SCHLOSS DAGSTUHL

INTERNATIONAL
CONFERENCE
AND RESEARCH CENTER
FOR COMPUTER SCIENCE

Dagstuhl News

January - December 2006

Volume 9
2008

Dagstuhl Publications
Dagstuhl News January-December 2006
http://drops.dagstuhl.de/opus/volltexte/2009/2055

ISSN 1438-7581

Period: January - December 2006

Frequency: One per year

The International Conference and Research Center for Computer Science is operated by a non-profit organization. Its objective is to promote world-class research in computer science and to host research seminars which enable new ideas to be showcased, problems to be discussed and the course to be set for future development in this field.

Online version: http://www.dagstuhl.de/files/Reports/News/

Associates (2006): Gesellschaft für Informatik e.V. (GI), Bonn
Technical University of Darmstadt
Johann Wolfgang Goethe University of Frankfurt
Technical University of Kaiserslautern
University of Karlsruhe (TH)
University of Stuttgart
University of Trier
Saarland University
Max Planck Society e.V. (MPG)
French National Institute for Research in Informatics and Automatic Control (INRIA)
Dutch National Research Institute for Mathematics and Informatics (CWI)

Membership: The Center is a member of the Leibniz Association.

Funded by (2006): Federal funding and state funding of Saarland
and Rhineland Palatinate

Information: Schloss Dagstuhl Office
Saarland University
P.O. Box 15 11 50
66041 Saarbrücken, Germany
Phone: +49-681-302-4396
Fax: +49-681-302-4397
E-mail: service@dagstuhl.de
http://www.dagstuhl.de/

Welcome

Here are the Dagstuhl News for 2006, the ninth edition of the “Dagstuhl News”, a publication for the members of the Foundation “Informatikzentrum Schloss Dagstuhl”, the *Dagstuhl Foundation* for short.

The main part of this volume consists of collected resumees from the Dagstuhl Seminar Reports 2006 and Manifestos of Perspectives Workshops. We hope that you will find this information valuable for your own work or informative as to what colleagues in other research areas of Computer Science are doing. The full reports for 2006 are on the Web under URL: `http://www.dagstuhl.de/Seminars/06/`

Our online-publication service, started to publish online proceedings of our Dagstuhl Seminars, is catching on as a service to the Computer Science community. The Dagstuhl Research Online Publication Server (DROPS) (`http://www.dagstuhl.de/publikationen/publikationsserver-drops/`) hosts the proceedings of a few external workshop and conference series. We will develop a business model using the open-access policy in case the effort becomes too big to be carried from our budget.

The policy with the Dagstuhl online proceedings is that authors keep the copyrights to their contributions in order not to harm their rights to submit them to conferences or journals. We hope that the reputation of our Dagstuhl Seminars will make their proceedings a valuable source of information. It encourages us that also external workshops have asked to be hosted on DROPS.

The State and the Activities of the *Dagstuhl Foundation*

The foundation currently has 45 personal members and 7 institutional members.
We are experiencing a growing number of requests for travel support or a reduction of the seminar fees. In 2006, we have supported a number of guests in either of these ways.

Thanks

I would like to thank you for supporting Dagstuhl through your membership in the *Dagstuhl Foundation*. Thanks go to Fritz Müller for editing the resumees collected in this volume.

Reinhard Wilhelm (Scientific Director)

Saarbrücken, June 2008

Contents

Chapter 1

Data Structures, Algorithms, Complexity

1.1 Kolmogorov Complexity and Applications

Seminar No. **06051** Date **29.01.–03.02.2006**
Organizers: M. Hutter, W. Merkle and P. Vitanyi

The Kolmogorov complexity of an object is the minimal number of bits required to effectively describe the object. This complexity measure becomes ever more important in the practical setting because it gives the ultimate limits to what is achievable by data compression (a central application area) and in the theoretical setting in an ever more diverse number of areas. Shortest description length is a central theme that is basic to the pursuit of science, and goes back to the principle known as Occam's razor, one version of which amounts to the rule that if presented with a choice between indifferent alternatives, then one ought to select the simplest one. Unconsciously or explicitly, informal applications of this principle in science and mathematics abound.

Kolmogorov complexity (also known as algorithmic information theory) is widely applied in computer science and a plethora of other scientific disciplines. The seminar was meant to be cross-disciplinary and to connect through the common technique of Kolmogorov complexity the areas information theory, individual randomness, algorithmic probability, recursion theory, computational complexity, machine learning and statistics, pattern recognition, data mining, and knowledge discovery. These topics were covered by 38 technical talks; in addition there were 5 historical talks and a subsequent panel discussion on the early history of Kolmogorov complexity. The seminar was attended by 50 participants, including a large number of leading researchers from the fields listed above. The seminar enabled the participating researchers to assess the state of the art and to inform themselves about new developments, the interdisciplinary character of the seminar helped to forge cohesion in the research community.

In 2005, the field of Kolmogorov complexity is vigorously alive, with new developments consisting of books in the making or just published about

(i) the "renaissance" of recursion theory focussed on the analysis of individual randomness in terms of Kolmogorov complexity and related formalisms (R. Downey and D. Hirschfeldt, Algorithmic Randomness and Complexity, Springer, to appear);
(ii) new trends in statistical inference and learning theory, artificial intelligence, based on compression (M. Hutter, Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability, EATCS Monographs, Springer 2004);
(iii) pattern recognition, clustering and classification based on compression, Kolmogorov's structure function and algorithmic sufficient statistic, and distortion theory MDL and relations between information theory and Kolmogorov complexity (P. Vitanyi, Algorithmic Entropy, Algorithmic Distortion Theory, Springer, in preparation).
There is (iv) the area of the incompressibility method based on Kolmogorov complexity that keeps resolving decade-long (sometimes half-century) open problems in mathematics and computer science. The current trends mentioned above have been very well represented by participants and talks of the seminar.

Further material on the seminar can be found on the external seminar home page maintained by Marcus Hutter, see `http://www.idsia.ch/~marcus/dagstuhl/`.

1.2 Data Structures

Seminar No. **06091** Date **26.02.–03.03.2006**
Organizers: L. Arge, R. Sedgewick, D. Wagner

The design and analysis of algorithms is a fundamental area in computer science. This also involves the development of suitable methods for structuring the data to be manipulated by these algorithms. Hence, algorithms and data structures form a unit, and the right choice of algorithms and data structures is a crucial step in the solution of many problems. For this reason, the design, analysis and implementation of data structures form a classical field of computer science that continues to spawn exciting new research problems.

The Dagstuhl Seminar on Data Structures in 2006 reported on ongoing research on classical data structuring problems as well as application areas such as text retrieval and computational geometry. Persistent themes include randomized, cache-oblivious, and succinct data structures. Dagstuhl meetings have played an important role in developing these themes over the past decade.

As in previous meetings, there was some shift of interest away from purely theoretical issues (asymptotic analysis) towards scientific studies that are directly relevant to the use of data structures in practical applications. This shift is motivated by the desire of increasing numbers of researchers in the field to make their results available in form of programs or software packages.

Interest in the topic remains high: another attendance record was set, and several invitees who could not attend expressed their sincere regrets and their strong desire to be invited to future meetings.

A last-minute call from the organizers asked participants to think about the following questions:

- What research problems are you working on lately?
- What critical roadblocks are you facing in addressing them?
- What is the most exciting outcome you could envision if successful?
- Why should anyone be interested in your results?
- What applications do you think are most in need of new research in data structures and algorithms?
- What problems do you think other people need to be working on?

Several of the presentations were provocative responses to these questions. Beyond the scientific talks, there was a particularly fruitful (and sometimes contentious!) session that centered on whether it might be fruitful to step back and gain consensus on significant open problems in the field whose solution would have important longterm impact. As the field has matured over the past 50 (!) years, careful examination of these sorts of issues is an important part of the research landscape.

1.3 Complexity of Boolean Functions

Seminar No. **06111** Date **12.03.–17.03.2006**
Organizers: M. Krause, P. Pudlak, R. Reischuk, D. van Melkebeek

Introduction and Goals

Estimating the computational complexity of discrete functions is one of the central and classical topics in the theory of computation. Mathematicians and computer scientists have long tried to classify natural families of Boolean functions according to fundamental complexity measures like Boolean circuit size and depth. A variety of other nonuniform computational models with individual bit operations have been considered: bounded fan-in circuits, formulae, branching programs, binary decision diagrams (BDDs), span programs, etc. The analysis and relative power of these models remains a major challenge. For models of low expressive power, non-trivial efficient realizations of certain hardware-relevant functions have been found, but this question is still open in many cases. Several lower bound techniques for explicitly defined Boolean functions have been developed – most of them are of combinatorial nature. Such negative results are not only of theoretical value, but would have constructive implications, for example in cryptography and derandomization.

Methods that were originally designed to analyze the expressive power of restricted circuit models have also yielded interesting applications in other areas, such as hardware design and verification, algorithmic learning, neural computing, and quantum computing. This leads to the problem as to what type of proof method might be developed and applied at all in this setting. For higher complexity classes, we now know that the existence of natural lower bound arguments would disprove widely believed hardness assumptions. Thus, novel approaches are needed to establish lower bounds for more expressive models in discrete computational complexity.

Nowadays, investigations on the computational complexity of discrete functions have diverged and specialized into many different branches such that it becomes hard to keep a close look at all approaches. Thus, it is important to bring together researchers from different subareas in a more relaxed atmosphere Dagstuhl provides (as compared to the situation at the major international conferences in this field like STACS, STOC, CC or FOCS) to foster interaction and exchange of new ideas that might be applied in other settings as well. On the one hand, we wanted to present some of the most recent results in the different subareas to a broader audience, in particular in currently fast developing areas like, for example, approximation, communication and proof complexity or quantum computing. Secondly, we wanted to give the opportunity to discuss extensions of different proof methods as well as their applications to other fields.

Organization of the Meeting

About 60 researchers accepted our invitation to come together in Dagstuhl for this meeting. Half of them had the chance to present their results in a plenary talk. The length varied between 25 and 60 minutes, the spectrum of their focus ranged from an overview on the state of the art in a larger subarea up to the recent solution of a specific problem. In addition, there were many discussions in smaller groups between talks or after dinner.

The plenary events were structured into five morning and three afternoon sessions. Each session focussed on a special topic. We had two sessions on the basic subject circuit complexity including related models like BDDs and another one on machine-based complexity. Further topics were communication complexity, randomness, algorithms in general, and algorithmic learning. A special session was devoted to cryptography, quantum computing and quantum protocols.

Topics Discussed and Achievements

In the following we list some of the major topics that have been considered during the meeting. More details can be found in the Abstracts Collection, which are ordered alphabetically by authors' names. It also contains additional material of the participants that has been presented in smaller groups or has been evolved from discussions in Dagstuhl.

Proving lower bounds for unrestricted Boolean circuits seems unlikely to be resolved within the next years. Despite the simplicity of the computational model there are only few cases for which it is known what optimal circuits look like. Quadratic functions can be computed by a single-level of AND-gates and there has been a long standing conjecture that this circuit design is close to optimal. Stasys Jukna showed that this conjecture is far from being true by establishing an almost linear gap between optimal circuits and single-level circuits.

A problem of similar flavour is the task to realize a set of monomials by AND-gate circuits of minimal size. Now, we have many outputs since the different monomials are not connected by OR-gates – they have to be evaluated separately. For this family of Boolean functions Jan Arpe showed upper and lower bounds on the best possible circuit design that can be computed efficiently. As usual, for the lower bound one has to assume some intractability property – in this case $\mathcal{P} \neq \mathcal{NP}$ suffices. Lisa Hellerstein discussed another approximation problem, to find a smallest DNF formula that is consistent with a given truth table. Whereas it was known for a long time that finding the optimum is $\mathcal{NP}$-hard, she now showed strong lower bonds on the best possible approximation ratio under a slightly stronger intractability assumption. In addition, one can find examples for which the obvious greedy strategy performs extremly bad.

Emanuele Viola proved lower bounds for approximating the majority function by depth-3 circuits and explained how this circuit result translates into a lower bound for the classical result that $\mathcal{BPP}$ is contained in $\Sigma_2^{\mathcal{P}}$. Approximation techniques have also been considered by Stephan Waack for the case of Parity-BDDs. Ilan Newman presented a nontrivial method to approximate the maximal number of clauses in a CNF formula that can be satisfied simultaneously.

Jehoshua Bruck devoted his contribution to the parity function and discussed in detail what has been known about it. He explained how parity functions can be used to design special codes to overcome faults in storage systems. The coding problem for binary sequences where only a subset of positions have to be specified was considered by Alexander Andreev. He presented codes of nearly optimal rates that possess efficient coding circuits. The complexity of multiplying two n-bit numbers is another classical problem. Ingo Wegener showed that Nechiporuk's method can be applied to this function in the branching program model. This way one gets lower bounds of order $n^{3/2}$. A new variation of branching programs called *incremental branching programs* was advocated by Piere McKenzie. He showed tight connections of this model to classical machine models. For branching programs with a read-once restriction Martin Sauerhoff presented a construction that gives an exponential separation between the classical model and the quantum version.

Representing Boolean functions as polynomials and investigating properties of these polynomials like degree or Fourier coefficients has turned out to yield quite strong proof techniques, in particular for Boolean circuits of unbounded fanin. Frederic Green gave a longer survey talk on these methods. As new results he presented exponential lower bounds for special depth-3 $\bmod m$ circuits computing $\bmod q$ functions for m, q relatively prime. Analysing exponential sums he showed that the correlation between $\bmod m$ and $\bmod q$ functions is quite small.

Algorithmic learning of Boolean functions given a sample of function values has also been discussed. Traditionally, one has focussed on the behaviour of the learner, assuming that the teacher provides examples at random. Frank Balbach and Thomas Zeugmann have considered the opposite scenario in which the teacher should select the examples carefully such that every learner – either on average or even a stupid one – learns the Boolean function as fast as possible. It turns out that in this setting the VC-dimension of Boolean concept classes has to be replaced by other properties of Boolean functions.

The distributed bit complexity of computing Boolean functions – also called communication complexity – has been investigated by Martin Dietzfelbinger in an average case setting and for probabilistic protocols by Ronald de Wolf. For example, we have learned that a common source for classical random bits (shared randomness) cannot be compensated by quantum bit communication. Anna Gal showed how multiparty communication complexity relates to combinatorial properties of matrices and Hadamard tensors. Privacy is an important issue when designing communication protocols. Andreas Jakoby gave examples that previous characterizations of privately computable functions were incorrect and proved that a new definition is able to correctly measure the leakage of information when running an arbitrary protocol. Eike Kiltz considered privacy in an algebraic setting.

