, called policy-sealed data, that lets data be sealed (i.e., encrypted to a customer-defined policy) and then unsealed (i.e., decrypted) only by nodes whose configurations match the policy. To provide this abstraction, Excalibur uses attribute-based encryption, which reduces the overhead of key management and improves the performance of the distributed protocols employed. To demonstrate that Excalibur is practical, we incorporated it in the Eucalyptus open-source cloud platform. Policy-sealed data can provide greater confidence to Eucalyptus customers that their data is not being mismanaged.

3.17 Swiftcloud: deploying conflict-free objects at large scale

Marc Shapiro (INRIA & LIP6, Paris, FR)

Joint work of Shapiro, Marc; Preguiça, Nuno; Baquero, Carlos; Zawirski, Marek
Main reference Conflict-free Replicated Data Types. 13th Int. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS). Grenoble, France, 10-12 October 2011
URL http://dx.doi.org/10.1007/978-3-642-24550-3_29

Conflict-Free Replicated Data Types (CRDTs) support Strong Eventual Consistency (SEC). They can be replicated at extremely large scale, while remaining extremely responsive, available, and fault tolerant.

In the first part of this talk, we first define SEC and CRDTs. Then we give some simple sufficient conditions for conflict-freedom in both the state-based (aka data-shipping) and the operation-based (aka function-shipping). Then we show how to design a CRDT, focusing on the example of a set, to ensure that it has intuitive semantics and that it uses memory efficiently.

In the second part, we describe the Swiftcloud CRDT store and its support for conflict-free transactions. Swiftcloud aims to deploy shared CRDT objects at extreme scale, very close to clients at the network edge. To make it easier to program with CRDTs, a conflict-free transaction presents the application with a consistent snapshot of the database and ensures that its results are transmitted atomically. We have implemented a social-network application; experiments show several orders of magnitude performance improvement over a more classical synchronisation-based approach.

3.18 Dynamic Reconfiguration of Primary/Backup Clusters (with application to Apache ZooKeeper)

Alexander Shraer (Yahoo! Research - Santa Clara, US)

Joint work of Shraer, Alexander; Reed, Benjamin; Malkhi, Dahlia; Junqueira, Flavio
Main reference Alexander Shraer, Benjamin Reed, Dahlia Malkhi and Flavio Junqueira, USENIX Annual Technical Conference 2012 and Hadoop Summit 2012
URL http://www.cs.technion.ac.il/ shralex/zkreconfig.pdf

Dynamically changing (reconfiguring) the membership of a replicated distributed system while preserving data consistency and system availability is a challenging problem. In this talk I will discuss this problem in the context of Primary/Backup clusters and Apache Zookeeper. Zookeeper is an open source system which enables highly reliable distributed coordination. It is widely used in industry, for example in Yahoo!, Facebook,Twitter, VMWare, Box, Cloudera, Mapr, UBS, Goldman Sachs, Nicira, Netflix and many others. A common use-case of Zookeeper is to dynamically maintain membership and other configuration metadata for its users. Zookeeper itself is a replicated distributed system. Unfortunately, the membership and all other configuration parameters of Zookeeper are static - they're loaded during boot and cannot be altered. Operators resort to ”rolling restart” - a manually intensive and error-prone method of changing the configuration that has caused data loss and inconsistency in production. Automatic reconfiguration functionality has been requested by operators since 2008. Several previous proposals were found incorrect and rejected. We designed and implemented a new reconfiguration protocol in Zookeeper and are currently integrating it into the codebase. It fully automates configuration changes: the set of Zookeeper servers, their roles, addresses, etc. can be changed dynamically, without service interruption and while maintaining data consistency. By leveraging the properties already provided by Zookeeper our protocol is considerably simpler than state of the art in reconfiguration protocols. Our protocol also encompasses the clients – clients are rebalanced across servers in the new configuration, while keeping the extent of migration proportional to the change in membership.

3.19 Storing data to the Intercloud

Marko Vukolic (Eurecom, FR)

Joint work of Basescu, Cristina; Cachin, Christian; Eyal, Ittay; Haas, Robert; Sorniotti, Alessandro; Vukolic, Marko; Zachevsky, Ido

Main reference Cristina Basescu, Christian Cachin, Ittay Eyal, Robert Haas, Alessandro Sorniotti, Marko Vukolic, Ido Zachevsky: Robust data sharing with key-value stores. DSN 2012: 1-12

A key-value store (KVS) have become the most popular way to access Internet-scale 'cloud' storage systems. In short, KVSs offer simple functions for storing and retrieving values associated with unique keys. Precisely because of the limited interface of a KVS, textbook-style solutions for reliable storage either do not work or incur a prohibitively large storage overhead.

We present an efficient wait-free algorithm that emulates multi-reader multi- writer storage from a set of potentially faulty KVS replicas in an asynchronous environment. Our implementation serves an unbounded number of clients that use the storage concurrently. It tolerates crashes of a minority of the KVSs and crashes of any number of clients. Our algorithm minimizes the space overhead at the KVSs and comes in two variants providing regular and atomic semantics, respectively.

Compared with prior solutions, our algorithm is inherently scalable and allows clients to write concurrently. It is hence a desirable solution for improving data availability in the Intercloud setting, i.e., beyond the availability of a single cloud provider.

Participants

- Artur Andrzejak
Universität Heidelberg, DE
- Alysson Neves Bessani
University of Lisboa, PT
- Herbert Bos
VU - Amsterdam, NL
- Sara Bouchenak
INRIA Rhône-Alpes, FR
- Gregory Chockler
IBM - Haifa, IL
- Miguel Pupo Correia
IST - TU of Lisbon, PT
- Maria Couceiro
INESC-ID - Lisboa, PT
- Alexandra Dmitrienko
Fraunhofer Inst. - Darmstadt, DE
- Dan Dobre
NEC Laboratories Europe - Heidelberg, DE
- Kurt Geihs
Universität Kassel, DE
- Nicolas Geoffray
Google - Aarhus, DK
- Gabriela Gheorghe
University of Luxembourg, LU
- Stephan Groß
TU Dresden, DE
- Flavio Paiva Junqueira
Yahoo Research - Barcelona, ES
- Rüdiger Kapitza
TU Braunschweig, DE
- Nikola Knezevic
IBM Research - Zürich, CH
- Guillaume Pierre
VU - Amsterdam, NL
- Peter R. Pietzuch
Imperial College London, GB
- Hans Peter Reiser
Universität Passau, DE
- Luis Rodrigues
Technical University - Lisboa, PT
- Ahmad-Reza Sadeghi
TU Darmstadt, DE
- Nuno Santos
MPI für Softwaresysteme - Saarbrücken, DE
- Matthias Schunter
INTEL ICRI - Darmstadt, DE
- Marc Shapiro
INRIA & LIP6, Paris, France
- Alexander Shraer
Yahoo! Research - Santa Clara, US
- Radu Sion
Stony Brook University, US
- Jan Stoess
KIT - Karlsruhe Institute of Technology, DE
- Paulo Jorge Verissimo
University of Lisboa, PT
- Marko Vukolic
Symantec Research Labs - Biot, FR
- Michael Waidner
TU Darmstadt, DE
- Alexander Wiesmaier
AGT Group (R&D) GmbH - Darmstadt, DE

Report from Dagstuhl Seminar 12282

Database Workload Management

Edited by

Shivnath Babu[1], Goetz Graefe[2], and Harumi A. Kuno[3]

1 **Duke University, USA** shivnath@cs.duke.edu
2 **HP Labs, USA** goetz.graefe@hp.com
3 **HP Labs, USA** harumi.kuno@hp.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 12282 "Database Workload Management". Dagstuhl Seminar 12282 was designed to provide a venue where researchers can engage in dialogue with industrial participants for an in-depth exploration of challenging industrial workloads, where industrial participants can challenge researchers to apply the lessons-learned from their large-scale experiments to multiple real systems, and that would facilitate the release of real workloads that can be used to drive future research, and concrete measures to evaluate and compare workload management techniques in the context of these workloads.

Seminar 09.–13. July, 2012 – www.dagstuhl.de/12282
1998 ACM Subject Classification H.2 Database Management, H.2.4 Systems, H.2.7 Database Administration
Keywords and phrases database workload management, robust query processing, cloud computing, query execution, hadoop, application availability, performance modeling
Digital Object Identifier 10.4230/DagRep.2.7.73

1 Executive Summary

Shivnath Babu
Goetz Graefe
Harumi A. Kuno

Introduction

Much database research focuses on improving the performance of individual queries. Workload management focuses on a larger question – how to optimize the performance of the entire workload, as a whole. Workload management is one of the most expensive components of system administration. Gartner listed workload management as the first of two key challenges to emerge from the data warehouse market in 2009. However, we believe that even while both researchers and industry are building and experimenting with increasingly large-scale workloads, there is a disconnect between the OLTP/OLAP/Mixed/Hadoop/Map-Reduce workloads used in experimental research and the complex workloads that practitioners actually manage on large-scale data management systems.

One goal of this seminar was to bridge this gap between research and practice. Dagstuhl Seminar 12282 provided a venue where researchers can engage in dialogue with industrial participants for an in-depth exploration of challenging industrial workloads, where industrial participants can challenge researchers to apply the lessons-learned from their large-scale

Database Workload Management, *Dagstuhl Reports*, Vol. 2, Issue 7, pp. 73–91
Editors: Shivnath Babu, Goetz Graefe, and Harumi Anne Kuno
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

experiments to multiple real systems, and that would facilitate the release of real workloads that can be used to drive future research, and concrete measures to evaluate and compare workload management techniques in the context of these workloads.

With regard to seminar participants, we took a system-centric focus, and invited participants who could speak to the management of workloads in a variety of systems. Seminar participants came from a variety of academic and commercial institutions: Cloudera, EMC/-Greenplum, LinkedIn, Microsoft, MIT, National University of Singapore, NEC, Queen's University, Stony Brook University, Teradatata, Tokutek, TU Berlin, TU Berlin, TU Ilmenau, TU München, UC Berkeley, Universität des Saarlandes, Universität Hamburg, University of Waterloo, and Yahoo.

2 Table of Contents

3 Overview of the week

The week was structured around producing the following artifacts:

1. Descriptions of the most significant database workload management challenges facing industry, each defined in terms of a rough specification of a target workload and its objectives.
2. For some number of these challenges, a sample workload that would demonstrate the challenge, and that would allow solutions to the challenge to be validated and compared.
3. Descriptions of the "best" workload management techniques and best practices (both proven and unproven) that might apply to these challenges (both in practice and in research), as well as a partially-annotated bibliography that lists the papers that discuss those techniques and that summarizes their potential benefits and limitations.
4. Identify new synergies and opportunities for new techniques and new applications of existing techniques.

On Monday, industry participants spoke about the top workload management challenges commercial systems face, and other participants reviewed ongoing work on benchmarks that expose those challenges. Tuesday featured a series of presentations surveying the current state of the art in terms of research. On Wednesday, participants formed break out groups and considered the commercial challenges and how best to cast them as challenges that researchers could address. The break-out groups reformed on Thursday, and sketched paper abstracts of research papers. On Friday morning, these abstracts were presented.

4 Monday: Individual presentations

On Monday, we focused on commercial workload management systems, their open challenges, and ongoing work on benchmarks that expose those challenges.

4.1 Monday morning: commercial workload managmeent

Teradata has the possibly most well-established and sophisticated workload management system of any commercial database system, as evidenced by a recent keynote that credits Teradata's workload management as a key component of eBay's success. It was thus fitting that *Douglas Brown* from *Teradata* kicked off the first day with a presentation about Teradata's workload management capabilities.

Rao Kakarlamudi from *HP* presented an overview of HP Seaquest's workload management system. The HP Workload Management Services (WMS) provides the infrastructure to help you manage query workloads and system resources in the mixed workload environment of a SeaQuest Data Warehousing Platform for Business Intelligence. In the mixed workload environment of an enterprise data warehouse, like the SeaQuest platform, a variety of query workloads must run smoothly without interruptions to their throughput and performance. As a database administrator, you need to be able to control when different types of queries enter the system and how they proceed to execute, and you need to ensure that system resources remain available for query execution. You do not want a rogue query to monopolize system resources and slow down or prevent other queries from running in the system. Using WMS, you can influence when queries run and how many system resources they are allowed to consume by assigning groups of queries (that is, query workloads) to WMS services. You can create your own services in WMS and configure them to have a relative priority and a set of thresholds for the execution environment. That way, you can maintain different query workloads and ensure that enough system resources are available to higher priority workloads while minimizing contention with lower priority workloads.

Sivaramakrishnan Narayanan from *EMC Greenplum* spoke about workload management at EMC Greenplum.