For efficiency and security, random bits are often essential. Michal Koucky discussed the problem to efficiently generate an almost unbiased random string by two agents that do not trust each other. Similarly, Ronen Shaltiel gave a new construction for disperser graphs from two weakly random sources. Several participants discussed randomness in a broader setting. Konstantin Pervyshev asked whether a single bit of advice can be helpful in randomized and quantum computations and gave a positive answer by establishing strong separations of corresponding complexity classes. Scott Diehl showed lower bounds for quantified Boolean formulae with a bounded number of quantifier alternations when given to a probabilistic TM with small space, even if a small error probability is allowed. Lance Fortnow, using the time-bounded variant of Kolmogorov complexity, established a non-trivial relation between worst-case and average-case complexity making an intractability assumption for a family of circuits with nonstandard gates.

For cryptographic applications Boolean functions should have the property that the input-output dependencies are highly masked. Claude Carlet explained what is known in this respect. He discussed in detail the notion of nonlinearity and algebraic degree. Miroslaw Kutylowski showed how bit faults in Boolean circuits can be used to efficiently attack pseudorandom generators. Bit commitment is an important basic primitive for various more complex cryptographic applications. The realization of this primitive under different system settings is still an open problem. Maciej Liskiewicz presented a quantum protocol for bit commitment that is quite robust against cheating of either party. Coding issues were addressed by Carsten Damm improving on a classical cryptosystem proposed by McEliece, and by Kazuo Iwama who investigated the capacity of a network with respect to quantum bits. A tight relationship between list-decoding of error correcting codes and amplification of the hardness of Boolean functions was presented by Valentine Kabanets.

The complexity of resolution proofs was discussed by Jacob Nordström. He considered the complexity measures width and space and showed a separation between both measures. Finally, we had several contributions adressing more general complexity theoretic questions. For example, Chris Umans gave an overview on the state of the art in matrix multiplication and presented a group theoretic approach how the upper bound on the exponent might be reduced to the value 2. David Barrington considered the classical reachability problem for graphs when restricted to grid graphs and obtained better space upper bounds. Combinatorial techniques that might be useful in the analysis of Boolean circuits were discussed by Eldar Fischer (Szemeredi's famous Regularity Lemma) and Thomas Thierauf (matchings). Peter Miltersen addressed relations towards numerics and Thomas Hofmeister efficient algorithms for string problems.

Conclusion

Understanding the complexity of Boolean functions is still one of the fundamental tasks in the theory of computation. At present, besides classical methods like substitution or degree arguments a bunch of combinatorial and algebraic techniques have been introduced to tackle this extremely difficult problem. These techniques have also found applications in other areas of computational complexity – in some cases it worked also the other way around. There has been significant progress analysing the power of randomness and quantum bits or multiparty communication protocols that help to capture the complexity of Boolean functions. For tight estimations concerning the basic, most simple model – Boolean circuits – there still seems a long way to go.

1.4 Algorithms and Complexity of Continuous Problems

Seminar No. **06391** Date **24.09.–29.09.2006**
Organizers: S. Dahlke, K. Ritter, I.H. Sloan, J.F. Traub

The seminar was devoted to the branch of computational complexity that studies continuous problems for which only partial information is available. As an important example we mention an operator equation $Lx = y$: here the right-hand side y and the coefficients of the (differential or integral) operator L are functions on some domain. These functions may only be evaluated at a finite number of properly chosen knots for the approximate computation of the solution x. Any such information about the coefficients is partial in the sense that it typically does not determine the solution x exactly.

The 8th Dagstuhl Seminar on Algorithms and Complexity of Continuous Problems attracted 50 participants from Computer Science and Mathematics, representing 11 countries and 4 continents. Among them have been 19 young researchers, some of whom have just received there diploma or master degree.

There were 43 presentations covering in particular the following topics:

- complexity and tractability of high-dimensional problems,
- complexity of operator equations and non-linear approximation,
- quantum computation,
- complexity of stochastic computation and quantization, and
- complexity and regularization of ill-posed problems,

together with applications in financial engineering and computer graphics. Abstracts are included in these Seminar Proceedings.

In addition to the substantial number of young participants another key feature of the seminar was the interaction between scientists working in different areas, namely, numerical analysis and scientific computing, probability theory and statistics, number theory, and theoretical computer science. In particular, distinguished researchers from numerical analysis were invited, and the mutual exchange of ideas was very inspiring and created many new ideas. Especially, one of the most challenging features of modern numerical analysis is the treatment of high-dimensional problems which requires several new paradigma. It has turned out that many developments that have been achieved in the IBC-community such as high-dimensional quadrature etc. will probably play a central role in this context, so that merging together the different approaches and ideas will be a very exciting topic in the near future.

Moreover, the meeting helped us to create new and to maintain the already existing various collaborations. Some ideas devoloped at the meeting have already flown into joint applications for reseach grants.

In a special event we have celebrated Henryk Wo'zniakowski, who has had his 60th birthday in 2006. Furthermore, Friedrich Pillichshammer has received the Information-based Complexity Young Researcher Award 2005, and Leszek Plaskota was the recipient of the 2006 Prize for Achievements in Information-based Complexity.

The participants of the seminar have been invited to submit a full paper to a Festschrift Issue of the Journal of Complexity.

1.5 Complexity of Constraints

Seminar No. **06401** Date **01.10.–06.10.2006**
Organizers: N. Creignou, P. Kolaitis, H. Vollmer

In a constraint satisfaction problem, the goal is to find an assignment of values to a given set of variables so that certain specified constraints are satisfied. Constraint satisfaction problems were introduced in the 1970s to model computational problems encountered in picture processing. It was quickly realized, however, that constraint satisfaction gives rise to a powerful general framework in which a wide variety of combinatorial problems can be expressed. As a matter of fact, it has been asserted that “Constraint satisfaction has a unitary theoretical model with myriad practical applications” (A. Mackworth, Foreword to *Constraint Processing* by Rina Dechter, 2003). Thus, nowadays constraint satisfaction problems (CSPs) are ubiquitous in many different areas of computer science, from artificial intelligence and database systems to circuit design, network optimization, and theory of programming languages. Consequently, it is important to analyze and pinpoint the computational complexity of certain algorithmic tasks related to constraint satisfaction. These include determining if a CSP has a solution (and, if so, finding such a solution), counting the number of solutions of a CSP, enumerating all solutions of a CSP, and finding the biggest number of constraints that can be simultaneously satisfied, if a CSP is unsatisfiable. Complexity-theoretic results about these tasks may have direct impact on, for instance, the design and processing of database query languages, or strategies in datamining, or the design and implementation of planners.

During the past two decades, an impressive array of diverse techniques from mathematical fields, such as propositional logic, model theory, Boolean function theory, universal algebra and combinatorics, have been used to analyze the computational complexity of algorithmic tasks related to CSPs. Although significant progress has been made on several fronts, some of the central questions remain unsolved so far; perhaps the most prominent of these is to obtain a complete classification of the complexity of CSPs over an arbitrary, but fixed, finite domain. One of the main aims of the Dagstuhl Seminar was to bring together researchers from all areas of activity in constraint satisfaction, so that they can communicate state-of-the-art advances and embark on a systematic interaction that will enhance the synergy between the different areas.

The organizers felt that the seminar would provide a unique opportunity to focus attention on a number of important research problems in the complexity of constraints, including the following:

- Islands of tractability of uniform CSP
- Complexity classifications for non-uniform CSP
- Quantified Constraint Satisfaction
- Study of complexity classes through the lens of Boolean CSP

1.6 Geometric Networks and Metric Space Embeddings

Seminar No. **06481** Date **26.11.–01.12.2006**
Organizers: J. Gudmundsson, R. Klein, G. Narasimhan, M. Smid, A. Wolff

Summary of the seminar

This seminar has, for the first time, brought together scientists from three different communities who are actively working on distance problems.

Geometric networks are at the core of any model for the flows of goods, traffic or information. They also play an important role in telecommunication, VLSI design, motion planning (robotics), pattern matching, data compression, bio-informatics (gene analysis), and sensor networks. One is interested in spanners with other useful properties like a linear number of edges, small total edge length, small node degree, few crossings, or small link diameter. Apart from these applications, geometric spanners have had great impact on the construction of approximation algorithms, e.g., for the traveling salesman problem. Such problems have been investigated by researchers in computational geometry and combinatorial optimization. For storage, visualization, and retrieval of high-dimensional data the question of reducing the dimension plays a crucial rôle. This has led to a theory of *metric space embeddings* that was most actively developed by scientists in discrete geometry and mathematics. Finally, mathematicians and biologists are interested in metric properties and the visualization of *phylogenetic networks*.

For each of the three fields, a survey talk was given at the seminar (Michiel Smid: *Geometric Spanner Networks*, Anupam Gupta: *Metric Embeddings*, Daniel Huson: *Application of Phylogenetic Networks in Evolutionary Studies*). In addition, twenty-two regular talks were given by the seminar participants. They encouraged a fruitful exchange of ideas and stimulated interesting discussions and co-operations.

List of Talks

presenter	*title*
Michiel Smid	Geometric spanner networks: A survey
Otfried Cheong	Computing a minimum-dilation spanning tree is NP-hard
Christian Knauer	Optimal edge deletion in polygonal cycles
Mohammad Farshi	Region-fault tolerant geometric spanners
Sergio Cabello	Multiple-source shortest paths in a genus-g graph
Anupam Gupta	Metric embeddings: A brief survey
Yuri Rabinovich	Hard metrics and Abelian Cayley graphs
Victor Chepoi	A constant factor approximation algorithm for fitting Robinson structures to distance matrices
Hubert Chan	Small hop-diameter sparse spanners for doubling metrics
Günter Rote	The geometric dilation for three points
Christian Sohler	A Fast PTAS for k-Means Clustering
Hans Burkhardt	Invariants for discrete structures – an extension of Haar integrals over transformation groups to Dirac delta-functions
Vincent Moulton	Embeddings, tight-spans and phylogenetic networks
Sándor Fekete	Geometric distance estimation for sensor networks and unit disk graphs
Daniel Huson	Survey talk: Application of Phylogenetic Networks in Evolutionary Studies
Hans-Jürgen Bandelt	Translating DNA data tables into quasi-median networks
Alexander Wolff	Computing 1-spanners – in Manhattan
Sergey Bereg	Rigid graphs and pseudo-triangulations
Matthew Katz	On two variants of the power assignment problem in radio networks
Jan Vahrenhold	Pruning dense spanners in the presence of hierarchical memory
Martin Zachariasen	Minimum-length two-connected networks
Meera Sitharam	Partial metric spaces, rigidity and geometric constraint decomposition
Mattias Andersson	Approximate distance oracles for graphs with dense clusters
Anastasios Sidiropoulos	Probabilistic embeddings of bounded genus graphs into planar graphs

Piotr Indyk	Approximation algorithms for minimum-distortion embeddings into low-dimensional spaces

1.7 Practical Approaches to Multi-Objective Optimization

Seminar No. **06501** Date **10.12.–15.12.2006**
Organizers: J. Branke, K. Deb, K. Miettinen, R. Slowinski

One can say that there are two communities dealing with multiobjective optimization problems: MCDM (multiple criteria decision making) and EMO (evolutionary multi-objective optimization) communities and they have remained rather isolated from each other during the years: they have their own conferences, journals, etc. This was the starting point and motivation of the First Dagstuhl Seminar on Practical Approaches to Multi-Objective Optimization which was organized in November 2004 (see `http://www.dagstuhl.de/de/program/calendar/semhp/?semnr=04461` for the seminar and `http://drops.dagstuhl.de/portals/index.php?semnr=04461` for the proceedings). The organizers were Juergen Branke (University of Karlsruhe, Germany), Kalyanmoy Deb (IIT Kanpur, India), Kaisa Miettinen (Helsinki School of Economics, Finland) and Ralph E. Steuer (University of Georgia, USA).

During the First Dagstuhl Seminar, two aspects clearly emerged and were unanimously agreed by all the participants. Firstly, getting both MCDM and EMO researchers and applicationists together in one seminar for five days and in a Dagstuhl environment was beneficial to both groups in terms of understanding each other's approaches better and fostering collaboration. Secondly, all the participants thought that it was a good starting event, but there was an urgent need for the two groups to arrange more such extended meetings to continue the interactions. For these reasons, the Second Dagstuhl Seminar was organized in December 2006.

In the Second Dagstuhl Seminar on Practical Approaches to Multi-Objective Optimization, about 80 researchers were invited, about 40 from the MCDM and 40 from the EMO community and, in all, about 50 researchers were able to attend the seminar. The organizers of the Second Dagstuhl Seminar were the same as in the First Seminar with the exception that Roman Slowinski (Poznan University, Poland) replaced Ralph E. Steuer.

In connection with the Second Dagstuhl Seminar, we (the organizers) decided to initiate an ambitions project of writing a book covering both MCDM and EMO approaches and their hybridization possibilities.

We believe that this book has the potential to become a key reference and inspiration for the growing community dealing with the challenges of multiobjective optimization. To start with, some of the world's best experts from both communities were invited to write chapters for the book, for example, about different approaches in MCDM and EMO and how their benefits can be put together in order to get new hybrid methods. Special attention was paid to interactive methods and methods utilizing preference information because many EMO approaches have lacked these properties until recently. The contents of the chapters were discussed in the seminar in order not to miss any important topics and, also to avoid overlaps. Besides talks devoted to book chapters, the seminar program consisted of talks on recent research trends. In addition, an important part of the seminar was active work in working groups.

The topics of the working groups were: real-world applications of multiobjective optimization, software, quality of Pareto set approximations, MODM V a learning perspective, parallel approaches for multiobjective optimization, and future challenges. Besides the invited chapters, a book chapter will be prepared based on the work of each working group.

The title of the book was decided to be "Multiobjective Optimization: Interactive and Evolutionary Approaches" and it will be published by Springer in the LNCS Series as a LNCS State-of-the-Art Survey.

On behalf of all the organizers, I would like to thank the participants for active discussions and attendance as well as for a very positive attitude towards the book project.

Chapter 2

Verification, Logic

2.1 Software Verification: Infinite-State Model Checking and Static Program Analysis

Seminar No. **06081** Date **19.02.–24.02.2006**
Organizers: P.A. Abdulla, A. Bouajjani, M. Müller-Olm

Introduction

Software systems are present at the very heart of many daily-life applications, such as in communication (telephony and mobile Internet), transportation, energy, health, etc. Such systems are often *critical* in the sense that their failure can have considerable human or economical consequences. Therefore, there is a real need of rigorous and automated methods for software development which guarantee a high level of reliability. It is well-known that, to ensure reliability, development methods must include *algorithmic analysis and verification techniques* which allow (1) automatic detection of defective system behavior and (2) automatic correctness analysis of systems with respect to their specifications.

For modern software systems, many complex aspects are of crucial importance such as manipulation of data over unbounded domains (integers, reals, etc.), object-orientation, dynamic memory structures (creation of objects, pointer manipulation), dynamic control (multi-threading, procedure calls), synchronization between concurrent processes, parameterization, real-time and hybrid modeling, etc. The development of software analysis and verification methods and tools allowing to deal efficiently with such aspects constitutes a major scientific and technological challenge. Two important and quite active research communities are particularly concerned with this challenge: the community of computer-aided verification, especially the community of (infinite-state) model checking, and the community of static program analysis. The two communities are adopting different approaches: The model-checking community studies complete methods for the verification/analysis of abstract models. These abstract models may involve infinity features such as those mentioned above. On the other hand, the program analysis community works with approximate analyses applied on formalisms that are closer to programming languages and specification formalisms used in practice.

While the approaches and the developed techniques are different, the two communities share a common mathematical background and their methods are based on common basic concepts and principles: transition systems and automata-based models, abstractions, fixpoint computations, reachability analysis based on symbolic representation structures of (potentially infinite) sets of configurations, etc.

Dagstuhl Seminar 06081

Dagstuhl Seminar 06081 "Software Verification: Infinite-State Verification and Static Program Analysis" brought together 51 researchers from these two communities in order to (1) improve and deepen the mutual understanding of the developed technologies, (2) compare these technologies and identify complementary aspects, and (3) trigger collaborations leading to new developments. The participants came from 12 countries, mainly from Europe and the US. More specifically, 1 participant came from Austria, 2 from Belgium, 1 from The Czech Republic, 1 from Denmark, 12 from France, 12 from Germany, 3 from Israel, 1 from Italy, 1 from Russia, 2 from Switzerland, 5 from the United Kingdom, and 10 from the United States.

In 31 talks, the participants presented results of their recent research. These talks touched many issues of automatic software verification including: abstraction techniques, invariant generation, termination analysis, automata-based representation structures and applications of regular model checking, analysis of pointer and heap structures, timed and hybrid systems, multi-threaded programs, parameterized systems, probabilistic models and verification methods, etc.

In a final session on Friday morning, the participants discussed how to progress further in the field of automatic software verification and how to get the developed technology to practice. Lack of common benchmarks and notations and a tendency to evaluate techniques on academic toy examples rather than on real code (e.g., from Java libraries) were identified as obstacles to fair comparison of different analyses and broader dissemination of the results. Reasons for this are that the area is quite broad with many different aspects, and that some of the techniques are still in an experimental stage.

Main Results and Approaches

This section summarizes briefly some of the main results and approaches which have been presented at the seminar.

Abstraction techniques.

A lot of effort is devoted to automated data abstraction methods, following the idea to combine predicate abstraction with counter-example guided abstraction refinement (CEGAR). One of the main issues in this context is to provide powerful and scalable techniques for automatic detection of abstractions which are sufficiently accurate for the given property. Recent developments on this topic are based on using the notion of interpolants.

Another important issue is to adapt these techniques to programs with complex control features such as procedure calls and multi-threading. Recent results propose the extension of the CEGAR framework to such programs using abstract model-checking techniques for communicating pushdown systems.

Abstraction techniques can also be used in order to enforce termination of symbolic reachability analysis. Instead of checking a property on an abstract model (which is the approach generally adopted in the model-checking community), it is possible to introduce abstraction in the analysis by considering approximate successor computations (which is usually the approach followed in the abstract interpretation community). Recent work tries to define such abstract analysis algorithms which are complete for significant classes of models (i.e., they can decide for these models whether some reachability property holds or not). Interesting results have been obtained concerning complete forward reachability analysis algorithms for the class of well-structured systems (such as Petri nets and lossy channel systems).

Automata-based techniques.

Finite-state (word/tree) automata can represent potentially infinite (regular) sets of (encodings of) system configurations. These representations can be used in the computation of reachability sets (or approximations of these sets) for the given system (when each operation of the system is modeled as a transformation on the word/tree encodings of the configurations). Techniques such as meta-transition based (or transitive closure based) acceleration, widening, or (finite range) abstraction are used in order to ensure the termination of the analysis. This approach (more and more known under the name of "regular model checking") has been adopted for dealing with various classes of systems such as counter systems (using binary encodings of integers), pushdown systems, FIFO channel systems, parameterized networks of processes, and more recently, systems with dynamic linked structures (such as lists, doubly linked lists, trees, etc).

Pointer and heap structure analysis.

Reasoning about programs with pointers and dynamic management of the memory is one of the most challenging issues in software verification. A lot of effort is devoted to finding powerful and scalable methods and techniques dealing with significant classes of such programs. Several of these works concentrate on the case of programs with lists (with possibility of sharing and cyclicality). Among these works there are approaches based on (1) logics such as fragments of separation logic or the first-order theory of Boolean algebra of sets, (2) word abstract regular model checking, (3) translations to counter automata (where counters allow to reason about the lengths of the lists), (4) instrumentation of programs, etc.

Other works provide approaches and frameworks for dealing with more general classes of programs based on (1) logics such as separation logic or fragments of first-order logic on graphs with reachability predicates, (2) tree abstract regular model checking, (3) graph rewriting, etc.

Few other works propose techniques allowing to reason on both the shapes of the dynamic linked structures and on the data they carry (i.e., the values in the data fields in each object of the structure).

Termination analysis.

Several groups are developing approaches for automatic verification of program termination. One of the approaches is based on checking the existence of a decreasing ranking function. Recent developments concern the reduction of this problem to an arithmetical decision problem. Another interesting approach is based on checking the existence of a finite union of well-founded relations covering the transitive closure of the transition relation of the program. Recent work concerning this approach proposes systematic techniques for discovering such well-founded relations iteratively using a counter-example guided principle.

More specific approaches have been developed for dealing with programs with lists. By including information on the length of the lists in the program models, it is possible to reduce the termination problem of such programs to the termination analysis of programs with counters. The latter problem can be solved using the techniques mentioned earlier.

Probabilistic models.

Recent research directions consider the verification problem of probabilistic (infinite-state) models of programs. Impressive new decidability results have been obtained recently concerning probabilistic pushdown systems and probabilistic lossy channel systems. Also decidability of equivalence and refinement checking of probabilistic programs have been studied.

Parametric verification.

Parametric verification intends to verify systems comprising a network of an arbitrary number of identical or similar components running concurrently. Typical examples of such systems are mutual exclusion, cache coherence, and broadcast protocols. Recent work in this area is concerned with inferring invariants of such networks automatically. Another technique uses abstraction: it views the network from the perspective of one component and abstracts the other components by a combination of predicate and counter abstraction.

Timed systems.

In the area of timed systems, recent decidability results about metric temporal logics and dense-time Petri nets were presented at the seminar. Although the details are quite different, a recurring idea is to reduce the problem to a problem about an untimed or discrete model for which decidability is well-understood, e.g., using the theory of well-structured systems based on the notion of well-quasi orderings.

2.2 Directed Model Checking

Seminar No. **06172** Date **26.04.–29.04.2006**
Organizers: S. Edelkamp, S. Leue, A. Lluch, W. Visser

Model checking is an increasingly popular verification technology in software and hardware systems analysis. While model checking was originally devised as a complete state space exploration technique aiming at proving models correct, its current primary use is in debugging systems by locating errors in the state space of the model. In particular in explicit-state model checking, which is often used in model checking software systems, debugging is aided by the ease with which offending system traces, also referred to as counterexamples, can be made available. Counterexamples which are system traces leading from the initial state to a property-violating state and their analysis often hints at the causes of errors.

The sheer size of the reachable state space of realistic models imposes tremendous challenges on the algorithmics underlying model checking technology. A complete exploration of the state space is often infeasible and approximations are needed. This has lead to a large body of research on optimizing the performance of model checking, including abstraction techniques, state space reduction techniques and improvements in the algorithmics of model checking.

To this end, a recent development was to reconcile model checking with heuristics guided search strategies well investigated in the area of artificial intelligence, in particular in action planning. In standard model checking, the selection of a successor node is performed in a naive fashion without taking knowledge about the problem structure into account. In action planning there is a long tradition of using heuristics guided informed search algorithms, such as A*, in the search for a state in which a planning goal has been reached. The artificial intelligence community has a long and impressive line of research in developing and improving search algorithms over very large state spaces under a broad range of assumptions. It has therefore been observed that there are many similarities between model checking and action planning as well as a large potential for synergies when reconciling these domains, see for instance the discussions at the Dagstuhl seminar 01451 Exploration of Large State Spaces. Even though directed model checking approaches have been developed for symbolic and explicit-state model checking, the most natural match seems to be between heuristics guided state space search and explicit state model checking.

The seminar brought together researchers from the system verification and the artificial intelligence domains in order to discuss the current state of the art, and to elicit and discuss research challenges and future directions. The current state of the art was documented by presentations given by the participants. The definition of research challenges was the goal of working groups that met during off-hour breakout sessions.

The seminar succeeded in documenting the state of the art in Directed Model Checking as well as defining challenges for future research. A number of publications have since emerged from the seminar. Amongst others, a paper by Dwyer, Person and Elbaum that was based on a presentation during the seminar was awarded the Best Paper Award at the Foundations of Software Engineering conference held in Portland in the Fall of 2006.

The seminar was held in the fabulous atmosphere that Dagstuhl has offered to the scientific community for more than a decade. We thank the Dagstuhl board and staff for their hospitality. The format of the seminar was that of a more unusual 3 day duration. The participants regretted the relative scarcity of time and agreed that a 5 day duration would have facilitated and enhanced the work of the working groups.

2.3 The Challenge of Software Verification

Seminar No. **06281** Date **09.07.–13.07.2006**
Organizers: M. Broy, P. Cousot, J. Misra, P. O'Hearn

Correctness of programs is a fundamental conceptual issue in computer science. But, ever since the basic works on correctness in the 1960s by Floyd, Hoare and Naur there has been a looming question: could mechanical verification be made viable, cost-effective for a wide range of software?

Throughout the world there is renewed excitement on the problem of mechanical verification of software. This can be seen in the development of software model checkers and expressive static analyses, in experimental programming languages that go beyond traditional typechecking by including assertions as part of their designs, as well as in the more traditional realms of machine-assisted proof and program construction by refinement. The purpose of this workshop is to bring together leading researchers to discuss the scientific challenge posed by software verification.

This is part of an effort at formulating a possible 15-year Grand Challenge in computer science on the problem of software verification. Several preliminary workshops on the topic have been held in the UK and US, and an international meeting on the topic was held as an IFIP working conference "Verifies Software: Tools, Theories and Experiments" in Zurich in October of 2005. A number of small commitees were charged with discussing and reporting on questions from the IFIP conference (e.g., on future work for theories, proof tools, tool integration), and presentations on these will be given at the Dagstuhl workshop.

This Dagstuhl meeting will give European researchers a chance to meet and develop their position regarding the challenge of software verification (complementing an NSF-sponsored workshop oriented to US researchers held in February 2005). That is one part of the workshop's motivation, but the overall project, and participation in this meeting, is international in nature.

A major focus of the workshop will be on understanding the state of the art in mechanical verification of industrial-scale systems, and we will invite people who have actually written and delivered code that has been proved manually or with partial machine assistance. Candidate projects include mission, safety, and security critical applications, such as the IBM CICS transaction-processing and the Mondex smart card system, as well as common system components such as file systems or even operating system kernels. Participants will be invited to contribute to this and report their findings at the workshop. After the workshop finishes we plan on producing a document outlining the current state, with an emphasis on specific practical projects, and with a view to how it might inform any future international grand challenge.

In addition to “experience talks” describing actual use of verification technology, we will also solicit presentations on challenge codes for future verification efforts. An example might be to establish the structural integrity of selected open-source infrastructure software, perhaps leading to the crash-proofing of a web server, or an operating system, or even the entire internet. Other than this, and the IFIP reports, the workshop structure will be informal, not overspecified in advance.

2.4 Specification, Verification and Test of Open Systems

Seminar No. **06411** Date **08.10.–13.10.2006**
Organizers: V. Goranko, R. Grosu, S. Merz, H. Schlingloff

Scientific motivation and goals:

In many present-day computational systems, the correct functioning of the system depends on the environment in which it is operating. An open system is one which is designed to interact with an environment only partially known in advance. The verification of an open system therefore requires the concept of alternation between system and environment. Related concepts are the construction of winning-strategies in two-player games, agents operating under imperfect knowledge, and controller synthesis for reactive systems. Currently, there are several different (and separate) scientific communities working on the correctness problem for open systems; in the computer aided verification community, it has been recognized that the correctness problem for open systems can be modelled by alternating temporal logics and automata, and there are methods and tools being developed for these formalisms. In the multi-agent-systems community, extensions of agent description languages and logics are being made which focus on the interaction of their objects in an open environment. These languages allow to describe and verify e.g. cryptographic protocols and Byzantine fault-tolerant computing. In an industrial context, test suites and controllers are automatically synthesized from formal descriptions. These artefacts are used to test and control applications such as web services and embedded systems. The goal of this seminar is to bring together researchers from different communities working on the correctness of open systems, in order to enhance the interaction and exchange of ideas. The topics will concentrate on the intersection of the three fields:

- Alternating-time logics, automata and other formalisms for open systems
- Specification and verification of multi-agent systems
- Test generation and controller synthesis for open systems

The purpose of the seminar is to establish a common understanding of the problems in the different terminologies of these application areas. It is expected that the communication between the three fields will stimulate new results and techniques of both theoretical relevance and practical usefulness. Each participant is expected to contribute to the workshop by presenting a survey of his work and / or novel results in the context of the topics of the seminar as well as by giving some open problems he or she is currently working on. It is intended to collect the state of the art and perspectives in a special volume as a "manifesto" of open systems research.