Russell Sears talked about his work at *Yahoo*: This talk summarizes Yahoo!'s current and next-generation data serving infrastructure. We provide an overview of three systems, Mobius, Walnut and bLSM, which promise to enable new classes of applications and to significantly improve the performance of Yahoo!'s existing workloads. However, fully leveraging these systems will require solutions to new workload management problems. We present the problems, and suggest new approaches to workload management. Mobius allows the workload manager to impact user-visible application semantics as well as performance SLAs. Walnut is designed to leverage extreme mismatches in system scale to provision services on spare capacity. Finally, unlike many data management systems, bLSM is designed to run in carefully provisioned environments, allowing its latency and throughput to be predicted directly from hardware parameters. We conclude the talk with a discussion of recent hardware trends.

Robert Chansler from *LinkedIn* spoke about LinkedIn Workflows In Large Hadoop Clusters. A workflow might be a coordinated collection of 20 job where each job might have a 1000 tasks (maps or reduces). A large Hadoop cluster will see thousands of jobs each day. For a carefully configured job, the number of tasks will be proportional to the input size (1 task 500 MB). Today the management of work load is limited to simple assignment of tasks to a small number of queues. Some conventional management tools are problematic for Hadoop—throttling and preemption have never been satisfactorily implemented

Archana Ganapathi, from *Splunk*, spoke about her experiences with managing Big Data Workloads in Splunk.

Jingren Zhou spoke about *Microsoft*'s Bing infrastructure and workloads.

4.2 Monday afternoon: benchmarks and workloads

In the afternoon, we heard about benchmarks and workloads. *Kai-Uwe Sattler* from *TU Ilmenau* spoke about the Tractor Pulling for DBMS Benchmarking, which began as a collaboration into benchmarking robust query processing at the 2010 Dagstuhl Seminar 12282 on Robust Query Processing and presented at DBTest 2011.

Michael Seibold from *TU München* spoke of the Mixed Workload CH-benCHmark from the Technical University of Munich. While standardized and widely used benchmarks address either operational or real-time Business Intelligence (BI) workloads, the lack of a hybrid benchmark led us to the definition of a new, complex, mixed workload benchmark, called mixed workload CH-benCHmark. This benchmark bridges the gap between the established single-workload suites of TPC-C for OLTP and TPC-H for OLAP, and executes a complex mixed workload: a transactional workload based on the order entry processing of TPC-C and a corresponding TPC-H-equivalent OLAP query suite run in parallel on the same tables in a single database system. As it is derived from these two most widely used TPC benchmarks, the CH-benCHmark produces results highly relevant to both hybrid and classic single-workload systems. Like the "Tractor Pulling" benchmark, this work was initiated at the 20101 Dagstuhl Seminar 12282 on Robust Query Processing and presented at DBTest 2011.

Finally, *Yanpei Chen* from *UC Berkeley* (he has since joined *Cloudera*) gave two presentations about on Hadoop/MapReduce workloads.

Interactive Analytical Processing in Big Data Systems: A Cross Industry Study of MapReduce Workloads. Within the past few years, organizations in diverse industries have adopted MapReduce-based systems for large-scale data processing. Along with these new users, important new workloads have emerged which feature many small, short, and increasingly interactive jobs in addition to the large, long-running batch jobs for which MapReduce was originally designed. As interactive, large-scale query processing is a strength of the RDBMS community, it is important that lessons from that field be carried over and applied where possible in this new domain. However, these new workloads have not yet been described in the literature. We fill this gap with an empirical analysis of MapReduce traces from six separate business-critical deployments inside Facebook and at Cloudera customers in e-commerce, telecommunications, media, and retail. Our key contribution is a characterization of new MapReduce workloads which are driven in part by interactive analysis, and which make heavy use of query-like programming frameworks on top of MapReduce. These workloads display diverse behaviors which invalidate prior assumptions about MapReduce such as uniform data access, regular diurnal patterns, and prevalence of large jobs. A secondary contribution is a first step towards creating a TPC-like data processing benchmark for MapReduce.

We Don't Know Enough to make a Big Data Benchmark Suite — An Academia-Industry View. This is a position paper that comes from an unprecedented empirical analysis of seven production workloads of MapReduce, an important class of big data systems. The main lesson we learned is that we do not know much about real life use cases of big data systems at all. Without real life empirical insights, both vendors and customers often have incorrect assumptions about their own workloads. Scientifically speaking, we are not quite ready to declare anything to be worthy of the label "big data benchmark." Nonetheless, we should encourage further measurement, exploration, and development of stopgap tools.

5 Summarizing Challenges and Open Questions from Commercial Systems

In the late afternoon, we broke into groups and focused on drilling down, categorizing and prioritizing the challenges we'd heard about from commercial systems. These challenges are captured below.

5.1 Workload Characterization/ Performance Modeling

Doug Brown, Yanpei Chen, Jens Dittrich, Archana Ganapathi, Harumi A. Kuno, Barzan Mozafari, Awny Al-Omari, Norbert Ritter, Y.C. Tay

5.1.1 Industry challenges

1. What will be the impact of expected growth and planned changes?
2. How to set realistic service level goals (SLGs)?
3. How to change a workload's priority (relative weight) to meet SLGs?

4. How to set workloads' concurrency levels (TASM throttling) to meet SLGs?
5. How to justify tuning measures to meet SLGs?
6. How to predict new application implementation impact?
7. How to justify hardware upgrades required to meet SLGs?
8. How to compare actual performance with expected?
9. Memory vs. CPU isolation
10. Empirical validation
11. How to model all models
12. A tool to take a real workload and scale it up or down with "relativity" (specifics TVD).
13. How to achieve resource isolation between workloads.

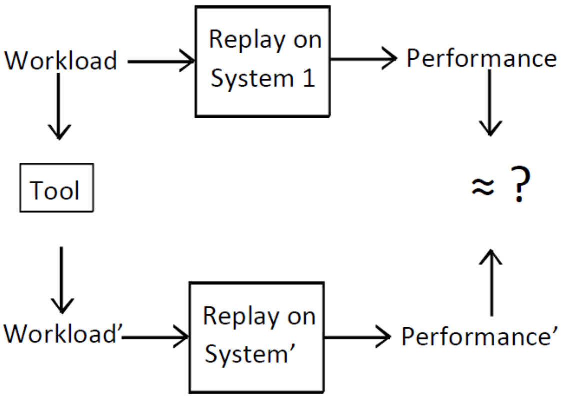

Figure 1 Sketch by the workload characterization/performance modeling breakout group summarizing the underlying challenge of workload characterization / performance modeling.

5.1.2 Top priorities (workload management)

1. How to change workload's priority (rela0ve weight) to meet SLGs?
2. How to set workloads' concurrency level to met SLGs?
3. How to predict new applicaion's implementatation impact?

5.1.3 Top Priorities (capacity planning)

1. How to change workload's priority (relative weight) to meet SLGs?
2. How to achieve resource isolation between workloads.
3. How to predict new application implementation impact?

5.2 Workload Isolation

Ashraf Aboulnaga, Shivnath Babu, Robert Chansler, Hakan Hacigümus, Rao Kakarlamudi, and Michael Seibold

The group defined workload isolation as:

1. *Space multiplexing of resources.* Each of the N workloads achieves the same performance that it would if it were running alone with unlimited computing resources

2. *Time multiplexing of resources.* Each of the N workloads is promised a share of the available computing resources, and it gets at least this share of the resources
3. *Application view of performance.* Each of the N workloads meets its SLOs

5.2.1 Research challenges

1. Level 1: Given 1 workload and its application level SLO, give me the resource requirements of this workload
2. Level 2: Given N workloads and application level SLOs, give me the resource requirements of each workload
3. Level 3: Given N workloads and application level SLOs that vary over time, give me the resource requirements of each workload
4. Workload Isolation Objective: Meet all requirements with minimum computing resources
   - It is easy to isolate workloads by overprovisioning

5.2.2 Open questions

1. What actions can a system take to ensure workload isolation?
   - Data replication, admission control, resource allocation, preemption, time multiplexing vs. space multiplexing
   - Not all actions applicable to all systems
2. Is virtualization a solution out of the box?
   - Dealing with multiple levels of virtualization that do not match with each other
3. Can elasticity be used as a mechanism to address overprovisioning?
   - How to provide elasticity in data intensive systems
   - Elasticity helps with changing workloads and SLAs
4. How to map from application level SLO to resource demands?
5. Quantifying and modeling the degree/impact of performance isolation.
   - Resource based metrics
   - SLA based metrics
6. Is data locality overrated given new developments in network
   - Inter-cluster bandwidth has gone up
   - Infiniband is becoming more popular
7. Can we just assume that data is spread across all available nodes and focus on compute-level isolation?

5.2.3 Benchmarking Workload

- Analytical workload
- SQL or MapReduce both possible
  - Capability and complexity (SQL) vs. predictability (MapReduce)
  - TPC-H, WordCo-occurrence
- Vary the number of queries/jobs from 1 to N
  - Control the amount of data and degree of overlap in data access (e.g., by varying a filter on a time attribute)
  - Control degree of overlap in arrival times
  - Control degree of similarity between workloads

5.3 Scheduling Workloads for Concurrency

Kai-Uwe Sattler and others

This break-out group considered the top-three challenges in the area of scheduling theory, performance prediction, and no-knob data management.

1. Scheduling theory
   - Goal: provide a performance theorem for: Given a DAG in which the nodes have CPU work W & I/O work I running on a machine with P processors & D disks, a simple scheduler can run the DAG with good performance and memory bounds
   - Why:
     - Theoretical foundation for scheduling
     - Provable property
   - How to test:
     - By proving
     - Implement and validate the performance against the theorem
   - Why:
     - Theoretical foundation for scheduling
     - Provable property
   - Challenge: In analogy with Brent's theorem which states that a graph with work W and span S running on P processors can run in time less than $W/P + S$. Extras (may be harder):
     - P and D change dynamically.
     - Processor and disk speeds change dynamically.
     - What is the locality problem for this kind of computation?
     - Given several such DAGs, how do you execute them concurrently (and what kind of bounds do you get)?
     - The DAG unfolds dynamically.
2. Performance prediction.
   - Goal: predict performance of individual requests as well as the overall workload
   - Why: required for scheduling, but difficult — see 30 years of research on query optimizers
   - No knobs
   - How to test: compare predictions to real execution times.
3. No knobs.
   - Goal: Eliminate tuning knobs in WLM (memory, parallelism, MPL, ...).
   - Why: Base for other techniques; cost issues, corrective actions.
   - First Step: Start with eliminating memory knobs (e.g., memory-adaptive sorting).
   - How to test: Compare with gold standard in static and dynamic settings.

For workloads, see powerpoint slideset in seminar materials.

6 Tuesday: individual presentations

Tuesday focused on research efforts relevant to the challenges highlighted on Monday.

6.1 Tuesday morning: performance modeling/prediction, algorithms

Archana Ganapathi, from *Splunk* but speaking of her work at *UC Berkeley.* Archana spoke about two topics: 1. Using Statistical Machine Learning to predict performance for a parallel database system. 2. Workload Synthesis and Replay for MapReduce.

Y.C. Tay from *National University of Singapore* presented his paper *Bottleneck Analysis for Cloud Transaction Architectures.* At the SIGMOD 2010 conference, Kossman, Kraska and Loesing presented an experimental comparison of four cloud architectures for transaction processing. The paper concluded that "It is still unclear whether the observed results are an artifact of the level of maturity of the studied services or fundamental to the chosen architecture". This issue is addressed here via a theoretical analysis that focuses on the bottleneck in each architecture.

Jens Dittrich from *Universität des Saarlandes* presented his work on Hadoop++/HAIL.

Norbert Ritter from *Universität Hamburg* presented an alternative solution to the problem of DBS self-management, which avoids the drawbacks of the existing self-management functions (as described in the PhD thesis of Marc Holze). Instead of extending a DBMS with a set of component-specific self-management functions, the developed solution is designed as one single self-management loop, which has a system-wide view on all configuration decisions. As long as the workload of the system does not change and the goals are met, this self-management loop only performs very light-weight monitoring operations on the workload, performance, and state information. For this purpose, a workload shift detection solution is provided. This also comprises a workload classification solution, that groups similar workload events in order to further reduce the monitoring overhead. Furthermore, the workload information is analysed for cyclic patterns in order to predict upcoming workload shifts. Only when the system-wide self-management solution detects a workload shift or a violation of the goals, it performs a heavy-weight reconfiguration analysis. Given the current workload and state of the DBS, this reconfiguration analysis has to derive a new set of configuration parameter values that meet the goals in the best possible way. For this purpose the system-wide self-management solution employs a system model, which quantitatively describes the behaviour of the DBS using mathematical models. At runtime, the system model is evaluated by the self-management logic using multi-objective optimization techniques, where the goal values are represented as constraints.

6.2 Tuesday afternoon: mixed workloads and robust query processing

Wendy Powley from *Queen's University – Kingston* provided an overview of the techniques and approaches that her research group has taken to workload management in DBMSs over the past 10 years.

Ashraf Aboulnaga from *University of Waterloo* provided an overview of his research on workload management, focusing on (1) query interactions, (2) workload management for Hadoop, and (3) virtualization.

Bradley Kuzsmaul (from *MIT* and *Tokutek*) and *Michael Bender* (from *State University of New York at Stony Brook* and *Tokutek*) spoke of Cilk, work stealing, write-optimized data structures, and scheduling algorithms.

Goetz Graefe from *HP Labs* spoke about robust query processing.

Kostas Tzoumas from *TU Berlin* spoke of his group's work on Stratosphere.

Shivnath Babu from *Duke University* presented "Perspectives on MapReduce Workload Management." Abstract: This talk gives an overview of problems that arise in workload management for MapReduce systems. We first discuss the reasons why the importance of MapReduce workload management has grown rapidly in recent years. We then present the preliminary techniques being used in the industry today for this problem, and why these techniques are inadequate. With the goal of laying out a research agenda, the talk concludes with an abstraction of the various aspects involved in MapReduce workload management.

7 Working Groups (Wednesday, Thursday, Friday)

For the second half of the week, participants worked in break out groups and considered the commercial challenges and how best to cast them as challenges that researchers could address.

7.1 Towards a Benchmark for Workload Management

Ashraf Aboulnaga, Awny Al-Omari, Shivnath Babu, Robert J. Chansler, Hakan Hacigumus, Rao Kakarlamudi, and Michael Seibold

Summary: This breakout group focused on (i) developing a framework for benchmarking workload management techniques, (ii) coming up with some example instantiations of this framework, and (iii) identifying use cases in which the proposed benchmark can be of value.

The focus was on benchmarking analytical or decision support systems, as opposed to transactional systems. The typical workflow in such systems is that data comes in through one or more feeds that are loaded into the system, and users submit analytical queries that are executed on the available data. The system must provide desired guarantees (SLAs) on data freshness (how fast is data loaded) and query response times. The system has to maintain these guarantees in the face of fluctuations in the workload and failures of the infrastructure; which is the task of workload management.

Given the above usage scenario, the benchmark framework that we developed requires benchmark creators to define the following six pluggable components:

1. Data feeds
2. Query workloads
3. SLAs and penalties for violating them
4. Failures (optional)
5. Temporal patterns for the arrival of data and queries, and for system failures
6. Scaling rules

A benchmark defined through this framework evaluates how well a workload management solution fulfills the requirements of the presented data feeds and query workloads while minimizing total cost. Thus, the benchmark measures performance metrics such as throughout and latency, and how they vary over time (a time series). The benchmark also measures the number of SLA violations and the total cost of the system, defined as Cost C = R + P, where R is the cost of the infrastructure and P is the cost of SLA violation penalties.

Temporal arrival patterns: A key feature of the benchmark framework is that the arrival of data and queries is described by temporal arrival patterns. These temporal patterns are also used to describe when failures happen. The temporal patterns can be deterministic or probabilistic. If they are probabilistic, then they are generated prior to a benchmark run to ensure repeatability. The patterns can be uniform or bursty. If they are bursty, then it is important to control whether the peak arrival rates happen in a correlated fashion (e.g., simultaneously) or independently. Simultaneous peaks stress workload management solutions.

Data feeds: Benchmark data may have some schema, which needs to be specified. A benchmark also has one or more data feeds. The data has to have a temporal dimension (a timestamp attribute) and has to be generated in increasing timestamp order. The data that appears in the feeds is described by the distribution for data generation (uniform or skewed, data feeds correlated or not, etc.).

The arrival pattern of the data is specified by a temporal arrival pattern. The temporal arrival pattern can specify two types of data loading: batch loading which happens every once in a while (e.g., initial loading into the system or periodic updates to dimension tables), and continuous streams (e.g., updating the fact tables). Batch loading would be described by a spike in the temporal arrival pattern.

Query workloads: A benchmark has one or more query workloads.Each workload consists of a stream of analytical queries (simple queries for tactical workloads and more complex queries for reporting) that read controlled amounts of data. The queries, or at least a significant fraction of them, must have a temporal predicate that restricts them to touching some time window of the data. The arrival of queries on the different workloads is defined by temporal arrival patterns, one per workload.

Queries have SLAs on how fast they need to be processed (query completion time). The specification of the queries also controls how much data overlap there is, which can have a significant effect on workload management, and how fresh the data needs to be, which places a constraint on data loading.

SLA penalties: SLAs are defined by a penalty function for each query type and each workload. The penalty function defines different penalties for missing different response time requirements. Different penalty functions allow us to express different priorities for workloads. For example, a penalty function in which the penalty is infinite means that we cannot under any circumstance violate the SLA. In general, there are types of workloads where there is significant (even if not infinite) penalty for violating the SLA, and others where violating the SLA is less harmful.

The cost of SLA violation is included in the total cost that is reported in the benchmark score. Another option is to count SLA violations and report them in the benchmark score.

Data overlap: Different workloads are more likely to affect each other if they touch the same data. One way to control data overlap is to have different queries touch data in different time windows. For example, we can have two workloads that both always touch the most recent data, which creates a high degree of overlap. On the other hand, we can have one workload that touches the most recent data and another workload that touches some fixed data in the past, which results in zero overlap. It may also be possible to control data overlap in other ways, for example by having queries look at the same tables or different tables.

Freshness: The correlation between the query arrival patterns and the data arrival patterns controls how fresh the data has to be. If a query is supposed to see some recent data, and this data has arrived but not been loaded yet, then the query will return a wrong answer.

To give the system some slack in the time to load the data, the benchmark queries should be allowed to touch queries only F seconds in the past or older. That is, the time window predicates that are supposed to touch the most recent data in the different queries are of the form: "WHERE timestamp BETWEEN pasttimestamp AND NOW() – F}". The higher the F, the more flexibility a system has in delaying the load of new data. In that sense, F is a freshness target.

A benchmark needs to specify a penalty cost that is incurred by a query if it does not see recent data that it is supposed to see. We can determine when this happens because the correct answer of each query – or how to obtain this correct answer – is part of the benchmark specification. The penalty cost for violating freshness is added into the total cost reported in the benchmark score. An infinite penalty cost means that a system is not allowed to violate freshness.

Failures: In today's world, it is important to consider failures when talking about workloads and workload management. A lot of work these days goes into making systems fault tolerant, and systems are often overprovisioned or geographically replicated to enable them to handle failures. For example, if you consider a system that is mirrored for high availability, such a system is incurring a cost overhead of 100

We envision two ways to incorporate failures into the benchmark:

1. The benchmark can specify a temporal arrival pattern for failures chosen from a specific set of failure types (e.g., node and rack failures). For example, "After 30 minutes of running the benchmark, a rack fails for 10 minutes and then recovers." The issue with this strategy is that specifying the meaning of failure and recovery from failure is not easy. Pulling the plug on the entire cluster is easy to understand, but what does it mean to lose 30

2. Instead of specifying failure types, the benchmark specifies the effect of these failures on the workload. For example, "After 30 minutes of running the benchmark, the data feed becomes unavailable for 10 minutes and, for the next 20 minutes the workload is double the past average as the system catches up."

A benchmark needs to specify the expected behavior of data feeds and query workloads during the failures. It is recommended that the data feeds continue while the system is failed. Queries can either continue arriving and incur a penalty for failing, or we can model a case in which the query streams stop when the system fails.

Scaling rules: A benchmark must specify scaling rules. A simple way to do scaling is to say that each component (data, queries, arrival patterns) scales independently. However, we acknowledge that more elaborate scaling rules may be needed.

The next steps for the breakout group are to come up with some example instantiations of this framework, and to identifying use cases in which the proposed benchmark can be of value.

7.2 Scheduling Workloads for Concurrency

7.3 Provably Good Scheduling for Database Workload Management

Bradley C. Kuszmaul (ed); Breakout Session B.

Problem 1. How should you schedule the DAG that includes CPU work W and I/O work I on a machine with P processors and D disks? What performance and space guarantees can you make? These variants may be useful: P and D may change dynamically. What if the processors or disks change speed dynamically? What is the locality problem for this kind of computation? Given several such DAGs, how do you schedule them when they are running at the same time? The DAG unfolds dynamically (as in Cilk).

Definition 2. The span of a DAG is the length longest path through the graph. (Sometimes this is referred to as depth or critical-path length.)

Definition 3. The span of the work of a graph G, written SW, is the length of the longest path through the graph, counting only the CPU work.

Definition 4. The span of the I/O of a graph G, written SI , counts only the I/O work.

Conjecture 5. A greedy schedule achieves time O(W=P+I=D+SW +SI), where I/O and CPU are both measured in units of time. Idea for a proof. The idea is to extend Brent-Graham as follows: At any time step at least one of the following is true: half the processors are busy, half the disks are busy, or there are some idle processors and idle disks. The number of time steps that half the processors are busy is at most 2W=P. The number of time steps that half the disks are busy is at most 2I=D. If more than half the disks are idle and half the processors are idle, then ... Application 6. If a query optimizer has several query-plan choices, it needs to be able to estimate the run time of each plan. If we start running queries using work stealing, we need a way to predict the performance. Conjecture 5, if true, may provide a way for the user or the query planner to predict the performance of a plan on a machine with D disks and P processors.

7.4 Workload Characterization/ Performance Modeling

Doug Brown, Yanpei Chen, Jens Dittrich, Archana Ganapathi, Harumi A. Kuno, Barzan Mozafari, Awny Al-Omari, Norbert Ritter, Y.C. Tay

Database vendors, developers, and administrators need to model and predict database performance given a workload and a system configuration in order to perform tasks such as workload management, capacity management, and system sizing. However, lack of data about workload characterization and workload performance is a significant obstacle for researchers working on modeling and prediction. Our breakout session produced a table showing functional workload types, logical workload characteristics that could be used to describe those types, and physical performance characteristics that could be observed from actual running workloads. We also proposed two tools. First, we proposed a workload characterization editor that could take an existing workload characterization and then manipulate one or more logical workload characteristics, producing a new workload characterization. We proposed that this tool could be made open and published, along with a repository of editable workload characterizations. Second, we proposed a tool that could take the resulting workload characterization and run the described workload. Finally, we described four use cases of how the workload editor could be used: (1) how to anticipate the performance impact of anticipated workload growth, (2) how to anticipate the performance impact of a workload throttling mechanism on a given workload; (3) how to compare the performance for two potential vendors (or systems); and (4) how to anticipate the performance impact of a new application.

A subset of the group co-authored a paper pursuing this idea further.

7.5 Eliminating memory knobs

Goetz Graefe, Wendy Powley, Kai-Uwe Sattler, Kostas Tzoumas, Jingren Zhou

One of the tasks of workload management is to allocate resources to consumers with conflicting demands. Systems typically contain "knobs" that dictate resource allocation, for example how much memory should be allocated to the buffer pool, how much memory should be allocated to sorting, etc. Eliminating knobs for tuning data management systems is an important task to reduce maintenance costs and make the systems more usable to standard customers. Eliminating knobs makes the problem of workload management significantly easier by removing free variables.

We consider the problem of memory tuning in such systems as a building block towards this overall goal: given an externally defined amount of memory (which may vary over time), how should this memory be internally distributed among different heterogeneous consumers? Using a basic model taking utility functions of the different consumers into account, we formulate this problem as an optimization problem and present scenarios as well as metrics for comparing and evaluating different strategies. Furthermore, we argue that consumers should be memory-adaptive in order to provide a diminishing marginal utility function which simplifies the optimal memory distribution significantly. Finally, we discuss how this can be achieved for typical memory consumers in a database system.

Different memory consumers that we aim to model in the context of a DBMS are the buffer pool (including the case of several buffer pools for multiple page sizes, devices, tables vs. indexes, etc.), query execution (including sorting, hashing, exchange, etc), procedure cache (compiled query execution plans and scripts), metadata, database utilities (e.g., load, reorganization, compression, index creation, statistics/histograms, etc), log buffers, query optimizer, complex UDFs (DB2 application memory and database heap), and UDFs in garbage-collected languages such as C# or Java [1, 2].

A utility function describes the quantification of the performance for a given amount of resource. In the context of memory management, we can quantify performance as the reduction in IO rate. Utility functions are needed as an indication for performance prediction to trade resource allocations between different consumers. It is important that utility functions for all memory consumers are comparable, i.e., they are expressed in the same units. IO rate reduction can be predicted, e.g., in the case of query optimizer by predicting the expected reduction in execution plan cost if further exploration will take place.