More detailed description

In many present-day computational systems, the correct functioning of the system depends on the environment in which it is operating. For example, a web service designed for the intranet may not function correctly for the internet, and a car tuned for the Sahara desert may not start in Alaska. Usually, in order to verify or test such a system one considers the composition of the system with its "maximal" or "worst case" environment. This approach, however, may not be adequate, since each possible environment has its own characteristics. Thus, the task of showing that a system is correct for every realistic environment may be different from showing its correctness in the maximal environment. It has been suggested to call a system open, if the main focus of attention is on its interaction with an unknown environment. Likewise, the composition of an open system and a particular environment is called closed, since there are no external interactions. For example, an embedded control circuit can be seen as an open system, since it connects with the plant via sensors and actuators. The loop consisting of controller and plant is a closed system. Often the envisaged environments for an open system are only partially specified; the system however should work correctly in each of them. The task of verifying an open system without closing it by its maximal environment was called module checking by some authors.

Currently, there are several different (and separate) scientific communities working on the correctness problem for open systems:

- In the computer aided verification community, it has been recognized that the correctness problem for open systems can be modelled by alternating temporal logics and automata, and there are methods and tools being developed for these formalisms.
- In the multi-agent-systems community, extensions of agent description languages and logics are being made which focus on the interaction of their objects in an open environment. These languages allow to describe and verify e.g. cryptographic protocols and Byzantine fault-tolerant computing.
- In an industrial context, test suites and controllers are automatically synthesized from formal descriptions. These artefacts are used to test and control applications such as web services and embedded systems.

Formally, the interaction between a system and its environment can be modelled as a two-player game. Thus, the module checking problem is related to the problem of finding winning strategies in certain two-player games. Another way of looking at the problem is as a multi-agent system where the strategy of one agent may be influenced by the strategy of the other agents, which is only partially known. Alternating temporal logics have been defined which allow to specify that an agent can attain a certain goal regardless of how the other agents behave. The satisfaction problem of alternating temporal logics is closely connected to the so-called controller synthesis problem: given a desired behaviour of a plant, the controller is an environment which guarantees that the plant behaves according to this specification. With suitable specification languages, the controller can be automatically generated from the specification. Moreover, it can be used as the test driver in a specification-based testing framework.

Although the problems, theories and procedures in the three abovementioned fields are closely related, it is our impression that there has not been sufficient interaction between the groups. The purpose of this seminar is to bring together researchers from the three fields. A common background for all participants is in logics and automatic verification, formal approaches to testing, and in application of state-of-the-art methods to validation of components. The main research topics to be discussed are

1. modelling formalisms for open systems,
2. logics for multi-agent systems, and
3. test generation and controller synthesis.

Correspondingly, the central themes of the seminar will be the expressiveness and complexity of alternating logics and automata, alternating epistemic logics and multi-player games, and the connections between alternation and the synthesis problem for reactive systems. The seminar, however, is not meant to be a purely theoretical one. Applications of the presented formalisms are in the description and verification of communication protocols for autonomous agents, in the semantics of composition for web services, and in algorithms for computer-aided verification tools. The organizers will try to obtain a "decent mix" of theoretical and practical presentations. In this seminar top level researchers and young researchers with a strong research perspective will congregate. Each participant is expected to contribute to the workshop by presenting his or her novel results in the context of the topics of the seminar as well as by giving some open problems he or she is currently working on. Since the participants are from separate communities, it is to be expected that some of the problems can be solved just by considering them from a different perspective.

Goals and research topics:

Here is a preliminary list of research topics which are to be tackled in this seminar.

- logics and automata for open systems
- component-based approaches and open systems

- correctness criteria for composition of open systems
- integration of verification and testing for open systems
- algorithms for module checking
- logics for multi-agent systems and multi-player games
- test case generation and controller synthesis
- synthesis of winning strategies for games

On the applied side, there are some topics which might be addressed depending on the current interest of the participants:

- testing of epistemic communication protocols
- verification and synthesis based on alternating automata
- correctness and test oracle synthesis for web services
- real-time constraints in web services

2.5 Circuits, Logic, and Games

Seminar No. **06451** Date **08.11.–10.11.2006**
Organizers: T. Schwentick, D. Thérien, H. Vollmer

Starting with the seminal paper by Furst, Saxe and Sipser, the last two decades of the previous century saw an immense interest in the computational model of Boolean circuits. Emerging powerful lower bound techniques promised progress towards solutions of major open problems in computational complexity theory. Within a very short time, further progress was made in papers by Fagin et al., Gurevich and Lewis, Håstad, Razborov, Smolensky, and Yao, to mention only a few, but then things slowed down considerably. No major progress seems to have been made in the last ten years.

As pointed out by Gurevich and Lewis, AC^o (the class of all languages accepted by families of Boolean circuits of polynomial size and constant depth) is equivalent to first-order logic. This connection is made even more precise in Immerman's theorem equating uniform AC^o with first-order logic with built-in predicates for arithmetic operations (plus and times). In fact, the mentioned results by Furst, Saxe and Sipser that parity is not in AC^o was obtained independently by Ajtai making use of model-theoretic arguments. All this indicated that lower bounds in complexity theory might be obtained via inexpressibility results in logic – an approach pioneered by Fagin. A major method for such inexpressibility results are model-theoretic games. Going back to ideas of Fraïssé and Ehrenfeucht, such games have been very successfully used in the context of first-order logic and some of its fragments and extensions, particularly monadic NP. However, all attempts to apply game-theoretic arguments in the presence of complex predicates, such as addition and multiplication, have been frustrated so far, and here also, progress has slowed down.

On the other hand, other developments in circuit theory and in logic have perhaps not been explored fully in the context of lower bounds. Most important here is the algebraic classification of regular languages. Results of Barrington, Straubing and Thérien show that most circuit classes, if they can be separated at all, can be separated by regular languages – which means that algebraic (or other) properties of such languages could be used in lower bound proofs. This also includes investigations of the so-called Crane Beach property; in some cases, but not always, the presence of a neutral letter in a language renders inoperative numerical predicates other than $<$; when this happens, lower bounds argument simplify greatly. Similar techniques have been successfully used in the context of Constraint Databases.

A further area of investigation is the structural complexity of dynamic algorithmic problems. There are, so far, no techniques available to prove that a problem does not have AC^o update complexity. This situation might be improved by finding suitable games and combining them with circuit lower bound techniques.

Organization of the Seminar and Activities

Most of the researchers invited to participate responded positively; at the end, the seminar brought together 36 researchers from different areas of complexity theory and logic with complementary expertise. The list of participants consisted of both senior and junior researchers, including a number of postdoctoral researchers and a smaller number of advanced graduate students. Altogether, though originally planned as a "small" half-week seminar, the number of participants almost reached that of a week long seminar.

Concluding Remarks and Future Plans

The organizers think that the Dagstuhl Seminar on Circuits, Logic, and Games was a big success. Bringing together researchers from different areas such of theoretical computer science and logic initiated an interaction and led to fruitful collaborations in attacking some of the open problems in this area.

Also, the participants felt that the Dagstuhl Seminar was very stimulating and provided an impetus for continuing their efforts and interaction in advancing the state-of-the-art in this area. The only negative point that came up in discussions with the organizers was that many participants thought the seminar better would have lasted for a whole week, not only three days. Finally, the organizers wish to sincerely thank the Scientific Directorate of the Center for its support of this event, and hope to have the opportunity to organize a follow-up seminar in the future (this time with a length of one week).

Chapter 3

Geometry, Image Processing, Graphics

3.1 Computational Aesthetics in Graphics, Visualization and Imaging

Seminar No. **06221** Date **28.05.–02.06.2006**
Organizers: B. Gooch, L. Neumann, W. Purgathofer, M. Sbert

The Dagstuhl-Seminar on Computational Aesthetics in Graphics, Visualization and Imaging took place from 28 May until 2 June, 2006, with 54 registered participants and some visiting PhD students from Germany. The high interest in the topics dealt at the seminar resulted in a tight scheduling of presentations and panel discussions. The program, according to the Dagstuhl tradition, was finished during the seminar. We have seen 36 presentations, and organized discussions in smaller groups at the last day of the seminar, summarizing the results and trends, and looking for answers of open questions and for application areas of computational aesthetics.

This seminar had some really important results. It was the second meeting of this topic, after the First EG Workshop on Computational Aesthetics in Girona, Spain, May 2005. At the closing session of the successful seminar it was evident that this new interdisciplinary area has already its international community established, containing mostly widely known high level researchers, their groups of students, and people from industry. Computational aesthetics has grown over its pioneer age, and became slowly a new discipline based on the techniques, algorithms of overlapping subfields of the computer imagery.

Computational Aesthetics has built new bridges and fruitful interactions between the different areas of computer imagery, and it represents a practical, new quality or meta-level. This new quality is similar to the relation of individual cars and the question of traffic regulation, safety, and control of the highways. In producing the first generation of cars these aspects or levels were not important and really predictable.

The perceptual, cognitive and artistic meta-level represented by computational aesthetics ensures in the future the communication between the researchers of computer imagery techniques, and the artist and designer community. For these people the algorithmic details are basically not really interesting, they need practical tools, e.g. in the visualization, printing, painting, movie industry, CAD systems, architecture, and other application areas.

The panel discussions have dealt with the following questions trying to address them regardless of the subfield (NPR, modeling, HDRI, color, etc...):

- What is Computational Aesthetics?
- What are the metrics in computational aesthetics? What about evaluation of success?
- What are the degrees of freedom and precision of intensioning in the field of computational aesthetics? Which are the constraints?
- Suggest unique workshop/conference formats for following up this Dagstuhl meeting
- Can we take the human out of the loop? At what point does this happen?
- How can we engage/incorporate art & design (and other) communities?
- What is the 'Holy Grail' of Computational Aesthetics? What are the grand challenges?
- Visual styles (only?) achievable by computer? (i.e. if we don't just repeat what artists have done, considering the field of computational aesthetics, what can we create? what is enabled?)
- Applications?
- Where does Computational Aesthetics come in: algorithm/process? Content? Intent? Observer?

The high interest of participants, the inspiring discussions, a lot of new ideas and the results demonstrated the impulse and potential of this rapidly launching area. The successful seminar will be continued in form of annual conferences, and further Seminars on Computational Aesthetics will be held in the future.

3.2 Human Motion – Understanding, Modeling, Capture and Animation

Seminar No. **06241** Date **11.06.–16.06.2006**
Organizers: R. Klette, D. Metaxas and B. Rosenhahn

Modeling, tracking and understanding of human motion based on video sequences is a field of research of increasing importance, with applications in sports sciences, medicine, biomechanics, animation (avatars), surveillance, and so forth. Progress in human motion analysis depends on research in computer graphics, computer vision and biomechanics. Though these fields of research are often treated separately, human motion analysis requires an interaction of computer graphics with computer vision, which also benefits from an understanding of biomechanic constraints. This seminar brought together specialists and students from these disciplines, studying and contributing to the subject of human motion analysis from different perspectives. The interdisciplinary character of the seminar allowed to bring people together which normally would not have met at disciplinary conferences.

Eadweard Muybridge (1830-1904) is known as the pioneer in motion capturing with his famous experiments in 1887 called “Animal Locomotion” (Do all feet leave the ground during the gallop of a horse? He used photography to answer the question.) The field of animal or human motion analysis has developed into many directions since then. However, human-like animation and recovery of motion is still far from being satisfactory. Various groups are dealing with different aspects of modeling, estimation and animation of human motions. Motivations differ, and define directions of research. Examples of motivations are the analysis of movements for disease detection (hip dislocations, knee injuries etc.), sports movement optimization (ski or high jumping, golf playing, swimming, etc.), the animation of avatars in movies (e.g. Gollum in Lord of the Rings), or the realistic character animation in computer games.

New results and specific research strategies have been discussed at this seminar to approach this highly complex field. The seminar intention was to discuss theoretical fundamentals related to those issues and to specify open problems and major directions of further development in the field of human motion related to computer vision, computer graphics or biomechanics. The seminar schedule was characterised by flexibility, working groups, and sufficient time for focused discussions. The participants of this seminar enjoyed the atmosphere and the services at Dagstuhl very much. The quality of this center is unique.

There will be an edited book (within Springer’s series on Computational Imaging) following the seminar, and all seminar participants have been invited to contribute with chapters. The deadline for those submissions is in September 2006 (allowing to incorporate results or ideas stimulated by the seminar), and submissions will be reviewed (as normal). Expected publication date is the end of 2007 or early 2008.

3.3 Sensor Data and Information Fusion in Computer Vision and Medicine

Seminar No. **06311** Date **30.07.–04.08.2006**
Organizers: J. Denzler, J. Hornegger, J. Kittler, C. R. Maurer JR.

More Than the Sum of Its Parts

Today many technical systems are equipped with multiple sensors and information sources, like cameras, ultrasound sensors or web data bases. It is no problem to generate an exorbitantly large amount of data, but it is mostly unsolved how to take advantage of the expectation that the collected data provide more information than the sum of its parts. The design and analysis of algorithms for sensor data and information acquisition and fusion as well as the usage in a differentiated application field was the major focus of the Seminar held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. 24 researchers, practitioners, and application experts from different areas met to summarize the current state-of-the-art technology in data and information fusion, to discuss current research problems in fusion, and to envision future demands of this challenging research field. The considered application scenarios for data and information fusion were in the fields of computer vision and medicine.

Applications of Sensor Data and Information Fusion

Computer Vision

The research field of computer vision deals with the problem of designing algorithms and building systems that gain information from images or multi-dimensional data. Typical examples are the tracking objects, the estimation of 3–D structure and motion, the recognition of objects or the analysis of complex scenes. In the seminar we studied two major tasks of fusion:

1. Strategies for the acquisition of images and data to solve individual computer vision tasks in an optimal manner, and
2. Efficient algorithms for image data and information fusion to solve computer vision and robotics problems reliably.

The fusion of statistical information extracted from images is applied to object tracking, multiple views and viewpoint planning to support active object recognition with robots. Multimodal sensor data allow for the implementation of an attention system for robots and advanced driver assistance systems rely on sensor data fusion.

Driver assistance systems, for instance, provide options like lane departure warning adaptive cruise control units. Multiple sensors are required to collect all the data to perceive the environment around the car. The fusion algorithms for this specific application underlay important constraints: It has to guarantee the demanded high degree of reliabilty and to fulfill the strong cost requirements of the automobile industry.