A beautiful utility function obeys Gossen's first law, i.e., is a monotonically increasing function with decreasing marginal gain. With beautiful utility functions, a simple strategy that trades resources based on possible gains converges to a solution that optimizes total utility [3]. Examples of non-beautiful utility functions include linear functions, step functions, and non-continuous functions. In the context of a DBMS, typical examples of operators with non-beautiful utility functions are sorting without graceful degradation, non-hybrid hashing, and LRU replacement strategy without scan protection.

Some memory consumers may be simply served by virtual memory paging to shrink their memory allocation. For other consumers, the internal data structures need to be adjusted. One research direction is changing the database engine in order to make utility functions beautiful. A core strategy for making utility functions beautiful is applying graceful degradation, i.e., the gradual and incremental use of page eviction, while retaining as much state as possible in the available memory. This enables a graceful transition from an in-memory to an external memory algorithm in presence of pressure. Examples include hybrid hashing [4] and adaptive-memory sort. For query optimization, an option would be a resource-guided query optimization enumeration strategy that prunes plans based on memory constraints. Finally, for data exchange in parallel DBMSs [5], a possible strategy would be to reduce the network buffers using hierarchical partitioning [7].

A second research direction is formalizing and solving the allocation problem. We can formalize memory distribution as an optimization problem:

We maximize the overall utility (possibly weighted to express externally-dictated priorities) minus the cost of adjustment subject to the externally dictated memory constraints. Assuming that utility functions are beautiful and all weights are equal to one, the problem can be solved by a simple trading approach. While one consumer's pain is less than another consumer's gain, we trade memory. The cost of trading is the aggregated utilities penalties over the adjustment period. To avoid expensive oscillation a minimum gain/pain can be imposed.

Previous work in adaptive memory management [6] notes "[..] the difficulty in determining the suitable experiment to test the efficiency of automatic memory tuner," and "the absence of any standard metric for evaluation..." A further research direction is devising scenarios for testing the performance, scalability, and robustness of various methods that (claim to) eliminate memory knobs. A final research direction is generalizing the problem to include other resources. While our outset is memory knobs, this work can be possibly generalized to other knobs on data processing systems (such the degree of parallelism). Two issues must be addressed by this generalization. First, appropriate utility functions for the resource usage have to be defined. Second, the algorithms have to be adapted to graceful degradation strategy for dealing with changing resources.

References

1 Dominic Battré, Stephan Ewen, Fabian Hueske, Odej Kao, Volker Markl, Daniel Warneke: Nephele/PACTs: a programming model and execution framework for web-scale analytical processing. SoCC 2010:119-130

2 Ronnie Chaiken, Bob Jenkins, Per-Åke Larson, Bill Ramsey, Darren Shakib, Simon Weaver, Jingren Zhou: SCOPE: easy and efficient parallel processing of massive data sets. PVLDB 1(2):1265-1276 (2008)

3 Diane L. Davison, Goetz Graefe: Dynamic Resource Brokering for Multi-User Query Execution. SIGMOD 1995:281-292

4 Goetz Graefe: Query Evaluation Techniques for Large Databases. ACM Comput. Surv. (CSUR) 25(2):73-170 (1993)

5 Goetz Graefe: Encapsulation of Parallelism in the Volcano Query Processing System. SIGMOD 1990:102-111

6 Adam J. Storm, Christian Garcia-Arellano, Sam Lightstone, Yixin Diao, Maheswaran Surendra: Adaptive Self-tuning Memory in DB2. VLDB 2006:1081-1092

7 Jingren Zhou, Nicolas Bruno, Wei Lin: Advanced partitioning techniques for massively distributed computation. SIGMOD 2012:13-24

7.6 Cache-Adaptive Algorithms for Databases

Michael A. Bender and Siva Narayanan (Eds.) Breakout session B.

Many commercial database execution engines reserve memory for operations statically and offer no opportunity to change this at runtime. This causes under- utilization and queries go slower than then can. Cache-adaptive algorithms offer a glimpse of hope.

We introduce class of cache-adaptive algorithms. Cache-adaptive algorithms adapt to memory parameters that change over time, e.g., the available memory M and the block-transfer size B. We exhibit optimal cache-adaptive algorithms for sorting, matrix multiplication, and sort-merge-join and hash join.

We first prove that so-called cache-oblivious algorithms are also cache-adaptive. Cache-oblivious algorithms are memory-hierarchy universal, i.e., the same algorithm works simultaneously on all memory-hierarchies and with no memory- specific parameterization. We then exhibit algorithms, whose parameters depend on B and M, and show that they are adaptive.

We first model how B and M are allowed to change. We then give a formal definition what it means to be optimal.

If we can prove this:

- Query execution engine can adapt its memory usage.
- Utilization can be maximized.
- Workload management can be more intelligent about memory as a resource.

7.7 Dataflow programming atop Cilk

Russell Sears, Bradley Kuzsmaul, Michael Bender (and others?).

Cilk is a superset of the C++ programming language for parallel programming tasks. It introduces two new keywords, "spawn" and "sync" that allow the runtime environment to optionally run portions of a computation in parallel on a second processor.

Dataflow programs are inherently parallel, and are therefore a seemingly natural fit for languages such as Cilk. The most natural way to encode relational query trees in Cilk is to treat each relational operator as an iterator embedded in a C function call graph. However, doing so yields invalid, deadlock-prone Cilk programs.

The problem is due to a mismatch between Cilk's "strict" semantics, which require function arguments to be executed before the function call itself, and dataflow programming semantics, which require each operator to be executed in parallel. Certain dataflow operators, such as merge join, stream data from multiple inputs without materializing intermediate results. In order to execute such programs, we must be able to guarantee that each operator makes progress independent of the other.

In a serial execution, it is possible that one of the join subqueries will be partially executed, and then block until its output is consumed. Similarly, the join algorithm will block until the other output produces data. Because the execution is serial, the second subquery will never be invoked, and the program will deadlock.

The solution seems to be to explicitly spawn a new pthread for each such operator. The Cilk runtime simultaneously guarantees that all pthread threads make progress, and that reasonable runtime scheduling decisions will be made, leveraging the inherent parallelism of each query operator.

In this work, we will examine the performance tradeoffs between conventional, dataflow-oriented execution environments and our new Cilk-style approach, characterizing queries for which each approach outperforms the other.

8 Post-seminar outcomes

A number of collaborations were begun during the seminar that continued after the seminar's end. Ashraf Aboulnaga and Shivnath Babu collaborated on a tutorial proposal, “Workload Management for Big Data Analytics”, that has been accepted for ICDE 2013. Douglas Brown from Teradata began a collaboration with Barzan Mozafari from MIT exploring performance modeling. Douglas Brown, Yanpei Chen, Jens Dittrich, Archana Ganapathi, Harumi A. Kuno, and Y. C. Tay from Teradata, Cloudera, Saarland University, Splunk Inc, HP Labs, National University of Singapore developed their break-out group's abstract into a vision statement (they are currently considering where to publish it). Robert Chansler from LinkedIn began a collaboration with Shivnath Babu's Starfish project at Duke University.

Participants

- Ashraf Aboulnaga
University of Waterloo, CA
- Awny Al-Omari
Teradata, US
- Shivnath Babu
Duke University, US
- Michael A. Bender
State University of New York at Stony Brook, US
- Douglas Brown
Teradata – Rancho Santa Fe, US
- Robert J. Chansler
Linkedin, US
- Yanpei Chen
University of California – Berkeley, US
- Jens Dittrich
Universität des Saarlandes, DE
- Archana Ganapathi
Splunk Inc., USA
- Goetz Graefe
HP Labs – Madison, USA
- Hakan Haciguemus
NEC Laboratories America, Inc. – Cupertino, US
- Rao Kakarlamudi
HP – Palo Alto, US
- Harumi A. Kuno
HP Labs – Palo Alto, US
- Bradley C. Kuszmaul
MIT – Cambridge, US
- Barzan Mozafari
MIT – Cambridge, US
- Sivaramakrishnan Narayanan
EMC Corp. – Bangalore, IN
- Wendy Powley
Queen's Univ. – Kingston, CA
- Norbert Ritter
Universität Hamburg, DE
- Kai-Uwe Sattler
TU Ilmenau, Germany
- Russell Sears
Microsoft Research – Mountain View, US
- Michael Seibold
TU München, US
- Y. C. Tay
National Univ. of Singapore, SG
- Kostas Tzoumas
TU Berlin, DE
- Jingren Zhou
Microsoft Research – Redmond, US

Report from Dagstuhl Seminar 12291

Structure Discovery in Biology: Motifs, Networks & Phylogenies

Edited by

Alberto Apostolico[1], Andreas Dress[2], and Laxmi Parida[3]

1 **Georgia Institute of Technology, US,** axa@cc.gatech.edu
2 **Shanghai Institutes for Biological Sciences, CN,** andreas@picb.ac.cn
3 **IBM TJ Watson Research Center, US,** parida@us.ibm.com

Abstract

From 15.07.12 to 20.07.12, the Dagstuhl Seminar 12291 “Structure Discovery in Biology: Motifs, Networks & Phylogenies” was held in Schloss Dagstuhl – Leibniz Center for Informatics. The seminar was in part a follow-up to Dagstuhl Seminar 10231, held in June 2010, this time with a strong emphasis on large data. Both veterans and new participants took part in this edition. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar, as well as abstracts of seminar results and ideas, are put together in this report. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Seminar 15.–20. July, 2012 – www.dagstuhl.de/12291
1998 ACM Subject Classification algorithms, theory, pattern matching, text analysis, biology and genetics, health
Keywords and phrases mathematical biology, computational biology, algorithmic bioinformatics, pattern discovery, networks, phylogenetics, stringology
Digital Object Identifier 10.4230/DagRep.2.7.92
Edited in cooperation with Fabio Cunial

1 Executive Summary

Alberto Apostolico
Andreas Dress
Laxmi Parida

In biological systems, similarly to the tenet of modern architecture, form and function are solidly intertwined. Thus to gain complete understanding in various contexts, the curation and study of form turns out to be a mandatory first phase.

Biology is in the era of the “Omes”: Genome, Proteome, Toponome, Transcriptome, Metabolome, Interactome, ORFeome, Recombinome, and so on. Each Ome refers to carefully gathered data in a specific domain. While biotechnology provides the data for most of the Omes (sequencing technology for genomes, mass spectrometry and toponome screening for proteomes and metabolomes, high throughput DNA microarray technology for transcriptomes, protein chips for interactomes), bioinformatics algorithms often help to process the raw data, and sometimes even produce the basic data such as the ORFeome and the recombinome.

The problem is: biological data are accumulating at a much faster rate than the resulting datasets can be understood. For example, the 1000-genomes project alone will produce more than 10^{12} raw nucleic acid bases to make sense of. Thus, databases in the terabytes, even petabytes (10^{15} bytes) range are the norm of the day. One of the issues today is that our ability to analyze and understand massive datasets lags far behind our ability to gather and store the data with the ever advancing bio- and computing technologies. So, while the sheer size of data can be daunting, this provides a golden opportunity for testing (bioinformatic) structure-discovery primitives and methods.

Almost all of the repositories mentioned here are accompanied by intelligent sifting tools. In spite of the difficulties of structure discovery, supervised or unsupervised, there are reasons to believe that evolution endowed biological systems with some underlying principles of organization (based on optimization, redundancy, similarity, and so on) that appear to be present across the board. Correspondingly, using evolutionary thoughts as a "guiding light", it should be possible to identify a number of primitive characteristics of the various embodiments of form and structure (for instance, simply notions of maximality, irredundancy, etc.) and to build similarly unified discovery tools around them. Again, the forms may be organized as linear strings (say, as in the genome), graphs (say, as in the interactome), or even just conglomerates (say, as in the transcriptome). And the fact that even the rate of data accumulation increases continuously becomes rather a blessing in this context than a curse. It is therefore a worthwhile effort to try and identify these primitives. This seminar was intended to focus on combinatorial and algorithmic techniques of structure discovery relating to biological data that are at the core of understanding a coherent body of such data, small or large. The goal of the seminar was twofold: on one hand to identify concise characterizations of biological structure that span across multiple domains; on the other to develop combinatorial insight and algorithmic techniques to effectively unearth structure from data.

The seminar began with a town-hall, round-table style meeting where each participant shared with the others a glimpse of their work and questions that they were most excited about. This formed the basis of the program that was drawn up democratically. As the days progressed, the program evolved organically to make an optimal fit of lectures to the interest of the participants.

The first session was on population genomics, covered by Shuhua Xu and Laxmi Parida. The second was on methods on genomic sequences, covered by Rahul Siddharthan and Jonas Almeida. The next talks were on clinical medicine: an interesting perspective from a practicing physician, Walter Schubert, on treatment of chronic diseases, and Yupeng Cun spoke about prognostic biomarker discovery. Algorithms and problems in strings or genomic sequences were covered in an after-dinner session on Monday and in two sessions on Tuesday morning and late afternoon. The speakers were Sven Rahmann, Burkhard Morgenstern, Eduardo Corel, Fabio Cunial, Gilles Didier, Tobias Marschall, Matthias Gallé, Susana Vinga and Gabriel Valiente. The last speaker presented a system called "Tango" on metagenomics, and in a bizarre twist concluded the session and the day with a surprise live Argentine Tango dance performance with one of the organizers of the seminar. The early afternoon session was on metabolic networks, with lectures by Jörg Ackermann, Jun Yan and Qiang Li.

The Wednesday morning session was loosely on proteomics, with lectures by Alex Pothen, Benny Chor, Axel Mosig, Alex Grossmann, and Deok-Soo Kim. Coincidentally, three lecturers of this session shared very similar first names, leading to some gaffes and some light moments at the otherwise solemn meeting.

The Thursday sessions were on phylogenies and networks, with lectures by Mareike

Fischer, Mike Steel, Katharina T. Huber, Christoph Mayer, James A. Lake, Péter L. Erdös, Stefan Gruenewald and Peter F. Stadler. James A. Lake presented an interesting shift in paradigm, based in biology, called *cooperation and competition in phylogeny.* Péter L. Erdös gave a fascinating talk on the realization of degree sequences. Yet another session on strings was covered by Matteo Comin and Funda Ergun on Thursday. The day concluded with a lecture by Andreas Dress on pandemic modeling.

There were a few after-dinner sessions on big data, thanks to Jonas Almeida. An eclectic set of lectures were given on the last session on Friday, by Raffaele Giancarlo on clustering and by Concettina Guerra on network motifs. The meeting concluded with a fascinating lecture by Matthias Löwe on the combinatorics of graph sceneries. The impact of this on biology may not be immediately clear, but such is the intent of these far-reaching, outward-looking seminars.

2 Table of Contents

Executive Summary

Overview of Talks

3 Overview of Talks

3.1 Reduction techniques for network validation in systems biology

Jörg Ackermann (Universität Frankfurt am Main, DE)

The rapidly increasing amount of experimental biological data enables the development of large and complex, often genome-scale models of molecular systems. The simulation and analysis of these computer models of metabolism, signal transduction, and gene regulation are standard applications in systems biology, but size and complexity of the networks limit the feasibility of many methods. Reduction of networks provides a hierarchical view of complex networks, and gives insight knowledge into their coarse-grained structural properties. Although network reduction has been extensively studied in computer science, adaptation and exploration of these concepts are still lacking for the analysis of biochemical reaction systems. Using the Petri net formalism, we describe two local network structures, *common transition pairs* and *minimal transition invariants.* We apply these two structural elements for steps of network reduction. The reduction preserves the CTI-property (*covered by transition invariants*), which is an important feature for completeness of biological models. We demonstrate this concept for a selection of metabolic networks, including a benchmark network of *Saccharomyces cerevisiae* whose straightforward treatment is not yet feasible even on modern supercomputers.

3.2 Fractal decomposition of sequence representation for socializable genomics

Jonas Almeida (University of Alabama, Birmingham, US)

Joint work of Almeida, Jonas; Vinga, Susana
Main reference Almeida, J.S.; Grüneberg, A.; Maass, W.; Vinga, S. (2012). Fractal MapReduce decomposition of sequence alignment. Algorithms for Molecular Biology 7:12.

Universal Sequence Maps provide a generic numerical data structure to represent biological sequences. Recent work decomposing both the representation and the comparison of sequences raises the prospect of highly portable descriptions of human genomes. In this presentation we explore the analytical features of a participated route for computational genomics that uses the web's read-write feature of social networking infrastructure.

3.3 Metazoan conservation profiles reveal species-dependent functional enrichment patterns

Benny Chor (Tel Aviv University, IL)

Joint work of Witztum, Jonathan; Persi, Erez; Horn, David; Pasmanik-Chor, Metsada; Chor, Benny

The availability of a large number of annotated proteomes enables the systematic study of the relationships between protein conservation and functionality. In this work, we explore this question based solely on the presence (or absence) of protein homologues – the so called *conservation profile*. We study the proteomes of 18 metazoans: 11 vertebrates (including 7 mammals) and 7 invertebrates, and examine them from two distinct points of view: the human's (*Homo sapiens*) and the fly's (*Drosophila melanogaster*).

Two relevant protein groups in this context are the "universal proteins" – human/fly proteins having homologues in all 17 other species – and the "orphan proteins" – those with no homologues. But there are many additional complex patterns of conservation profiles (e.g. proteins having homologues in all vertebrates, but no invertebrate homologue), which are also of interest. In order to characterize the relations between such patterns and proteins functionality, and compare the two viewpoints, we employ Quantum Clustering (QC) and the Gorilla gene ontology tools.

We show many common enriched GO terms in universal proteins of human and fly, and lack thereof for non-universal proteins. Working solely with conservation profiles patterns, and clustering them with QC, we uncover interesting functional enrichments for resulting groups of proteins.

3.4 Whole-genome phylogeny based on non-overlapping patterns

Matteo Comin (University of Padova, IT)

With the progress of modern sequencing technologies, a number of complete genomes are now available. Traditional alignment tools cannot handle this massive amount of data, therefore the comparison of complete genomes can be carried out only with ad hoc, alignment-free methods.

In this talk we propose a distance function based on subword compositions called *Underlying Approach* (UA). We prove that the matching statistics, a popular concept in stringology that captures the statistics of common words between two strings, can be derived from a small set of "independent" subwords, namely the irredundant common subwords. We further refine this statistics by avoiding to count the same subword multiple times. This filter removes the overlaps, by discarding the subwords that occur in regions covered by other more significant subwords.

The UA builds a scoring function based on this set of patterns, called *underlying*. We prove that this set is by construction linear in the size of input, without overlaps, and it can be computed efficiently. Results show the validity of our method in the reconstruction of phylogenetic trees. The Underlying Approach outperforms the current state of the art methods. Moreover, we show that the accuracy of UA is achieved with a very small number of subwords, that in some cases carry meaningful biological information.

References

1 Sims, G.E.; Jun, S.-R.; Wu, G.A.; Kim, S.-H. (2009). Whole-genome phylogeny of mammals: evolutionary information in genic and nongenic regions. PNAS, 2009, 106(40): 17077–82.

2 Ulitsky, I.; Burstein, D.; Tuller, T.; Chor, B. (2006). The average common substring approach to phylogenomic reconstruction. J. Comput. Biol. 13(2): 336–50.

3 Comin, M.; Verzotto, D. (2012). Comparing, ranking and filtering motifs with character classes: application to biological sequences analysis. In "Biological knowledge discovery handbook: preprocessing, mining and postprocessing of biological data", M. Elloumi, A.Y. Zomaya (Eds.). Wiley.

4 Comin, M.; Verzotto, D. (2012). Whole-genome phylogeny by virtue of unic subwords. Proceedings of BIOKDD 2012, pp 190-195.

3.5 Rainbow graphs, alignments and motifs

Eduardo Corel (University of Evry, FR)

We present a graph-based approach to tackle the problem of integrating partial sequence similarity data into a multiple sequence alignment. The problem of finding shared similarities among the sequences is formulated as the clustering of a vertex-coloured graph (the incidence graph of the set of similarities) into *colourful* or *rainbow* components, where every colour appears at most once. We further present several combinations of algorithms to solve the NP-hard problem of finding colourful components by minimum edge-deletions. We show that including matching protein domains, or RNA secondary structure predictions, leads to improved multiple sequence alignments.

3.6 Faster variance computation for patterns with gaps

Fabio Cunial (Georgia Institute of Technology, US)

Main reference F. Cunial, "Faster variance computation for patterns with gaps," in Proc. of the Mediterranean Conference on Algorithms (MEDALG), 2012.

Determining whether a pattern is statistically overrepresented or underrepresented in a string is a fundamental primitive in computational biology and in large-scale text mining. We study ways to speed up the computation of the expectation and variance of the number of occurrences of a pattern with rigid gaps in a random string. Our contributions are twofold: first, we focus on patterns in which groups of characters from an alphabet Σ can occur at each position. We describe a way to compute the exact expectation and variance of the number of occurrences of a pattern w in a random string generated by a Markov chain in $O(|w|^2)$ time, improving a previous result that required $O(2^{|w|})$ time. We then consider the problem of computing expectation and variance of the *motifs* of a string s in an IID text. Motifs are rigid gapped patterns that occur at least twice in s, and in which at most one character from Σ occurs at each position. We study the case in which s is given offline, and an arbitrary motif w of s is queried online. We relate computational complexity to the structure of w and s, identifying sets of motifs that are amenable to $o(|w| \log |w|)$ time online computation after $O(|s|^3)$ preprocessing of s. Our algorithms lend themselves to efficient implementations.

3.7 Integrating prior knowledge into prognostic biomarker discovery

Yupeng Cun (Universität Bonn, DE)

Joint work of Cun, Yupeng; Frohlich, Holger

Stratification of patients according to their clinical prognosis is a desirable goal in cancer treatment in order to achieve a better personalized medicine. Reliable predictions on the basis of gene signatures could support medical doctors on selecting the right therapeutic strategy. However, during the last years, the low reproducibility of many published gene signatures has been criticized. It has been suggested that incorporation of network or pathway information into prognostic biomarker discovery could improve prediction performance. In the meanwhile, a large number of different approaches have been suggested for the same purpose.

First, we compared 14 published classification approaches (8 using network information) on six public breast cancer datasets with respect to prediction accuracy and gene selection stability [1]. A gene set enrichment analysis for the predictive biomarker signatures by each of these methods was done to show the association with disease related genes, pathways and known drug targets. We found that, on average, incorporation of pathway information or protein interaction data did not significantly enhance prediction performance, but increased greatly the interpretability of gene signatures. The results indicated that no single algorithm performs best with respect to all three categories in our study. Incorporating network of prior knowledge into gene selection methods in general did not significantly improve classification accuracy, but greatly increased the interpretability of gene signatures compared to classical algorithms.

Second, we present our newly developed algorithm, called fNet, which integrates protein-protein interaction network information into gene selection for prognostic biomarker discovery [2]. Our method is a simple filter-based approach, which focuses on central genes with large differences in their expression. Compared to several other competing methods, our algorithm reveals a significantly better prediction performance and higher signature stability. Moreover, obtained gene lists are highly enriched with known disease genes and drug targets. We extended our approach further by integrating information on candidate disease genes and targets of disease associated transcription factors. The first can additionally increase the association of gene lists to biological knowledge.

References

1 Cun, Yupeng; Fröhlich, Holger (2012). Prognostic gene signatures for patient stratification in breast cancer: accuracy, stability and interpretability of gene selection approaches using prior knowledge on protein-protein interactions. BMC Bioinformatics, 13:69 doi:10.1186/1471-2105-13-69.

2 Cun, Yupeng; Abnaof, Khalid; Fröhlich, Holger (2012). Integrating prior knowledge into prognostic biomarker discovery. Submitted.

3.8 Variable length decoding II

Gilles Didier (CNRS, Marseille, FR)

Let us consider a prefix code P (i.e. a set of words in which no word is prefix of another) in which each words is associated to a unique identifier. The classic way of coding with such a code is to transform a sequence over the alphabet of idents into a sequence over the prefix code alphabet by replacing each ident by its counterpart in the code (the prefix property making the reverse operation unambiguous). This is not what we are doing here: we use P to code sequences over the prefix code alphabet into a sequence of idents. The coding of a sequence s is the sequence t in which the ident at position i is the one corresponding to the (unique, thanks to the prefix property) words of the prefix code which occurs at the position i of s (we assume that such a word always exists). We have proved that this coding can be somehow reversed. Our result states that, being given the coding of sequence s by a prefix code P, there exists a sequence s' and a prefix code P' such that:

1. the coding of s' by P' is equal to that of s by P;
2. the words of P' and P have the same length;
3. if a pair (s'', P'') satisfies the two preceding assertions, then s'' can be obtained from s' by a letter-to-letter application.

The sequence s' is what we called the (variable length) local decoding of s.

From the third item above, the alphabet of the local decoding of s extends that of s (it has a greater diversity of symbols). Another, and most widely used in DNA sequence analysis, way of augmenting the alphabet of sequences is to consider k-mers (conversely, the sequence of k-mer of s can be seen as the coding – not the decoding – of s by a prefix code of constant length k). It turns out that the local decoding can be used in place of k- mers and that it somehow shows better properties.