Medicine

Medical imaging is an emerging field which has experienced recently a tremendous reduction of innovation cycles. Progress and advances in medical imaging have an immediate impact on commercial products and clinical practice. Today various imaging modalities with completely different capabilites are available for diagnosis, intervention, surgery, or monitoring. Each modality like X-ray, computerized tomography, magnetic resonance imaging or ultrasound has its own advantage. In multi-modal image registration, images of different modalities are transformed into a single coordinate system such that an overlay fading of the images is possible. Physicians get simultaneous access to the patient's image data and no mental combination of various information sources is required anymore. Today industry provides combined, i.e. hybrid systems like SPECT/CT–, PET/CT–, or PET/MR–scanners to allow for the registration of functional and morphologic images. Software-based image registration is still considered unsolved and is an important basic research task in the field of medical image processing. The mathematical modeling and judgment of non-rigid deformations, the definition of proper similarity measures between images acquired by different modalities, the treatment of different image dimensions and the incorporation of prior knowledge are considered to be the major scientific challenges.

Methods

Image registration and data fusion can be considered as an optimization task: a proper objective function is defined, and the the fusion task is solved by the optimization of the objective function. This includes both classification and regression problems.

In image registration, for instance, the objective function is defined through distances of assigned image intensities or matched point features. Within the optimization process the transformation parameters are estimated that minimize the objective function. The registration problem based on point correspondences can be considered as a mixed integer optimization problem. Intensity based image registration requires the optimization of an objective function that measures the similarity of intensities of assigned image grid points. Commonly variational approaches are used as well as gradient descent methods. In the seminar it was also shown that the variational formulation can be interpreted in the context of optimal control of partial differential equations. Other contributions have demonstrated that registration can be done in an unified framework with image pre-processing like intensity correction, image enhancement, and segmentation. Prior knowledge for image registration can be generated by hybrid scanners or manually registered data sets. By the incorporation of specific statistically motivated regularization terms, the objective function can take account for priors. The diagnosis of incomplete multimodal image data makes use of priors, too. In the seminar novel statistical learning methods for the analysis of incomplete data as well as for the acquisition of priors were discussed.

Statistical approaches contribute to advances in decision making and classification in the presence of multiple information sources. The Bayesian theory of designing multiple expert systems provides a formalism for the treatment of sensor data fusion in pattern recognition and opens up new dimensions in classification theory. This technology can be applied to standard pattern recognition problems as well as applications like driver assistance, object tracking or robot attention control.

Conclusions

Computer vision problems are traditionally motivated by robotics and surveillance applications. Most medical image analysis problems come from the application fields medicine and biology. Though no less important, the information fusion problems posed by the robotics community tend to be more related to basic research than those arising from medicine. As they are typically dynamic in nature, must work quickly, and must effectively deal with rapidly changing and unknown environments. By comparison, the majority of information fusion problems in medicine and biology are static and operate under more controlled conditions. Not surprisingly, techniques to perform information fusion have evolved differently in these communities with minor to no overlap.

In an inspiring environment the seminar brought together members of both communities and initialized scientific discussions that yield hope for the huge potential of synergies.

Chapter 4

Artificial Intelligence, Computer Linguistic

4.1 Towards Affordance-Based Robot Control

Seminar No. **06231** Date **05.06.–09.06.2006**
Organizers: P. Doherty, G. Dorffner, J. Hertzberg, E. Rome

Today's mobile robot perception is insufficient for acting goal-directedly in unconstrained, dynamic everyday environments like a home, a factory, or a city. Subject to restrictions in bandwidth, computer power, and computation time, a robot has to react to a wealth of dynamically changing stimuli in such environments, requiring rapid, selective attention to decisive, action-relevant information of high current utility. Robust and general engineering methods for effectively and efficiently coupling perception, action and reasoning are unavailable. Interesting performance, if any, is currently only achieved by sophisticated robot programming exploiting domain features and specialties, which leaves ordinary users no chance of changing how the robot acts.

The latter facts are high barriers for introducing, for example, service robots into human living or work environments. In order to overcome these barriers, additonal R&D efforts are required. The European Commission is undertaking a determined effort to fund related basic, inter-disciplinary research in a line of Strategic Objectives, including the Cognitive Systems calls in their 6th Framework Programme (FP6). One of the funded Cognitive Systems projects is MACS ("Multisensory autonomous cognitive systems interacting with dynamic environments for perceiving and using affordances").

In Cognitive Science, an affordance in the sense of perceptual psychologist J.J. Gibson is a resource or support that the environment offers an agent for action, and that the agent can directly perceive and employ. Only rarely has this concept been used in Robotics and AI, although it offers an original perspective on coupling perception, action and reasoning, differing notably from standard hybrid robot control architectures. Taking it literally as a means or a metaphor for coupling perception and action directly, the potential is obvious that affordances offer for designing new powerful and intuitive robot control architectures.

Perceiving affordances in the environment means perception as filtered through the individual capabilities for physical action and through the current goals or intentions, thereby coupling perception and action deep down in the control architecture and providing an action-oriented interpretation of percepts in real time. Moreover, affordances provide on a high granularity level a basis for agent interaction and for learning or adapting context-dependent, goal-directed action.

The main objective of the MACS project is to explore and exploit the concept of affordances for the design and implementation of autonomous mobile robots acting goal-directedly in a dynamic environment. The aim is to develop affordance-based control as a method for robotics. That involves making affordances a first-class concept in a robot control architecture. By interfacing perception and action in terms of affordances, the project aims to provide a new way for reasoning and learning to connect with reactive robot control. The potential of this new methodology will be shown by going beyond navigation-like tasks towards goal-directed autonomous manipulation in the project demonstrators. All over, MACS aims at embedding its technical results into cognitive science.

Gibson's concept of affordances has a strong appeal. It has been used in design and other areas. Reasons for the lack of usage of the concept in the Robotics literature probably include the non-technical way in which affordances are described in the Cognitive Science literature, making it hard to operationalize the concept in the context of a robot control program. In addition, the concept of affordances as a coupling of perception and action of an individual in its environment is not unanimously accepted in the Cognitive Science literature.

During the MACS proposal phase in late 2003, the idea of organizing an interdisciplinary Dagstuhl seminar related to the core MACS topics emerged. The planned purpose of the Seminar was threefold, namely

1. to disseminate the MACS project ideas and concepts into related scientific communities,
2. to receive feedback on and discuss these ideas, and
3. to discuss the usage of affordances in other research areas.

The organizers saw researchers in four broad areas (philosophy and logic, artificial intelligence and computer science, psychology, and economics and game theory) addressing highly related (in some cases, the same) problems, in which work in one area in all likelihood would benefit research in another. Hence for the Dagstuhl seminar, the organizers felt that there would be valuable interactions and contributions that could be anticipated by bringing people together from these areas.

Goals of the Seminar

The aim of the seminar was to bring together researchers from Robotics, Informatics and the Cognitive Sciences to exchange their experiences and opinions, and generate new ideas regarding the following questions:

- How could or should a robot control architecture look like that makes use of affordances as first-class items in perceiving the environment?
- How could or should such an architecture make use of affordances for action and reasoning?
- Is there more to affordances than function-oriented perception, action and reasoning?

The answers to these questions are currently widely open. Two points can be stated with certainty, however. First, an affordance-based robot control architecture cannot simply be an extension (an "added layer", so to speak) to existing modern control architectures. The reason is that affordances would spring into existence in low-level perception, would have to pass filters in the control, such as attentional mechanisms, in order not to flood the robot's higher processing levels, and serve in some explicitly represented form of a structured result of perception as a resource for action selection, deliberation, and learning. So if there is such a thing as an affordance-based control architecture, affordances will have to play a role in all of its layers.

Second, the answers to the seminar questions do not depend on whether or not the Cognitive Sciences agree that Gibson is "right" in the sense that affordances exist in biological brains or minds or exist in the interaction between biological individuals and their environment. The point is, if Gibson's description of phenomena of functional coupling between perception and action is correct, then it is of high interest for robot control designers, independent of how it is best understood according to Cognitive Science standards. Therefore, the seminar would profit from either proponents or opponents of the affordance model. The aim here was discussion and exchange, not unanimity.

4.2 Multi-Robot Systems: Perception, Behaviors, Learning, and Action

Seminar No. **06251** Date **19.06.–23.06.2006**
Organizers: H.-D. Burkhard, M. Riedmiller, U. Schwiegelshohn, M. Veloso

The Dagstuhl Seminar on Multi-Robot Systems (06251) was held in June 20-23, 2006. It had the goal to bring researchers together from different areas of robotics to discuss current research topics on autonomous and interacting robots. The technical focus was on perception, behaviors, learning, and action. The seminar took directly place after the RoboCup robot soccer competitions and the subsequent symposium in Bremen. Thus researchers from many different countries were able to join the seminar and address issues without taking into account upcoming competitions or events.

The seminar consisted of brief presentations by many participants and extensive discussions on three issues of general interest:

- A comparison between formal and empirical methods in robotics

- Multi-robot systems with teams of homogeneous robots versus teams of heterogeneous robots
- The benefits of standard versus customizable hardware platforms in robot soccer

Reflecting the many different research issues in multi-robot systems, the talks presented a large number of different topics. A strong emphasis was on various modeling aspects while other presentations covered subjects from the analysis of human soccer games to the design of robot platforms that are well suited for robot soccer.

All discussions were very vivid. Some of presented results in the area of behavior specification for multi-agent systems led to first discussion about advantages and disadvantages of formal and empirical methods in the field of robotics. In general, it has been agreed that in new research areas, it is easier to start with empirical approaches and validate the results achieved through experiments and statistics. However, for certain systems, an empirical validation of the system and determination of its reliability are not possible either due to high costs (complex autonomous vehicles) or the consequences of a failure (autonomously guided nuclear power plants). In those cases, formal methods can help to find upper and lower bounds or even optimal solutions in some situations. However, this kind of analysis is often time consuming and in many cases the theoretical optimality of a method can not be proved for a long time, if at all.

As multi-robot systems consist of several individual robots, it must be decided whether those robots must all be identical or whether they may differ from each other. Conceptually, heterogeneous systems have advantages if there is shortage of resources such that each individual cannot be equipped with all possible features. However during the discussion, nobody was able to present an application for multi-robot systems where heterogeneous robot teams are clearly advantageous provided the cost factor is neglected. Moreover, the design and maintenance of homogeneous systems generally appears to be easier and cheaper. Only if a complex task can be decomposed in a strictly sequential series of smaller jobs, specialized heterogeneous robots can be more effective, like for example in the assembly line of a factory.

Due to the end of production for the popular “Aibo” robot, that presently is the only standard hardware platform in RoboCup, a discussion emerged about the benefits of standard versus customizable hardware platforms in robot soccer. This was specifically targeted on humanoid robot soccer and the potential successor of the “Aibo” that has been presented at this seminar for the first time. In the discussion, it turned out that it is still too early for the humanoid league to propose a standard platform as too much research is necessary to improve the hardware of the robots and determine a suitable platform. In general, a custom hardware design gives the advantage of creating a robot specifically for the desired task and therefore to overcome the limitations of commercially available solutions. On the other hand, standard platforms are usually cheaper and put a greater emphasis on software development. Thus they allow a better comparison between different algorithmic approaches.

In general, all participants agreed that the seminar was very fruitful and would like to thank the staff for their extensive support.

4.3 Foundations and Practice of Programming Multi-Agent Systems

Seminar No. **06261** Date **25.06.–30.06.2006**
Organizers: R. Bordini, M. Dastani, J.-J. Meyer

The "Foundations and Practice of Programming Multi-Agent Systems" Dagstuhl Seminar aimed at bringing together researchers interested in programming languages for multi-agent systems, agent-oriented software engineering, and various related aspects such as verification, and formal semantics. We were delighted with the result of this seminar, which gave participants a clear view of the most advanced techniques being currently investigated in research on those topics throughout the world, and also a clear understanding of all the most important open problems that need to be addressed by this research community. The seminar was particularly successful in elucidating the relationship between work being done by the "programming languages for multi-agent systems" (ProMAS) research community and the "agent-oriented software engineering" (AOSE) research community. Even though the initiative for this seminar arose from the ProMAS community, we were delighted to attract many prominent researches from the AOSE community, which allowed us to achieve the positive result on the connection of ProMAS and AOSE research.

In order to achieve the most productive setting for technical presentations and discussion in the various aspects we wanted to cover, we structured the talks in thematic days or half-days. In the first and last days, we included only AOSE-related talks. The second day was dedicated to ProMAS specifically, the morning of the third day we dedicated to Verification of Multi-Agent Systems; the fourth day had talks on Semantics in the morning and AOSE in the afternoon.

There were three discussion sessions, on the following general topics:

- agent-oriented software engineering;
- programming multi-agent systems;
- semantics and verification.

It was in the discussion sessions that many interesting ideas emerged. We were fortunate to have Maarten Sierhuis generating a detailed report of all dicussions using the Compendium Tool. He also kindly agreed to make the Compendium diagrams available online, which provides a complete report of all discussion sessions, so we feel it is not necessary to include that material in this summary. The interested reader will find all the details about the discussions we had at the following URL: `http://www.dur.ac.uk/r.bordini/Dagstuhl06261/` (choose "Discussions" in the main menu).

From the discussion session, we just would like to add a list of agent programming languages currently being used in university courses taught by the participants. The growing number of universities teaching agent-oriented programming languages is a clear indication of the increased maturity of the area and an excellent promise for a major growth of the paradigm in the medium term.

Anyone who has attended a Dagstuhl seminar agrees that Dagstuhl seminars are the most productive of all types of academic events. We were certainly expecting an excellent seminar, but this seminar exceeded our expectations in all aspects. With so many issues still to be resolved in this exciting and promising area of research, we are hoping that a follow up of this seminar will be organised in a couple of years, and we are looking forward to it already!

4.4 Robot Navigation

Seminar No. **06421** Date **15.10.–20.10.2006**
Organizers: S. Fekete, R. Fleischer, R. Klein, A. Lopez-Ortiz

For quite a number of years, researchers from various fields have studied problems motivated by Robot Navigation. On the theoretical side, a robot is faced with a number of algorithmic issues that are geometric in nature. This includes mapping a given environment, searching all possible locations in such an environment, or localizing the robot's position on a given map; typically, available information is visibility-based, but motion-planning may also require the computation of a collison-free trajectory for a rigid body, if one exists. These geometric aspects are pursued in the field of *Computational Geometry*, where quite a bit of expertise has been developed, including deep results on visibility problems and motion planning.

Another crucial feature of robot navigation is that path-planning has to be performed without full knowledge of all necessary data; such information only becomes available during the course of the robot's motion, requiring optmization with incomplete information. Complete knowledge of the scenario only becomes known after a strategy has actually been applied. This means that in addition to the geometric issues described above, an algorithm has to protect against various possibilities (including faulty sensors or inaccurate data), instead of basing its decisions on a complete description of the tasks ahead. Problems of this type are studied in the field of *Online Algorithms*.

On the other hand, computer scientists and engineers from the field of *Robotics* who work with real robots have made tremendous progress in developing systems that can perform a multitude of practical tasks. These technical posibilities give rise to a number of scenarios that have been studied in theory for a number of years. Thus, practitioners can benefit from the expertise of theoreticians. On the other hand, actual real-world scenarios tend to impose requirements that are more or less different from the ones previously considered in theory; moreover, some novel capabilities give rise to additional theoretical questions that pose new and exciting challenges.

As the possibilities and the need for real breakthroughs increase, the demand and opportunities for close interaction between practitioners and theoreticians grows. This became apparent in the Open Problems Session which saw a very lively debate on how interaction between theory and practice is seen by the various communities and how it might be improved. The central question seemed to be what is the best or correct way to model real robots such that theorectical results become meaningful for practitioners.

A predecessor workshop took place December 7 to 12, 2003. An excellent example of a successful interaction between theoreticians and practitioners is the direct result of this workshop: The video "Searching with an autonomous robot" (available at the website `http://videos.compgeom.org/socg04video/`) is based on discussions between the theoreticians Sándor Fekete (TU Braunschweig) and Rolf Klein (Universität Bonn), and the practitioner Andreas Nüchter (Fraunhofer Institute for Autonomous Intelligent Systems), who met at this Dagstuhl workshop. Using the specifications of an existing autonomous robot, a new strategy was developed for optimally locating an object hidden behind a corner. Currently, further work on broad extensions of this scenario is in the planning, showing that theory meeting practice can lead to real breakthroughs. This fruitful contact has only become possible by the previous Dagstuhl workshop on Robot Navigation.

The workshop in 2006 brought together 31 researchers from 9 different countries. The 25 presentations, varying in length, covered a large variety of topics, including selected results from online algorithms, search problems and search games, self-localization, motion and path planning, mapping, and swarm navigation. Talks were spread over the week to allow for plenty of time for discussions between the talks, thus giving participants a chance to exchange problems and ideas. We are positive that many of them will lead to new results and publications.

Chapter 5

Programming Languages, Compiler

5.1 Latently-Typed Languages

Seminar No. **06181** Date **01.05.–06.05.2006**
Organizers: R. Johnson, S. Krishnamurthi, T. Kühne, M. Sperber

Even as research into typed programming languages continues apace, languages with no prescribed type system, which we shall refer to as latently-typed languages — such as Erlang, Lua, Python, Ruby, Smalltalk, Scheme, Self — continue to be a fertile terrain for innovative research. This research spans a broad range of subject areas such as language design, programming environments, programming methodology, education, cross-language integration, and application frameworks. It is not entirely coincidental that many of these innovations have emerged in languages without a single, fixed type system.

Innovative work on latently-typed languages, however, has been done in diverse communities that have tended to not publish in the same conferences or attend the same venues, thereby losing valuable cross-pollination. These barriers have been erected partly because of an unfortunate segregation into historical language paradigms (primarily "object-oriented" versus "functional") and because of differences in emphasis ("development" versus "semantics," "industrial" versus "research," and so on).

The goal of the seminar was to unite these disparate communities to exchange ideas and identify key areas for future research, and to lay the groundwork for future cooperation. Thus, the purpose of the workshop was acquainting the participants with work that has been happening in other communities, rather than being a forum for presenting novel research ideas.

Workshop organization

Prior to the workshop, a mailing list with all participants was established. Its purpose was to solicit input on organization and content. The mailing list established areas of interest for the discussion, and collected many other suggestions that helped the organizers shape and prepare the workshop itself. The organizers consolidated the areas of interest, and used the mailing list to form groups of speakers to prepare survey talks on these subjects. Each of these groups then collaborated on their talks, which formed the backbone of the workshop schedule.

The presentation of novel research was left to a “soap-box session” of ten-minute talks, as communication of novel research was not a primary aim of the workshop. Moreover, a distributed demo session allowed implementors to show their systems.

As the workshop progressed, the lively after-talk discussions and evening conversations soon made it clear that a large number of subjects could not be accomodated in the survey talks, but was nevertheless interesting to many people. A “wish list” of additional talk subjects, paired with offers from the participants to give talks, led to yet another open session with longer talks. An open discussion session focussed on controversial statements, collected in a “Controversy Corner” during the workshop.

More discussion after the workshop produced a list of major “must-read” publications related to latently-typed languages. (The list is being maintained at `http://www.deinprogramm.de/dagstuhl-06181/papers.html`.)

Chapter 6

Software Technology

6.1 Rigorous Methods for Software Construction and Analysis

Seminar No. **06191** Date **07.05.–12.05.2006**
Organizers: J.-R. Abrial, U. Glässer

Focusing on applied formal methods, the final seminar program covered a wide range of applied research spanning from theoretical and methodological foundations to practical applications of Abstract State Machines, B, and beyond, emphasizing universal methods and tools that, regardless of their application orientation, are still committed to the ideal of mathematical rigor.

Two overarching themes were the persistent demand to

- foster further cross-fertilization between academic research and industrial development in the quest for innovative methods and tools to critically evaluate their potential in the light of new challenges as posed by new technological developments and paradigms in software engineering, and the ever-present question of
- convergence of methods, clarifying their commonalities and differences to better understand how to combine related approaches for accomplishing the various tasks in modeling, simulation, and verification of complex hardware/software systems.

For the dissemination of results, revised and refereed versions of major contributions to the seminar will be collected over the Summer 2006. Springer-Verlag kindly agreed to publish the proceedings as LNCS Festschrift.

6.2 Empirical Software Engineering

Seminar No. **06262** Date **26.06.–30.06.2006**
Organizers: V. Basili, H.D. Rombach, K. Schneider

In 1992, a Dagstuhl seminar on “Experimental Software Engineering Issues” (seminar no. 9238) was conducted. Its goal was to discuss the state of the art of Empirical Software Engineering (ESE); to assess past accomplishments, to raise open questions, and to propose a future research agenda.

Since 1992, the topic of empirical software engineering has been adopted more widely by academia as interesting and promising research topic, and by industrial practice as a necessary infrastructure technology for goal-oriented sustained process improvement. At the same time, the spectrum of methods applied in ESE has broadened. For example, in 1992, the empirical methods applied in Software Engineering were basically restricted to quantitative studies (mostly controlled experiments), whereas since then, a range of qualitative methods has been introduced, from observational to ethnographical studies. Thus, the field can be said to have moved from experimental to empirical software engineering.

We believe that it is time now to again bring together practitioners and researchers to identify the progress since 1992 and the most important challenges for the next five to ten years.

Goals of the seminar:

- Identify progress since 1992 (Dagstuhl Seminar No. 9238*)
- Summarize the state-of-the-art in ESE
- Summarize the state-of-the-practice in ESE in industry
- Develop a roadmap for research, practice, education and training

6.3 Duplication, Redundancy, and Similarity in Software

Seminar No. **06301** Date **23.07.–26.07.2006**
Organizers: R. Koschke, A. Lakhotia, E. Merlo, A. Walenstein

A venerable and long-standing goal and ideal in software development is to avoid duplication and redundancy. Duplication and redundancy can increase the size of the code, make it hard to understand the many code variants, and cause maintenance headaches. The goal of avoiding redundancy has provided the impetus to investigations on software reuse, software refactoring, modularization, and parameterization. Even in the face of the ethic of avoiding redundancy, in practice software frequently contains many redundancies and duplications. For instance the technique of “code scavenging” is frequently used, and works by copying and then pasting code fragments, thereby creating so-called “clones” of duplicated or highly similar code. Redundancies can also occur in various other ways, including because of missed reuse opportunities, purposeful duplication because of efficiency concerns, and duplication through parallel or forked development threads.

Because redundancies frequently do exist in code, methods for detecting and removing them from software are needed in many contexts. Over the past few decades, smatterings of research on these issues have contributed towards addressing the issue. Techniques for finding similar code and on removing duplication have been investigated in several specific areas such software reverse engineering, plagiarism in student programs, copyright infringement investigation, software evolution analysis, code compaction (e.g., for mobile devices), and design pattern discovery and extraction. Common to all these research areas are the problems involved in understanding the redundancies and finding similar code, either within a software system, between versions of a system, or between different systems. Although this research has progressed over decades, only recently has the pace of activity in this area picked up such that significant research momentum could be established. This seminar gathers leading scientists from all different areas related to software redundancy and young researchers ready to pick up the ball.

As organizers, we hoped the seminar would bring about a new understanding of the field and, in so doing, help lay the foundations for future research in the area. In reflecting back on the seminar, we have to conclude that it produced many successes. The discussions were lively and we know that many interesting ideas for future research were discussed in the working groups and in the open discussions during the working group reporting sessions. We believe that the variety of interests of the participants served a key purpose: we think it helped broaden the scope and forced a critical reexamination of foundational assumptions, including terminology and concepts.

6.4 Aspects For Legacy Applications

Seminar No. **06302** Date **26.07.–29.07.2006**
Organizers: S. Clarke, L. Moonen, G. Ramalingam

Programming languages and programming technologies have continued to evolve since the birth of computing, bringing significant improvements to programmer productivity and addressing software engineering issues and concerns that have become apparent over time. Unfortunately, the vast majority of the applications in use today have not benefitted from many of these advances. For example, large-scale legacy applications written in Cobol still constitute the computing backbone of many large businesses. Such applications are notoriously difficult and time-consuming to update in response to changing business requirements. Similarly, a large body of technical software and system software is written in C. Maintaining and evolving such software is even harder.

One of the fundamental reasons that such legacy applications are hard to understand, maintain, evolve, or reuse, is the scattering and tangling of code fragments addressing many different concerns. While this problem can show up even in applications written in modern languages, this is an even bigger issue with applications written in legacy languages such as Cobol and C.

Aspect-oriented software development (AOSD) has emerged over the last decade as a paradigm for separation of concerns: avoiding the aforementioned problem of code scattering and tangling. It seeks to achieve a separation of concerns even for those cross cutting concerns that are difficult to decompose and isolate with earlier programming methodologies. Aspect-oriented programming permits programmers to organize code, design and other artifacts together in a more logical, natural, way according to the concern they address. A complete application is produced by composing, or weaving, separate concerns together.

Aspect-oriented software development promises significant benefits in the areas of software comprehension, maintenance and evolution and could therefore play an important role in the revitalization of aforementioned legacy systems. However, most of existing AOSD approaches seem to have focused primarily on identifying, specifying and implementing aspects for systems that are developed from scratch. Fully exploiting AOSD in the context of legacy software systems imposes different requirements and constraints and has it's own set of challenges and open problems that need to be solved.

The seminar focused on the following question: how can aspects help us to understand, maintain, and transform legacy systems? The term "legacy systems", in this context, includes both "classic legacy" (applications written in languages such as Cobol and C) as well as "modern legacy" (applications written in object-oriented languages such as C++ and Java) and "future legacy" (applications that were developed using AOSD and need evolution).

Goals

The goal of the seminar was to bring together researchers from the domains of aspect oriented software development, software reengineering (with a focus on reverse engineering, program comprehension, software evolution and software maintenance) and program analysis to investigate how legacy systems can benefit from aspect oriented software development techniques, discuss the state of the art in aspect identification and refactoring, identify open research questions and establish a roadmap for joint research in this area.

The following topics and problems were considered relevant to the topic, though not all of these were addressed in the seminar due to time limitations:

- Idioms and design patterns in (different kinds of) legacy systems, which serve to identify common concerns/aspects in legacy systems.
- Use of aspects as views of a legacy system to improve ones understanding of that system.
- Recovering aspects (i.e., identifying the lines of code that implement a particular concern) from legacy systems.
- Aspect languages and weavers for legacy languages (especially non-object-oriented languages).
- Using aspects to guide reimplementation of a legacy system (also known as 'early aspects').

- Supporting migration from legacy implementation to an aspect-based implementation (either in the same or different base programming language).
- Evolution of systems that were developed using aspect oriented software development techniques (the legacy systems of the future).

6.5 Methods for Modelling Software Systems (MMOSS)

Seminar No. **06351** Date **27.08.–01.09.2006**
Organizers: E. Brinksma, D. Harel, A. Mader, P. Stevens, R. Wieringa

Introduction

A proper engineering approach to the construction of complex software systems requires models for their design and verification. Such models must represent both the software system and its (intended) environment. Modelling the environment is needed to define both the events in the environment to which the system must respond, and the effects that the system must have on its environment. A typical environment may consist of other software, hardware, and physical and social systems. The environment of a manufacturing control system, for example, consists of manufacturing information systems, the physical plant, and work procedures to be followed by human operators, all of which affect and complicate the task of modelling at some point of the design and verification processes.

It is clear that the quality of a design or verification process is directly affected by the quality of the models that are being used. They should meet certain quality criteria, such as correctness, understandability and maintainability. An important question that we have to address is what the relevant quality criteria for design and verification modelling are. Are there guidelines on how to achieve them? And how can we validate them?

Models invariably introduce abstractions, and the modeller has, in principle, an obligation to show that he or she has introduced the "right" abstractions. This justification is essentially informal for a number of reasons:

- Complexity: many systems are too complex to be represented without a substantial recourse to abstraction. In practical cases, formal proofs of the adequacy of such abstractions are infeasible, because they somehow rely on the availability of a non-existing model of the "full" system.
- Physical reality: this is a non-formal domain by definition, and the quality of a model with respect to its physical environment must be validated by experimental or informal means.
- Social aspects: the user environment of a software system is also informal in nature, but must be taken Into account to ensure that it will respond properly.

It is clear that most abstractions introduced by design and verification models of real systems cannot be justified formally, and must also rely on informal argumentation. The quality of any analysis based on such models, therefore, depends crucially on the quality of informal arguments.