In this talk, we first present variable length decoding, some of the ideas behind it, and a linear algorithm which computes it. We next apply the decoding to two questions where k-mers approaches are widely used: sequence comparison and sequence assembly. We show that local decoding approaches have good results with respect to k-mers, and try to explain why.

In both cases, a critical point in applying our method is the selection of a prefix code relevant with regard to the question. This is done in the following, heuristic way. We first define a score over a decoding which somehow predicts its relevance. Next, we consider a family of prefix codes parameterized by some quantity (something like a depth) and pick up the prefix code among this family which leads to the local decoding with the best score. Naturally, a better way would be to select the prefix code with the best code among the whole set of prefix codes. Because of some (or rather, a lack of) properties of the set of prefix codes and the associated local decodings, this problem is hard to solve in a feasible time.

3.9 Pandemics and the dynamics of quasispecies evolution: facts, models, and speculations

Andreas Dress (Shanghai Institutes for Biological Sciences, CN)

In my lecture, I argue that viral quasispecies dynamics offers the possibility of a "natural vaccination program", due to different degrees of virus virulence in highly heterogeneous viral populations which might explain why, so far, all pandemics have eventually petered out.

3.10 Graphical degree sequences and realizations

Péter L. Erdös (Hungarian Academy of Sciences, HU)

Joint work of Erdös, Péter L.; Kiraly, Zoltan; Miklos, Istvan
Main reference P.L. Erdös, Z. Kiraly, I. Miklos, "On the swap-distances of different realizations of a graphical degree sequence," arXiv:1205.2842v2 [math.CO], 2012.
URL http://arxiv.org/abs/1205.2842v2

One of the first graph theoretical problems which got serious attention (already in the fifties of the last century) was to decide whether a given integer sequence is equal to the degree sequence of a simple graph. One method to solve this problem is the greedy algorithm of Havel and Hakimi, which is based on the *swap* operation. Another, closely related question is to find a sequence of swap operations to transform one graphical realization into another one of the same degree sequence. This latter problem got particular emphasis in connection of fast mixing Markov chain approaches to sample uniformly all possible realizations of a given degree sequence.

Earlier there were only crude upper bounds on the shortest possible length of such swap sequences between two realizations. In this lecture we present formulae for these *swap-distance*s of any two realizations of simple undirected or directed degree sequences. The exact values in those formulae seem to be not computable efficiently. However the formulae provide sharp upper bounds on swap-distances.

3.11 Periodicity in data streams

Ayse Funda Ergun (Simon Fraser University, Burnaby, CA)

As our data sets grow in size, the need for techniques for processing them under limited resources becomes more critical. One model for processing large data sequences is that of *streaming computation*: the input is read sequentially, i.e., "streamed" in one long pass, and the computation is performed while using small (typically logarithmic in the size of the input) memory.

Streaming techniques are well studied in terms of statistical properties such as the frequency of elements in a stream, but not nearly as much in terms of the particular ordering of the input elements. In this talk, we discuss techniques for analyzing order-related trends of a stream, in particular, its self-similar properties. In this context, we show how to find the period of a stream by using polylogarithmic space if the stream is periodic. Surprisingly, we also show a linear space lower bound using communication theory techniques, i.e. we show that it takes linear space to show that a stream is aperiodic (has a very long period). We also show that one can approximate the distance to periodicity in small space.

3.12 Phylogenetically decisive taxon coverage

Mareike Fischer (Universität Greifswald, DE)

In a recent study, Sanderson and Steel defined and characterized *phylogenetically decisive sets* of taxon sets. A set is called phylogenetically decisive if, regardless of the trees chosen for each of its taxon sets, as long as these trees are compatible with one another, their supertree is always unique. It remained unclear whether deciding if a set of taxon sets is phylogenetically decisive can always be made in polynomial time or not. This question was one of the "Penny Ante" prize questions of the Annual New Zealand Phylogenetics Meeting 2012. In my talk, I explain phylogenetic decisiveness and demonstrate a new characterization for it, which then leads to a polynomial time algorithm both for the (simpler) rooted case as well as for the (more complicated) unrooted case.

3.13 On largest-maximal repeats

Matthias Gallé (Xerox Research Center Europe, Grenoble, FR)

Main reference M. Gallé, "Searching for compact hierarchical structures in DNA by means of the Smallest Grammar Problem," Ph.D. Thesis, Université Rennes 1, 2011.
URL http://tel.archives-ouvertes.fr/tel-00595494/en/

Largest-maximal repeats (also known as *near-supermaximal repeats*) are a class of exact repeats. Stricter than maximal-repeats, they are less redundant and form a basis of all repeats (mirroring the definitions of irredundant/tiling motifs for rigid motifs). In this short talk I present what is known of them and argue that they could be more interesting for several applications that use maximal repeats.

3.14 A unifying framework for stability-based class discovery in microarray data

Raffaele Giancarlo (Università di Palermo, IT)

Joint work of Giancarlo, Raffaele; Utro, Filippo
Main reference R. Giancarlo, F. Utro, "Algorithmic paradigms for stability-based cluster validity and model selection – Statistical methods, with applications to microarray data analysis," Theoretical Computer Science, Vol. 428, pp. 58–79, 2012.
URL http://dx.doi.org/10.1016/j.tcs.2012.01.024

Clustering is one of the most well known activities in scientific investigation in many disciplines [1, 2]. In this beautiful area, one of the most difficult challenges is the *model selection* problem, i.e., the identification of the correct number of clusters in a dataset [4, 5, 6]. Among the few novel techniques for model selection proposed in the last decade, the stability-based methods are the most robust and best performing in terms of prediction, but the slowest in terms of time [3]. Unfortunately, such a fascinating area of statistics as model selection, with important practical applications, has received very little attention in terms of algorithmic design and engineering. Therefore, in order to partially fill this gap, we highlight: (a) the first general algorithmic paradigm for stability-based methods for model selection; (b) a novel algorithmic paradigm for the class of stability-based methods for cluster validity, i.e., methods assessing how statistically significant is a given clustering solution; (c) the main idea behind a paradigm that describes a very efficient speed-up for stability-based model selection methods.

References

1 Andreopoulos, B.; An, A.; Wang, X.; Schroeder, M. (2009). A roadmap of clustering algorithms: finding a match for a biomedical application. Briefings in Bioinformatics 10(3), 297–314.

2 D'haeseleer, P. (2006). How does gene expression cluster work? Nature Biotechnology 23, 1499–1501.

3 Giancarlo, R.; Scaturro, D.; Utro, F. (2008). Computational cluster validation for microarray data analysis: experimental assessment of Clest, Consensus Clustering, Figure of Merit, Gap Statistics and Model Explorer. BMC Bioinformatics 9, 462.

4 Giancarlo, R.; Scaturro, D.; Utro, F. (2008). A tutorial on computational cluster analysis with applications to pattern discovery in microarray data. Mathematics in Computer Science 1, 655–672.

5 Giancarlo, R.; Scaturro, D.; Utro, F. (2009). Statistical indices for computational and data driven class discovery in microarray data. In: Chen, J.Y., Lonardi, S. (eds.) Biological Data Mining, pp. 295–335. CRC Press, San Francisco, USA.

6 Handl, J.; Knowles, J.; Kell, D. (2005). Computational cluster validation in Post-genomic data analysis. Bioinformatics 21(15), 3201–3212.

3.15 Comparing geometries of chains

Alex Grossmann (Laboratoire Statistique & Génome, Evry, FR)

Joint work of Grossmann, Alex; Lesieur, Claire; Feverati, Giovanni; Vuillon, Laurent

A chain is an ordered set of points in three-dimensional space. Let $N \geq 4$ be the number of points in a chain. The chain contains $N - 1$ pairs of consecutive points. Consider the distances between them. The chain contains $N - 2$ triplets of consecutive points. Consider the areas of the corresponding triangles. The chain contains $N - 3$ quadruplets of consecutive points. Consider the volumes of the corresponding tetrahedrons. We have so introduced $3N - 6$ numbers that are manifestly independent of the choice of the reference frame used to define the coordinates. So we have eliminated the six parameters of global translations and rotations, and are dealing with the intrinsic geometry of the chain. The talk illustrates the procedure with elementary examples, and by data from the Protein Data Bank. We also discuss the case where there are several chains in the same reference frame. The first step is the examination of of histograms of the data. This allows a partial decoupling of the geometry from the primary structure. The main problem is then understanding the histograms, which of course requires non-geometrical inputs.

3.16 On the quartet distance between phylogenetic trees

Stefan Gruenewald (Shanghai Institutes for Biological Sciences, CN)

Joint work of Gruenewald, Stefan; Huang, Libo

The quartet distance is one way to quantify how different two phylogenetic trees on the same taxa set are. It is defined to be the number of subsets of cardinality 4 of the taxa set for which the restrictions of the trees are different. Bandelt and Dress showed in 1986 that the maximum distance between two binary trees, when normalized by the number of all 4-sets, is monotone decreasing with n. They conjectured that the limit of this ratio is 2/3 (the 2/3-conjecture). In order to prove this conjecture, it seems to be helpful to look at a generalization for not necessarily binary trees. This allows us to compare trees with few splits (i.e. few interior edges) but many taxa.

A quartet is a split of a 4-set into to pairs. The quartet that splits a set $\{a, b, c, d\}$ into the pairs $\{a, b\}$ and $\{c, d\}$ is denoted by $ab|cd$. A phylogenetic tree whose taxa set contains $\{a, b, c, d\}$ displays the quartet $ab|cd$ if the paths between a and b and between c and d are vertex-disjoint. For two phylogenetic trees T, T' with identical taxa set of cardinality n, let $q(T, T')$ be the number of 4-sets for which each of T and T' displays one of the three possible quartets. Let $s(T, T')$ be the number of 4-sets for which both of T and T' display the same quartet. We conjecture that $s(T, T') \geq 1/3q(T, T') - o(n^4)$, and I give the easy proof for the case where both trees have only one interior edge. Clearly, the new conjecture implies the 2/3-conjecture, and it turns out that the converse is also true.

3.17 Conservation of complexes in protein-protein interaction networks

Concettina Guerra (Georgia Institute of Technology, US)

Joint work of Guerra, Concettina; Ciriello, Giovanni, Mina, Marco; Guzzi, Pietro Hiram; Cannataro, Mario
Main reference G. Ciriello, M. Mina, P.H. Guzzi, M. Cannataro, C. Guerra, "AlignNemo: A local network alignment method to integrate homology and topology," PLoS ONE 7(6):e38107, 2012.
URL http://dx.doi.org/10.1371/journal.pone.0038107

Comparative analysis of protein-protein interaction networks of different species is an important approach to understanding the mechanisms used by living organisms. One of the computational goals of network comparison (or alignment) is revealing sub-networks, or protein complexes, that are conserved throughout evolution.

In this talk, I analyze the behavior of algorithms for the alignment of protein-protein interaction networks, with respect to the architecture of protein complexes. Protein complexes in PPI networks of certain organisms, such as yeast, are thought to have a modular organization. For instance, according to one model proposed in the literature, complexes consist of core components and attachments. The core is defined as a small group of proteins that are functionally similar and have highly correlated transcriptional profiles. The core is surrounded by less strongly connected proteins, defined *attachments*, which allow diversification of potential functions.

I present the recently developed algorithm AlignNemo that identifies conserved complexes in a pair of PPI networks. The discovered conserved sub-networks have a general topology and need not correspond to specific interaction patterns, so that they more closely fit the models of functional complexes proposed in the literature. Based on reference datasets of protein complexes, AlignNemo shows better performance than other methods in terms of both precision and recall. The obtained solutions are biologically sound according to the semantic similarity measure as applied to Gene Ontology vocabularies.

3.18 Recognizing treelike k-dissimilarities

Katharina T. Huber (Univ. of East Anglia, Norwich, GB)

Joint work of Herrmann, Sven; Huber, Katharina T.; Moulton, Vincent; Spillner, Andreas
Main reference Herrmann, Sven; Huber, Katharina T.; Moulton, Vincent; Spillner, Andreas (2012). Recognizing treelike k-dissimilarities. Journal of Classification, (DOI: 10.1007/s00357-012-9115-2).
URL http://arxiv.org/abs/1206.1374

Many methods for constructing phylogenetic trees from distances essentially work by projecting an arbitrary pairwise dissimilarity onto some "nearby" tree metric. Even so, it is well-known that such methods can suffer from the fact that pairwise distance estimates involve some loss of information. As a potential solution to this problem, Pachter et al proposed using k-wise distance estimates, $k > 2$, to reconstruct trees. Their rationale was that k-wise estimates (as opposed to 2-wise estimates) are potentially more accurate since they can capture more information than pairwise distances, a point that was also made by Felsenstein.

In this talk we focus on the following question, which was originally posed by Pachter et al.: given an arbitrary k-dissimilarity, $k > 2$, how do we test whether this map comes from a tree?