To build models of a certain quality, we need guidelines on how to construct them well. Too often, however, such guidelines are nonexistent or simply ignored in practice, e.g. in verification modelling often "model hacking precedes model checking". And the less we understand about the way a model is built, the harder it is to validate its quality. In design modelling, designers like to proceed as if they were starting from scratch, ignoring any models that have already been developed by others. How can we improve this practice, so that we know how to build models that can be validated?

Although the problems of modelling in design and verification have many similarities, there are differences too. On the one hand, design models represent (an aspect of) the intended structure of the software-to-be. They are prescriptions that must be used by implementors and must include details needed by the implementor. Usability of the model by the implementor is an important quality criterion of these models; completeness is another. For verification models, on the other hand, abstraction is a crucial technique, preserving just as much information about the system as needed to prove correctness, and providing models that can be efficiently verified. Each property to be verified may require a different model. Still, all models, design or verification, must have some isomorphy to the modelled system, so that a from the fact that the model has a property, we can conclude that the system has this property too. This requires a good understanding of the relationship between the model and the modelled system.

The questions that we invited participants in the seminar to address were:

Model quality:

- What are quality criteria for models? How can they be quantified and checked?
- What is the relationship between models and systems in design and in verification?
- What makes an abstraction reasonable?

Modelling method:

- What are the sources and principles for the construction of good models? What is the relation between design and verification models?
- How can the structure of a model be coupled to the structure of the system? What criteria should be used for the structuring of models?
- How to bridge between informal knowledge and formal representation?
- How can we use domain knowledge, and especially engineering documentation to build correct models?

Effectiveness:

- Can we build libraries of problem frames in the domain of embedded software, or in subdomains?

Maintainability:

- Can we build models in such a way that changes in the system (versions) can be easily mapped to versions of models?
- How can changes in the verification property imply changes in the verification model?

Structure of the workshop

Almost forty participants from all over the world accepted our invitations and over thirty of them gave talks, represented by papers or abstracts in this volume. Most of the talks were thirty minutes long, some an hour. Summarising what was said seems impractical: one key observation is that participants were working in a wide variety of domains which differ in almost every important respect. The importance of (de)composition, and, closely related, abstraction, was a recurring theme, but participants' ideas of how to address it varied widely. In some contexts, it is possible to decompose a problem according to the demands of verification, for example, so as to isolate and verify a crucial element of the design. In others, problem decomposition is driven by the engineering needs of the system development, and verification must work with what it can get. We heard about many successes applying sophisticated modelling and verification procedures, especially in the domain of embedded systems. Another recurring theme was, however, that we must not and cannot assume that engineers will adopt our formalisms and notations. Success and failure can depend on aspects of the languages that formalists would not always regard as important: for example, pragmatic features such as the ability to lay out diagrams with related elements together may be important to practical usability even if they make no difference to semantics. There are several possible reactions to this, and which is best will depend on the domain: projects can plan to include a verification specialist and to insulate most engineers from the need to understand verification and its associated notations in detail, or projects can carefully choose only those formalisms that can actually be used.

Chapter 7

Applications, Multi-Domain Work

7.1 Organic Computing – Controlled Emergence

Seminar No. **06031** Date **15.01.–20.01.2006**
Organizers: K. Bellman, P. Hofmann, Ch. Müller-Schloer, H. Schmeck, R. Würtz

Organic Computing has emerged recently as a challenging vision for future information processing systems, based on the insight that it won't be long before we are surrounded by large collections of autonomous systems equipped with sensors and actuators to be aware of their environment, to communicate freely, and to organize themselves in order to perform the actions and services that seem to be required. This presence of networks of intelligent systems in our environment opens fascinating application areas but, at the same time, bears the problem of their controllability. Hence, we have to construct these systems – which we increasingly depend on – as robust, safe, flexible, and trustworthy as possible. In particular, a strong orientation of these systems towards human needs as opposed to a pure implementation of the technologically possible seems absolutely central. In order to achieve these goals, our technical systems will have to act more independently, flexibly, and autonomously, i.e. they will have to exhibit life-like properties. We call those systems "organic". Hence, an "Organic Computing System" is a technical system, which adapts dynamically to the current conditions of its environment. It will be self-organizing, self-configuring, self-healing, self-protecting, self-explaining, and context-aware.

The vision of Organic Computing and its fundamental concepts arose independently in different research areas like Neuroscience, Molecular Biology, and Computer Engineering. Self-organizing systems have been studied for quite some time by mathematicians, sociologists, physicists, economists, and computer scientists, but so far almost exclusively based on strongly simplified artificial models. Central aspects of Organic Computing systems have been and will be inspired by an analysis of information processing in biological systems. Nevertheless, the anticipated first generation of organic computing systems will still be based on well known silicon technology. Their life-like properties will arise from opening up certain degrees of freedom in the functionality of technical application systems and by the transfer of organisational concepts observable in natural systems into their system architecture.

This Dagstuhl seminar was meant as a forum for scientists from various disciplines working on key aspects of “Organic Computing” or on closely related concepts. Its objective was to initiate cooperative research on the major challenges of this vision of tomorrow?s informatics systems. Although the occurrence of emergence has been well documented in previous papers and conferences, the seminar addressed the new challenge of combining processes leading to emergence with system engineering. The challenge of “Controlled Emergence” is the possible contradiction of free running emergent processes generating new and unexpected results on the one hand, and the requirement of system engineering to design and manage a system with emergent properties in order to guarantee desired system behaviors and to avoid unwanted side effects. These problems have been discussed from the perspective of different neighbouring disciplines (like physics, chemistry, biology) and computer science with the objective of investigating the applicability of self-organizing and emergent mechanisms to technical systems.

The Results

The crosscutting themes of the seminar were intensive discussions on the exact meaning of the terms self-organization and emergence, with the accompanying emphasis on creating not only better theoretical foundations for the use of these terms, but also better operational definitions, methods, and measurements of emergence and related phenomena. While no concise final definition could be reached, the terms have been narrowed down to a more practical and touchable meaning, excluding nonscientific notions of emergence and focusing on quantitative approaches.

7.2 Atomicity: A Unifying Concept in Computer Science

Seminar No. **06121** Date **19.03.-24.03.2006**
Organizers: C.B. Jones, D. Lomet, A. Romanovsky, G. Weikum

This seminar was based on and continued the interaction of different computer-science communities that was begun in an earlier Dagstuhl seminar in April 2004. Both seminars have aimed at a deeper understanding of the fundamental concept of atomic actions and their roles in system design, execution, modeling, and correctness reasoning, and at fostering collaboration, synergies, and a unified perspective across largely separated research communities. Each of the two seminars brought together about 30 researchers and industrial practitioners from the four areas of database and transaction processing systems, fault tolerance and dependable systems, formal methods, and to smaller extent, hardware architecture and programming languages. The interpretations and roles of the atomicity concept(s) vary substantially across these communities. For example, the emphasis in database systems is on algorithms and implementation techniques for atomic transactions, whereas in dependable systems and formal methods atomicity is viewed as an intentionally imposed or postulated property of system components to simplify designs and increase dependability. Nevertheless, all four communities share the hope that it will eventually be possible to unify the different scientific viewpoints into more coherent foundations, system development principles, design methodologies, and usage guidelines.

The 2004 seminar was very successful on connecting the communities. It raised first skepticism and then curiosity about each other's viewpoints and methodologies. As a major achievement, it led to a strategic position paper, entitled "The Atomic Manifesto: a Story in Four Quarks", which appeared, in identical form, in widely read publication venues in the different communities: ACM SIGMOD Record, ACM Operating Systems Review, the Journal of Universal Computer Science, and also within the Dagstuhl Seminar Proceedings. In addition, the seminar produced a special issue of the Journal of Universal Computer Science with 8 full papers that were presented in preliminary form at the seminar and one full paper co-authored by two researchers who had not met before the seminar. The Atomicity seminar in March 2006 was intended to intensify and extend this barely blooming plant of cross-community collaboration, to revisit and refine the technical challenges identified in 2004, and to discuss the progress made in the last two years.

Results of the Seminar

The presentations and discussions at the seminar reconfirmed that a unified foundation for atomicity is a strategically important and pressing research avenue. Furthermore, the seminar was again successful in spawning new collaborations among participants, some of which span communities. It is planned to prepare another special issue for the Journal of Universal Computer Science, with full papers that hopefully emerge from this ongoing work and the results presented at the seminar.

In terms of specific research issues, the following topics led to intensive discussions and were identified as key directions within the broad theme of atomicity:

1. the integration of open nested transactions into programming languages and their run-time environment,
2. methods for providing strong guarantees about system behavior based on weaker guarantees by the underlying components,
3. handling atomic actions in time-critical environments like operating system kernels.

7.3 The Role of Business Processes in Service Oriented Architectures

Seminar No. **06291** Date **16.07.–21.07.2006**
Organizers: F. Leymann, W. Reisig, S.R. Thatte, W. van der Aalst

Motivation

More and more, applications are no longer built from scratch but by integrating pieces of software that have been built independently from each other. As a consequence, the various pieces of an application must be loosely coupled. Service oriented architectures (SOA) provide a general prescription and guidelines of how to loosely couple pieces of application functionality. Web services are a concrete instantiation of a service oriented architecture. Clearly, business processes are essential when aggregating loosely coupled functions into new application functionality. For the combination of business process technology and Web service technology the terms choreography or orchestration has been coined. These technologies are expected to become the foundational layer for tomorrow's information systems and are influencing already many application areas like Enterprize Application Integration, Software Engineering, Systems Management, Data provisioning, BPI, B2B – to name but just a few.

Software vendors today deliver products for modeling and executing choreographies. Research in this area is currently done scattered across different disciplines. The major goal of the seminar is to bring representatives from the different communities (from research, software vendors and users, of business processes and SOA) together for a first stock-taking, a joint in-depth understanding of the issues, to identify and prioritize the main research items, identify standardization needs, and to discuss demanding questions and open problems in detail. The areas to discuss include:

- Modeling Languages and Techniques for Business Processes
- Composition and refinement methods for Business Process Models
- Matching/searching process models
- Processes as constraints/semantics for Web services
- Wiring processes together (i.e., relation between BPEL and WS-CDL)
- Relation between BPEL abstract and executable processes
- Analysis of BPEL specifications
- Different transaction models in BPEL
- Business activity monitoring and process mining in the context of SOA
- Business Processes in Grid Architectures

The seminar clearly improved the understanding of the field of "Business Processes in SOA". Furthermore, new collaborations between the different communities were triggered by this event.

Presentations

The following presentations were given during the seminar:

1. Mathias Weske, Hasso-Plattner-Institut, Potsdam
   Towards Services-based Process Platforms

2. Peter Dadam, Universität Ulm
ADEPT and AristaFlow - Towards a New Dimension for Process-Aware Information Systems
3. Karsten Wolf, Universität Rostock
Controllability: A Soundness Criterion for Services
4. Kohei Honda, Queen Mary College, London
WS-CDL and Pi-Calculus
5. Jorge Cardoso, University of Madeira, Funchal
Business Process Complexity Analysis
6. Dieter König, IBM, Boeblingen, Germany
Service Composition: BPEL and SCA
7. Dimka Karastoyanova, U of Stuttgart, Germany
Semantics-Based BPEL Flexibility
8. Gerhard Pfau, IBM, Boeblingen, Germany
Management of Human Tasks
9. Satish Thatte, Microsoft, Redmond, USA
Windows Workflow Foundation: Overview and Role in SW-Stack
10. Ivana Trickovic, Alan Rickayzen, SAP, Walldorf, Germany
From Enterprise Services to Composite Applications
11. Chris Bussler, Cisco, Palo Alto, USA
SOA and BPM Technologies as Enterprise-class Computing Architecture Building Blocks
12. Sanjiva Weerawarana, WSO2, Sri Lanka
Revisiting Service Composition: Server-side Mashups
13. Oliver Guenther, Humboldt University Berlin, Germany
Price Formation for Composite Web Services
14. Guadalupe Ortiz Bellot, Centro Univ. de Merida, Spain
Aspect-Oriented Techniques for Web Services: A Model Driven Approach
15. Egon Börger, University of Pisa
An architecture for web service mediation and discovery
16. Gregor Hohpe, Google, San Francisco
Conversation Patterns
17. Carolin Letz, University of Münster, Germany
Web Service Matching - A Relational Approach

18. Kees van Hee, TU Eindhoven
    Relationships between services, components and workflows

19. Peter Massuthe, HU Berlin
    Operating Guidelines for Services

20. Jussi Vanhatalo, IBM Research, Zürich
    Techniques for Business-Driven Development

21. Peter Dadam, Ulm
    ADEPT tool demo

22. Bernhard Steffen, Universität Dortmund
    Service-Oriented Design: The jABC Approach

23. Tiziana Margaria, Universität Potsdam
    Application of the jABC Approach

24. Uwe Zdun, TU Wien
    Patterns in SOA

25. Ekkart Kindler, Universität Paderborn
    AMFIBIA and SOA

26. Jörg Desel, KU Eichstätt-Ingolstadt
    Petri nets and SOA

27. Matthias Kloppmann, IBM
    BPEL4people standard

28. Niels Lohmann, HU Berlin
    Analyzing Interacting BPEL Processes: A Tool Demo

29. Ekkart Kindler, Universität Paderborn
    AMFIBIA tool demo

Workshops and Panel

A substantial part of the week was reserved for interaction other than giving talks and discussions based on these talks. In total four half-day workshops and one panel discussion were organized. The panel discussion was chaired by Satish Thatte and had the title “On the relevance and practicality of process modeling”.

The four workshops were:

- Workshop A: “Patterns” (Gregor Hohpe)
- Workshop B: “Components” (Kees van Hee and Dieter König)

- Workshop C: “What can Theory do for Practice and what does Practice need from Theory” (Jörg Desel and Wolfgang Reisig)
- Workshop D: “Process Mining/Monitoring Processes and Services” (Wil van der Aalst)

Each of the half-day workshops led to interesting insights and fruitful discussions. The results of the workshops are enclosed in the proceedings.

Conclusion

As indicated above, the seminar clearly improved the understanding of the field of “business processes in SOA”. Furthermore, new collaborations between the different communities were triggered by this event. Several authors contributed to the proceedings of this seminar. Moreover, there will be a special issue of International Journal of Business Process Integration and Management (IJBPIM) based on this seminar.

Given the success of the seminar the organizers plan to organize a new seminar in one or two years.