3.19 Understanding molecular geometries via the beta-complexes

Deok-Soo Kim (Hanyang University, Seoul, KR)

Joint work of Kim, Deok-Soo; Cho, Youngsong; Sugihara, Kokichi; Ryu, Joonghyun; Kim, Donguk
Main reference D.-S. Kim, Y. Cho, K. Sugihara, J. Ryu, D. Kim, "Three-dimensional beta-shapes and beta-complexes via quasi-triangulation," Computer-Aided Design, Vol. 42, Issue 10, pp. 911–929, 2010.
URL http://dx.doi.org/10.1016/j.cad.2010.06.004

It has been generally agreed that structure is important in understanding biomolecular functions. Among others, geometry is one of the most important aspects of molecular structure. Despite of its importance, the theory for molecular geometry has not been sufficiently investigated. In this presentation, we present a unified geometric theory, the beta-complex theory, for biomolecules, and demonstrate how this theory can be used for solving important molecular structure problems.

The beta-complex is a generalization of the alpha-complex, which is a structure derived from the Voronoi diagram. For point sets, the Voronoi/Delaunay structures are useful for understanding the spatial structure of the point sets. Being a powerful computational tool, the generalization of the Voronoi/Delaunay structures has been made in various directions, including the Voronoi diagram of spherical balls (or also sometimes called spherical atoms). The Voronoi diagram of spherical atoms nicely defines the proximity among the atoms and its dual structure, called the quasi-triangulation, conveniently represents the topology structure of the Voronoi diagram.

This talk introduces the Voronoi diagram of spherical atoms and the quasi- triangulation in the three-dimensional space. Based on the quasi-triangulation, we define a new geometric structure called the *beta-complex* that concisely represents the proximity among all atoms. It turns out that the beta-complex can be used to precisely, efficiently, and easily solve many seemingly unrelated important geometry and topology problems for the atom set within a single framework. Among many potential application areas, structural molecular biology is the most immediate.

Application examples include the following: the most efficient/precise computation of van der Waals volume (and area), the volumes within an accessible/Connolly surface; an efficient docking simulation; the recognition of internal voids and their volume computation; the recognition of molecular tunnels; the comparison (or superposition) of the boundary structures of two proteins; shape reasoning such as measuring the sphericity of protein; the efficient computation of the optimal side-chain placement, etc. We anticipate that more important applications will be discovered. Several pieces of application software based on the Voronoi diagram and the beta-complex are freely available at the Voronoi Diagram Research Center (VDRC, `http://voronoi.hanyang.ac.kr`).

References

1 Kim, Deok-Soo; Cho, Youngsong; Sugihara, Kokichi; Ryu, Joonghyun; Kim, Donguk (2010). Three-dimensional beta-shapes and beta-complexes via quasi- triangulation. Computer-Aided Design, Vol. 42, Issue 10, pp. 911-929.

2 Kim, Deok-Soo; Cho, Youngsong; Sugihara, Kokichi (2010). Quasi-worlds and quasi- operators on quasi-triangulations. Computer-Aided Design, Vol. 42, Issue 10, pp. 874-888.

3 Cho, Youngsong; Kim, Jae-Kwan; Ryu, Joonghyun; Won, Chung-In; Kim, Chong-Min; Kim, Donguk; Kim, Deok-Soo (2012). BetaMol: a molecular modeling, analysis and visualization software based on the beta-complex and the quasi-triangulation. Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol.6, Issue 3, pp. 389-403.

3.20 Using genomes to track the evolution of life on Earth

James A. Lake (Univ. of California, Los Angeles, US)

Today evolutionary genomics is in a state of crisis because we mistakenly assumed that once complete genome became available, the complete tree of life on Earth could be easily reconstructed in considerable detail. Instead, all of us in the field agree that we cannot easily determine a single tree. Different genes have different histories. However, everyone seems to have different reasons for why they think that this happens. Here, I make the case that Darwinian tree-like evolution, and the "survival of the fittest" metaphor, give an incomplete view of evolution, and that we need to focus more upon both tree-like evolution and cooperation between organisms (endosymbioses and other types of gene sharing). Trees are easy to calculate from genomic data, but we must combine "survival of the fittest" and "cooperation", if we are to reconstruct the evolution of life on Earth. Methods to do this are vastly more complex and are just being developed. I describe some of the remarkable findings that are now being obtained using these new methods.

3.21 Systematic identification of novel gene members of mammalian metabolic pathways

Qiang Li (Shanghai Institutes for Biological Sciences, CN)

The metabolic functions of known enzymes in the metabolic pathways are among the best studied gene functions so far. However, how these enzymes are regulated and how they are linked to other metabolism-related genes such as metabolite transporters is still unclear. Using the fact that functionally related genes are often co-expressed, we develop an efficient computational method to predict novel genes participating in known metabolic pathways by screening genome-wide expression data. We identify the sets of enzymes associated with consecutive metabolic reactions that also show co-expression. Using these co-expressed consecutive enzymes as query sets or baits, we screen the entire mouse microarray datasets in the Gene Expression Omnibus (GEO) database for additional co-expressed genes. Using this method, we also gain insights into the physiological conditions that affect metabolic pathways. Our extended list of co-expressed metabolism-related genes facilitates the identification of their potential regulators using promoter analysis. We further validate that these novel genes also show spatial co-localizations with known enzymes in metabolic pathways by high-resolution in situ hybridization (ISH) data in E14.5 mouse embryos. Our prediction provides novel gene candidates with putative functional roles in metabolic pathways, which will be further investigated and validated by experiments.

3.22 Reconstruction of random scenery

Matthias Loewe (Universität Münster, DE)

A random scenery is a random coloring of the integer lattice Z^d. Usually this coloring is chosen to be IID. The problem of scenery reconstruction starts with a random scenery that cannot be directly observed. The only thing that can be seen are the observations along the path of one (infinite) realization of a random walk. The question now is: can one reconstruct the scenery when only given the observations? We survey results that answer this question in the affirmative (up to shift of the origin and rotation/reflection) under certain technical assumptions.

3.23 Speeding up exact motif discovery by bounding the expected clump size

Tobias Marschall (CWI, Amsterdam, NL)

Joint work of Marschall, Tobias; Rahmann, Sven
Main reference T. Marschall, S. Rahmann, "Speeding up exact motif discovery by bounding the expected clump size," in Proc. of 10th Workshop on Algorithms in Bioinformatics (WABI), pp. 337–349, 2010.
URL http://dx.doi.org/10.1007/978-3-642-15294-8_28

The overlapping structure of complex patterns, such as IUPAC motifs, significantly affects their statistical properties and should be taken into account in motif discovery algorithms. The contribution of this talk is twofold. On the one hand, we give surprisingly simple formulas for the expected size and weight of motif clumps (maximal overlapping sets of motif matches in a text). In contrast to previous results, we show that these expected values can be computed without matrix inversions. On the other hand, we show how these results can be algorithmically exploited to improve an exact motif discovery algorithm. First, the algorithm can be efficiently generalized to arbitrary finite-memory text models, whereas it was previously limited to IID texts. Second, we achieve a speed-up of up to a factor of 135. Our open-source (GPL) implementation is available at `http://mosdi.googlecode.com`.

3.24 Biases in phylogenetic reconstruction

Christoph Mayer (ZFMK, Bonn, DE)

Joint work of Mayer, Christoph; Wägele, Wolfgang; Kück, Patrick; Meid, Sandra; Richter, Stefan

We define a phylogenetic bias as every effect, which influenced the observable site patterns of a molecular data set, such that they cannot be explained by an evolution governed by a single stationary, homogeneous time-reversible Markov process with site rate heterogeneity constrained to invariant sites and gamma distributed rates. For nucleotide data sets this implies a deviation from an evolution governed by a single, time independent GTR+I+G substitution model. Such a bias can severely impede the reconstruction success of all model based tree reconstruction methods. If time reversibility is broken, a bias can be interpreted as a plesiomorphy effect, which is demonstrated using evolution scenarios based on time-dependent base frequencies as well as time-dependent heterogeneous site rates. Our results highlight the vulnerability of model-based tree reconstruction methods under realistic evolutionary scenarios.

3.25 Using heterogeneous sources of information for multiple sequence alignment

Burkhard Morgenstern (Universität Göttingen, DE)

Traditional methods for multiple sequence alignment are based on primary sequence information alone. Numerous algorithmic approaches have been proposed to calculate (near-)optimal alignments in the sense of some objective function defined in terms of sequence similarity. However, many more sources of information are nowadays available to find homologies between nucleic acid or protein sequences.

We use a recently published graph-theoretical approach to multiple protein alignment to combine primary-sequence similarity and other information, such as similarities to known protein domains, in order to obtain improved multiple protein alignments.

3.26 Co-localization and co-segmentation: algorithms and applications in image analysis

Axel Mosig (Ruhr-Universität Bochum, DE)

This talk introduces co-localization as a concept that occurs naturally in the analysis of bioimage data, either in the analysis of multi-label fluorescence microscopy or the segmentation of spectral images obtained from Raman or CARS microspectroscopes. As a universal tool for studying co-localization, the concept of co-segmentation, i.e. the simultaneous segmentation of two related images at the same time, is introduced. We propose algorithms that allow to compute co-segmentations between hierarchical images of different types. A main result are algorithms that allow to compute co-segmentations between fluorescence and spectral image, which has important applications in the annotation and registration of spectral images.

3.27 Combinatorial algorithms for flow cytometry

Alex Pothen (Purdue University, US)

Flow cytometry is a nearly 50-year old technology for studying properties of single cells via scattering and fluorescence induced by lasers, with applications in immunology and diagnosis of diseases. In recent years, flow ctyometry has become multispectral (thirty or more signals can be detected simultaneously), and high-throughput (millions of cells can be analyzed per minute at the single cell level). However, for analyzing the high dimensional, large-scale data generated by the new experimental methodologies, new algorithms from many areas of computer science, mathematics, and statistics are needed. We describe a few of the computational problems in this context.

We also describe FlowMatch, an algorithm for registering different cell types from patient samples using matchings in graphs and hierarchical template construction algorithms from multiple sequence alignment. These cell types are then followed across multiple time points and experimental conditions. We report results from flow cytometry data generated from leukemia, multiple sclerosis, and phiosphorylation shifts in T-cells. High throughput, multispectral flow cytometry coupled with new algorithmic advances enable systems biology discoveries at the single-cell level, leading to personalized medicine and new approaches to drug discovery.

3.28 The reverse complementarity relation is more complex than we thought

Sven Rahmann (Universität Duisburg-Essen, DE)

Joint work of D'Addario, Marianna; Kriege, Nils; Rahmann, Sven
Main reference M. D'Addario, N. Kriege, S. Rahmann, "Designing q-unique DNA sequences with integer linear programs and Euler tours in De Bruijn graphs," in Proc. of German Conference on Bioinformatics 2012. OpenAccess Series in Informatics (OASIcs), vol 26, pp. 82–92, 2012.
URL http://dx.doi.org/10.4230/OASIcs.GCB.2012.82

DNA nanoarchitectures require carefully designed oligonucleotides with certain non-hybridization guarantees, which can be formalized as the q-uniqueness property on the sequence level. We study the optimization problem of finding a longest q-unique DNA sequence. We first present a convenient formulation as an integer linear program on the underlying De Bruijn graph, that allows to flexibly incorporate a variety of constraints; solution times for practically relevant values of q are short. We then provide additional insights into the problem structure using the quotient graph of the De Bruijn graph with respect to the equivalence relation of reverse complementarity. Specifically, for odd q the quotient graph is Eulerian, and finding a longest q-unique sequence is equivalent to finding an Euler tour, hence solved in linear time (with respect to the output string length). For even q, self-complementary edges complicate the problem, and the graph has to be Eulerized by deleting a minimum number of edges. Two sub-cases arise, for one of which we present a complete solution, while the other one remains open.

3.29 Treatment of chronic diseases: is there a logic of failure?

Walter Schubert (Universität Magdeburg, DE)

The steadily increasing inefficiency in treating chronic diseases – in spite of elegant and logic molecular biological studies and helpful applied mathematics – has recently evoked urgent warnings in a report of the World Health Organization (WHO) 2011. The overall question asked by seriously worried scientists and editors of highly ranked journals is "Where are we going wrong?" (e.g. Nat. Rev. Clin. Oncol. 8, 189-190, 2011), whilst the disillusioned pharmaceutical industry closes discovery facilities and dismisses thousands of employees. Surprisingly, corresponding early warnings in a report entitled "The fruits of genomics" (Lehman Brothers and Mc Kinsey, 2001) were totally disavowed by the scientific community, whilst further promoting the likely cause of the problem, commonly known as large-scale expression profiling. This presentation provides insight into a logic of failure caused by a low content trap, which is a blind spot, but can have fatal impact on a patient's survival. Can mathematics help?

3.30 Evolution of eukaryotic centromeres, and the importance of correct sequence alignment

Rahul Siddharthan (The Institute of Mathematical Sciences, Chennai, IN)

Joint work of Siddharthan, Rahul; Jayaraman, Gayathri; Chatterjee, Gautam; Thattikota, Yogitha; Padmanabhan, Sreedevi; Jain, D; Raghavan, D.K.; Sanyal, Kaustuv

I discuss the alignment of non-coding DNA sequence, and show that the majority of alignment programs, written with proteins in mind, fare very poorly on non-coding DNA. I discuss an alternative approach using an evolutionary model, implemented in the program Sigma-2. I then discuss some recent work on the evolution of centromeres in Hemisacomycetous yeasts of the Candida clade, in collaboration with K. Sanyal (Bangalore). While the biology is very interesting, it also illustrates the importance of a careful approach to alignment.

References

1 Jayaraman, G.; Siddharthan, R. (2010). BMC Bioinformatics 11:464.
2 Padmanabhan, S.; Thakur, J.; Siddharthan, R.; Sanyal, K. (2008). Proc. Natl. Acad. Sci. USA, 105(50):19797-802.
3 Chatterjee, G.; Thattikota, Y.; Padmanabhan, S.; Jain, D.; Raghavan, V.K.; Siddharthan, R.; Sanyal, K. (2012). Submitted.

3.31 Computing a consensus of multilabeled trees

Andreas Spillner (Universität Greifswald, DE)

In this talk we consider two challenging problems that arise in the context of computing a consensus of a collection of multilabeled trees. Forming such a consensus is part of an approach to reconstruct the evolutionary history of a set of species for which events such as genome duplication and hybridization have occurred in the past. We outline exact algorithms that have an exponential run time in the worst case, and highlight the impact of several structural properties of the input on the performance of the algorithms. We conclude discussing some open problems and directions for future work.

3.32 Orthologs, co-graphs, gene trees, and species trees

Peter F. Stadler (Universität Leipzig, DE)

Orthology detection is an important problem in comparative and evolutionary genomics. Although most practical approaches start from inferred gene and/or species trees, orthology can be estimated at acceptable levels of accuracy without any phylogenetic information based on determining sets of closest relatives. We recently characterized orthology relations mathematically as co- graphs. This is equivalent to a (in general not fully resolved) gene tree together with an event labeling that identifies each interior node as either a speciation or a gene duplication event. This characterization opens a new avenue to improving computational methods for orthology detection without incorporating phylogenetic information for this purpose. Inferred orthology relations can then be used as partial information or constraints in the reconstruction of gene trees and their associated species trees. Indeed, given an event-labeled gene tree, the assignment of genes to species defines a partially resolved species tree, and hence establishes constraints on the possible phylogenetic relationships deriving directly from the orthology assignments. In this presentation an overview of the mathematical framework and a first glimpse at computational results is be presented.

References

1 Hellmuth, M.; Hernandez-Rosales, M.; Huber, K.T.; Moulton, V.; Stadler, P.F.; Wieseke N. (2012). Orthology relations, symbolic ultrametrics, and cographs. J. Math. Biol. 2012 DOI: 10.1007/s00285-012-0525-x.

2 Hernandez-Rosales, Maribel; Hellmuth, Marc; Wieseke, Nicolas; Huber, Katharina T.; Moulton, Vincent; Stadler, Peter F. (2012). From event-labeled gene trees to species trees. RECOMB RG Niteroi 2012 (accepted).

3.33 What can probability theory tell us about life's past and future?

Mike Steel (University of Canterbury, Christchurch, NZ)

Joint work of Steel, Mike; Gascuel, Olivier; Mooers, Arne; Li, Thomas; Stadler, Tanja

In a landmark 1925 paper, George Udny Yule FRS described a Markov process for explaining the observed distribution of species into genera [6]. In this model, each species can give rise to a new species according to a constant-rate pure-birth process. Eighty-five years later this Yule process and its extensions provide a basis for studying the shape of macroevolution [1]. In this talk, I highlight some results concerning Yule-type processes in evolutionary biology. First I show that the reliable estimation of ancestral information in the distant past depends on whether or not the ratio of speciation to mutation exceeds a critical ratio [2]. For a simple symmetric mutation model, this critical ratio turns out to be 4 (or 6 depending on the estimation method). Then I study the expected distribution of times between speciation events [4, 5] and its implications for how much evolutionary heritage might be lost under simple field of bullets models of extinction [3].

References

1 Aldous, D. (1995). *Probability distributions on cladograms.* In: D. Aldous, R. Pemantle (Eds.), Random Discrete Structures, IMA Volumes in Mathematics and its Applications 76, Springer, pp. 1–18.

2 Gascuel, O.; Steel, M. (2010). *Inferring ancestral sequences in taxon-rich phylogenies.* Mathematical Biosciences, 227: 125–135.

3 Mooers, A.; Gascuel, O.; Stadler, T.; Li, H.; Steel, M. (2012). *Branch lengths on Yule trees and the expected loss of phylogenetic diversity.* Systematic Biology. 61(2): 195–203.

4 Stadler, T.; Steel, M (2012). *Distribution of branch lengths and phylogenetic diversity under homogeneous speciation models*, J. Theoretical Biology, 297, 33–40.

5 Steel, M.; Mooers, A. (2010). *Expected length of pendant and interior edges of a Yule tree.* Applied Mathematics Letters 23(11): 1315–1319.

6 Yule, G. U. (1925). *A mathematical theory of evolution. Based on the Conclusion sof Dr. J.C. Willis.* FRS Phil. Trans. Roy. Soc. 213, 21-87.

3.34 Sequence classification using reference taxonomies

Gabriel Valiente (TU of Catalonia, Barcelona, ES)

Joint work of Ribeca, Paolo; Valiente, Gabriel
Main reference P. Ribeca, G. Valiente, "Computational challenges of sequence classification in microbiomic data," Briefings in Bioinformatics 12(6):614-625, 2011.
URL http://dx.doi.org/10.1093/bib/bbr019

Next generation sequencing technologies have opened up an unprecedented opportunity for microbiology by enabling the culture-independent genetic study of complex microbial communities, which were so far largely unknown. The analysis of metagenomic data is challenging, since a sample may contain a mixture of many different microbial species, whose genome has not necessarily been sequenced beforehand. In this talk, we address the problem of analyzing metagenomic data for which databases of reference sequences are already known. We discuss both composition and alignment-based methods for the classification of sequence reads, and present recent results on the assignment of ambiguous sequence reads to microbial species at the best possible taxonomic rank.

3.35 Pattern matching through iterative function systems: bridging numerical and graph structures for biosequence analysis

Susana Vinga (Technical University, INESC-ID, Lisboa, PT)

Joint work of Vinga, Susana; Carvalho, Alexandra M; Francisco, Alexandre P; Russo, Luís MS; Almeida, Jonas S.
Main reference S. Vinga, A.M. Carvalho, A.P. Francisco, L.M.S. Russo, J.S. Almeida, "Pattern matching through Chaos Game Representation: bridging numerical and discrete data structures for biological sequence analysis," Algorithms for Molecular Biology, 7:10, 2012.
URL http://dx.doi.org/10.1186/1748-7188-7-10

Background Chaos Game Representation (CGR) is an iterated function that bijectively maps discrete sequences into a continuous domain. As a result, discrete sequences can be object of statistical and topological analyses otherwise reserved to numerical systems.

Characteristically, CGR coordinates of substrings sharing an L-long suffix will be located within 2^{-L} distance of each other. In the two decades since its original proposal, CGR has been generalized beyond its original focus on genomic sequences and has been successfully applied to a wide range of problems in bioinformatics. This presentation explores the possibility that it can be further extended to approach algorithms that rely on discrete, graph-based representations.

Results The exploratory analysis described here consists of selecting foundational string problems and refactoring them using CGR-based algorithms. We find that CGR can take the role of suffix trees and emulate sophisticated string algorithms, efficiently solving exact and approximate string matching problems such as finding all palindromes and tandem repeats, and matching with mismatches. The common feature of these problems is that they use longest common extension (LCE) queries as subtasks of their procedures, which we show to have a constant time solution with CGR. Additionally, we show that CGR can be used as a rolling hash function within the Rabin-Karp algorithm.

Conclusions The analysis of biological sequences relies on algorithmic foundations facing mounting challenges, both logistic (performance) and analytical (lack of unifying mathematical framework). CGR is found to provide the latter and to promise the former: graph-based data structures for sequence analysis operations are entailed by numerical-based data structures produced by CGR maps, providing a unifying analytical framework for a diversity of pattern matching problems.

3.36 Admixture, recombination, human population history, and local adaptation

Shuhua Xu (Shanghai Institutes for Biological Sciences, CN)

Recently available data on genome-wide high-density single nucleotide polymorphisms (SNPs), and the advent of whole-genome sequencing data for human populations, have demarcated a transition from single-locus based studies to genomics analysis of human population structure, history and local adaptation. Apart from the significant increase in the number of loci or markers, the accumulated recombination events in the genome are expected to provide additional information; in addition, it is now applicable to study human admixture at both population-level and individual-level. Here I report our recent research progress on human population history and local adaptation using recombination information in admixed genomes.

3.37 Inter-organ metabolic transport in mammals

Jun Yan (Shanghai Institutes for Biological Sciences, CN)

Complex organisms have evolved separate organs for specialized metabolic functions so that a metabolite is often synthesized in one organ but further catabolized in another. Membrane transporters, especially solute carrier (Slc) proteins, play important roles in shuttling metabolites in and out of the cells. Here we aim to reconstruct the network of inter-organ metabolic transport on the "-omic" scale. This is realized by systematically analyzing the organ- specific expression of enzymes and Slcs using microarray data and high- resolution in situ hybridization data. We provide convincing evidences that the entire metabolic network is segregated in different tissues and inter-organ transport of metabolites is facilitated by strategically located Slcs. Our study provides molecular correlates for the known inter-tissue metabolic transport systems as well as the unknown ones. We discover that there is a "metabolic code" of metabolite fluxes by combinatorial expression of enzymes and Slcs across tissues.

Participants

- Jörg Ackermann
Goethe-Universität Frankfurt am Main, DE
- Jonas Almeida
University of Alabama – Birmingham, US
- Alberto Apostolico
Georgia Inst. of Technology, US
- Benny Chor
Tel Aviv University, IL
- Matteo Comin
University of Padova, IT
- Eduardo Corel
University of Evry, FR
- Yupeng Cun
Universität Bonn, DE
- Fabio Cunial
Georgia Inst. of Technology, US
- Gilles Didier
CNRS – Marseille, FR
- Andreas Dress
Shanghai Institutes for Biological Sciences, CN
- Péter L. Erdös
Hungarian Acad. of Sciences, HU
- Ayse Funda Ergun
Simon Fraser University – Burnaby, CA
- Mareike Fischer
Universität Greifswald, DE
- Matthias Gallé
Xerox Research Center Europe – Grenoble, FR
- Raffaele Giancarlo
Universitá di Palermo, IT
- Alex Grossmann
Laboratoire Statistique & Génome – Evry, FR
- Stefan Gruenewald
Shanghai Institutes for Biological Sciences, CN
- Concettina Guerra
University of Padova, IT
- Katharina T. Huber
Univ. of East Anglia – Norwich, GB
- Deok-Soo Kim
Hanyang University – Seoul, KR
- Jack Koolen
POSTECH – Pohang, KR
- James A. Lake
Univ. of California – Los Angeles, US
- Qiang Li
Shanghai Institutes for Biological Sciences, CN
- Matthias Löwe
Universität Münster, DE
- Tobias Marschall
CWI – Amsterdam, NL
- Christoph Mayer
ZFMK – Bonn, DE
- Burkhard Morgenstern
Universität Göttingen, DE
- Axel Mosig
Ruhr-Universität Bochum, DE
- Laxmi Parida
IBM TJ Watson Research Center – Yorktown Heights, US
- Alex Pothen
Purdue University, US
- Sven Rahmann
Universität Duisburg-Essen, DE
- Walter Schubert
Universität Magdeburg, DE
- Rahul Siddharthan
The Institute of Mathematical Sciences – Chennai, IN
- Andreas Spillner
Universität Greifswald, DE
- Peter F. Stadler
Universität Leipzig, DE
- Mike Steel
University of Canterbury – Christchurch, NZ
- Gabriel Valiente
TU of Catalonia – Barcelona, ES
- Susana Vinga
Technical Univ. – Lisboa, PT
- Shuhua Xu
Shanghai Institutes for Biological Sciences, CN
- Jun Yan
Shanghai Institutes for Biological Sciences, CN