7.4 Computing Media and Languages for Space-Oriented Computation

Seminar No. **06361** Date **03.09.–08.09.2006**
Organizers: A. DeHon, J.-L. Giavitto, F. Gruau

Traditional models of computation have abstracted out physical locations in space (e.g. the Internet, superscalar processors, unit delay wires, uniform memory delay) and implementations predominantly perform computations in time (i.e. sequentially). Most of our common data structures are spatially agnostic (e.g. arrays). But:

1. As scaling continues (both as primitive elements shrink to the atomic scale, and the number of elements composed scales up), computations must be distributed in space and location in space impacts the performance and feasibility of the computation.
2. As we couple and embed computing in the physical world (e.g. smart building, reactive surfaces, programmable matter, distributed robotics), position and shape are primary, serving as both the input to computation and a key part of the desired result of the computation.
3. As we understand natural computing systems (e.g. cells, ant colonies, system’s biology) location and topology define the computation. Consequently, it is important to make space not an issue to abstract away, but a first-order effect that we optimize. The distinguishing feature of spatial computing then is that computation is performed distributed in space and position and distance metrics matter to the computation.

During the workshop, three thematic areas have been identified: intensive computing where space is used as a mean and as a resource, computation embedded in space where location is important for the problem and space computation where space is fundamental to the problem and is a result of a computation.

With the cheap availability of high capacity spatial computing substrates, an emerging understanding of natural systems, and the possibility of computationally engineered matter, the importance of spatial aspects of computation is growing. These different manifestations of spatial computing have clear intersections where they can share common theory, tools, and insights. A solid mastery of spatial computation will allow us to transform our engineering capabilities, our understanding of the natural world, and ultimately the world in which we live.

7.5 From Security to Dependability

Seminar No. **06371** Date **10.09.–15.09.2006**
Organizers: C. Cachin, F.C. Freiling, J.-H. Hoepmann

Security remains an elusive property for many systems today. Despite the research efforts of the last decades, the tremendous progress made, for example in the area of cryptography, and the impressive security technology being deployed with modern operating systems, security problems have not gone away. One reason why security technology may not have been able to fulfill its promise may be a lack of integration with the existing systems, and in particular with the technologies for fault tolerance.

Although fault tolerance and security are both necessary attributes of dependable systems, these properties have traditionally been treated separately and lead to distinct and orthogonal research areas. Both research areas are based on formal models, but their separation has lead to different approaches on achieving and validating the respective properties, and the approaches have become the subject of different communities.

As one particular example, consider the area of fault-tolerant systems on the one hand and secure systems (in particular those using cryptography) on the other: Researchers in fault-tolerance often make statements about systems by treating cryptographic primitives as black boxes. This is done to keep the model tractable, i.e., to simplify analysis and (sometimes) avoid number and probability theory. In the area of safety-critical systems, such models have been successfully applied in practice, with support from automated analysis and verification tools. However, by abstracting away the basic properties of the cryptographic primitives, this severely constrains the ability to conduct rigorous security proofs. Various examples of the past show that by over-abstraction, important attributes got neglected, contributing to attack vulnerabilities in the resultant protocols.

The separate areas are only recently being viewed as complementary, with work underway to unify the two approaches. We mention the current work on tool-supported formal verification of cryptographic protocols and the concept of intrusion-tolerant systems, i.e., systems that continue to provide their service despite the corruption or failure of some of their parts.

As indicated by the above and confirmed by many researchers, there are strong similarities between the ways of modeling and handling uncertainty in the different areas of dependable systems. But there also seem to be fundamental tradeoffs that lead different communities into different directions.

Topics of the Seminar

The Dagstuhl seminar brought together researchers and practitioners from the different areas of dependability (in particular, from fault-tolerance, safety, security, and cryptography) in order to discuss the foundations of these areas, their similarities and differences. Some of the research questions discussed during the seminar included:

- What are the relations between safety, fault-tolerance, security, and cryptography with respect to methodologies and models?
- What classifications and metrics for dependability and security properties exist and how can they be compared?
- What are the differences between methods to specify, model and analyse fault-tolerant and secure systems?
- Under which circumstances can fault-tolerance techniques be used to achieve security and security methods be used to achieve fault-tolerance?
- What is the role of cryptography in the development of protocols that are both secure and fault-tolerant?

7.6 Computer Science in Sport

Seminar No. **06381** Date **17.09.–20.09.2006**
Organizers: A. Baca, L. Katz, J. Perl, O. Spaniol

The seminar dealt with a dynamically developing interdisciplinary area, where qualitative and non-deterministic paradigms from Sport like behavioural processes and modelling meet technological and structural paradigms from Computer Science. New demands, new concepts and technologies, and future trends in both disciplines were discussed.

The event brought together working areas of common interest without being focused on a single area such as pure practice, sport theories, or sport technologies. In particular, future-oriented working areas like process modelling and information processing were discussed.

Process modelling, for example, can have quite different understandings in sport science and computer science, respectively. While computer scientists mostly have their focus on deterministic and clearly defined structural processes like in industrial production, sport scientists think of non-deterministic and fuzzy-defined processes like those from games, motion, or training and rehabilitation. The seminar provided a platform for an exchange of experiences in order to establish new ideas and new solutions.

The fields of RoboCup (i.e. soccer playing robots) and, more generally, robotics, as another example, build a thrilling up-to-date working area, where computer scientists and sport scientists can collaborate effectively and successfully, dealing with motor phenomena as well as with tactical aspects of behaviour. In particular, artificial neural networks play an important role here, stretching from pattern and situation recognition over analysing and optimising learning processes to behaviour control. Internationally well known researchers as well as researchers from the younger generation have been invited to participate in this seminar. From 17.9.2006 to 20.9.2006 28 researchers from 7 countries discussed their recent work and actual tendencies in Computer Science in Sport.

There were 24 oral presentations and a discussion session on Doping and Computer Science, which was introduced by lectures given by Matthias Blatt (NADA – Bonn) and Martin Lames (Universität Augsburg). The presentations were thematically grouped into the following areas.

- General aspects
- Modelling
- Biomechanics
- RoboCup/Motion Tracking
- Multimedia/eLearning
- Data acquisition/IT

7.7 Negotiation and Market Engineering

Seminar No. **06461** Date **12.11.–17.11.2006**
Organizers: N. Jennings, G. Kersten, A., Ockenfels, C. Weinhardt

Negotiations are ubiquitous and during the last decades electronically supported negotiations became indispensable for business life. Recent years have witnessed significant changes in electronic markets and trading organization enabled by new technologies. These new technologies have created significant opportunities for reduced transaction costs, negotiation support, and automated trading. The workshop series aims to foster international collaboration among researchers working on (electronic) negotiations with different levels of analytical negotiation support, communication support, and automation as well as on trust, enforcement, and other aspects of electronic markets.

Market Engineering – the structured, systematic, and theoretically founded procedure of analyzing, designing and introducing electronic market platforms – provides a comprehensive and interdisciplinary view on electronic negotiations and markets. The structure of an electronic market is compiled of three pillars as sketched in figure 1. The focus of the workshop series is the interrelation of the ITinfrastructure, the microstructure, and the business structure for electronic negotiations. Beyond this, the influence of the overall market structure on the market performance is a key question to address. The market structure thereby does not directly determine the performance – it acts via the agent behavior. Thus, understanding the decision strategies and heuristics agents apply in electronic markets, especially in negotiations, is crucial.

The *IT-infrastructure* offers a wide range of options: starting from simple communication support (e.g. two negotiators communicating via eMail), over analytical decision support (e.g. negotiation support systems aiding in the offer analysis) to entirely automated negotiations (e.g. trading software agents). Currently, these different research branches evolve rather separately.

The *microstructure* of electronic markets does not evolve like in many traditional markets; it has to be consciously designed. The processes and protocols for an electronic negotiation have to be implemented in code and, thus, "code is law". Therefore, one should carefully engineer the rules and respective laws of the microstructure before building an electronic negotiation system. Research on negotiation protocols and cognitive processes mainly comes from economics.

The *business structure* comprises the negotiation fees and everything related to the shareholders around a negotiation or market platform or software. Questions like accessibility, monitoring, accounting, billing, and the revenue model have to be addressed for commercially operating the infrastructure. To date, the business structure is a very open field for research and the seminar intends to encourage work in this direction.

The workshop series aims at bringing together researchers working on: (1) negotiation support systems and automated negotiations as part of the IT-infrastructure, (2) researchers working on microstructure aspects like negotiation protocols, reputation mechanisms, and enforcement, (3) researchers working on negotiation strategies, heuristics, and the cognitive processes of negotiators, and (4) researchers working on the socio-economic environment including legal aspects of negotiations and electronic markets, especially on contracting and regulation. This interdisciplinary combination of applied computer science and microeconomic as well as legal, psychological, sociological research promises to lead to a holistic view on engineering electronic negotiations. The wide variety of approaches, technologies, and experiences shared by seminar participants are brought together to produce a comprehensive view on engineering economic transactions over the Internet.

Current research topics in the area of this seminar include, but are by far not limited to:

- IT-Infrastructure
    - Decision Support Systems
    - Automated Negotiations
- Microstructure
    - Negotiation and auction protocols

  - Reputation mechanisms

- Business structure
  - Evaluating, pricing, and billing negotiation support services
  - Pricing of transaction services
- Analyzing agent behaviour and testing market performance
  - Performance measurements across different protocols
  - Laboratory and field experiments; computer simulations and multi-agent systems
- Market and environment analysis
- Engineering of services supporting market platforms and participants

7.8 Perspectives Workshop: Quo vadis Informatik – Innovation dank Informatik

Seminar No. **06471** Date **19.11.–22.11.2006**
Organizers: H.-J. Bungartz, C. Meinel, R. Wilhelm

In diesem Jahr feiern wir in Deutschland das „Informatikjahr 2006“, und der vierzigste Jahrestag der Einrichtung der ersten Informatik-Fachbereiche und -Studiengänge in Deutschland steht kurz bevor. Die Informatik hat eine Phase atemberaubenden Wachstums erlebt, in wissenschaftlicher und wirtschaftlicher Hinsicht sowie bezüglich der gesellschaftlichen Auswirkungen, betreffend sowohl die Innenschau als auch den Einfluss nach außen. Sie hat sich als eigenständige Wissenschaft etabliert. Mit dem Perspektiven-Workshop wollen wir uns dem „Wie geht's weiter?“ widmen und einige Schlüsselfragen diskutieren:

Einfluss auf das Weltbild:
Welche Beiträge philosophischer Relevanz hat die Informatik geleistet? Wie haben diese einerseits das heutige Erscheinungsbild der Informatik geprägt und andererseits deren fundamentale Bedeutung unterstrichen?

Identität:
Angesichts ihrer grundlegenden Beiträge und Einblicke hat die Informatik eher Nähe zu Mathematik und Physik, ja sogar zur Philosophie; ihre omnipräsente Praxisrelevanz sowie ihre Nähe zu Technologien und konkreten Produkten lassen sie eher als Ingenieurwissenschaft erscheinen. Welches dieser „Gesichter“ ist das einflussreichere, welches inspiriert mehr? Die unmittelbaren Konsequenzen einer diesbezüglichen Einschätzung für die Gestalt zukünftiger Studienpläne, für das Qualifikationsprofil unseres wissenschaftlichen Nachwuchses oder auch – ganz profan – für die Definition

von Mechanismen zur leistungsbezogenen Mittelzuteilung verdeutlichen, dass dies weit mehr als ein rein philosophisches Thema ist.

Nur noch Service-Provider?
Wird sich die Informatik zu einer Basis- und Querschnittstechnologie ähnlich der Mathematik entwickeln – mit allen damit verbundenen Implikationen wie dem oft beklagten Zustand, mehr Service-Provider für die Forschungsaufgaben anderer zu sein, als mit eigenen Herausforderungen punkten zu können?

Unsichtbare Innovation:
Viele heutige Technologien sind Computer-gestützt und Software-dominiert – ermöglicht „dank Informatik“ eben. Abgesehen von allem rund um den Computer wird das resultierende Endprodukt aber meistens mit einer anderen Disziplin assoziiert – ein prominentes Beispiel hierfür ist das Automobil. Wie gehen wir damit um?

Erfolgsfaktoren?
Welche Anteile an den Erfolgen der Informatik sind der Hardware (Moore & more), welche der Algorithmik und welche den Informatik-Systemen zuzuordnen?

Zerreißprobe?
Kann bzw. soll überhaupt die Informatik ihre integrale Identität wahren ? vor dem Hintergrund, dass Teilgebiete wie Wirtschaftsinformatik und Höchstleistungsrechnen beispielsweise vielleicht weniger miteinander zu tun haben als Wirtschaftsinformatik und BWL einerseits oder Höchstleistungsrechnen und Angewandte Mathematik oder Computational Sciences andererseits?

Herr im Haus?
Sollte die Informatik versuchen, die Hoheit über Informatik-F&E zu behalten bzw. zu gewinnen, oder werden ohnehin immer größere Anteile in Anwendungsgebiete ausgelagert werden?

Wie soft?
Wird die Informatik in 10 Jahren Mathematik-freie Zone sein? Oder anders herum: Wie „soft“ darf Informatik sein?

Eher Fluch als Segen?
Was ist wo schief gegangen, wenn manche Errungenschaften der Informatik eher als Fluch denn als Segen wahrgenommen werden?

Chapter 8

Distributed Computation, Networks, VLSI, Architecture

8.1 Perspectives Workshop: Autonomic Networking

Seminar No. **06011** Date **03.01.–06.01.2006**
Organizers: G. Carle, R.H. Katz, B. Plattner, M. Smirnov

The Internet did transform from a scientific network for specialists into an ubiquitously used network that provides the basis for a growing number of applications. Among the issues of highest importance for the future evolution of communication networks are functional adaptability or extensibility and robustness against a wide range of possible faults and attacks.

Existing approaches for extending or adapting functionality, and for identifying sources for malfunctioning and faults (such as device failures, mis-configuration or attacks) typically require a significant amount of human intervention and are therefore slow, error-prone and expensive. For both areas, functional flexibility and fault management, a promising approach is to introduce self-organizing algorithms into the network that capable of autonomic behaviour, thereby providing some kind of intelligence that reduces the need of human intervention.

The term **Autonomic Networking** is used to describe communication networks consisting of self-managing elements capable to support self-configuration, self-healing and self-optimization. These desired properties require components for observation, for assessing the observed data, for representing and applying knowledge about constraints and goals. Due to the complexity of the network, and due to the multitude of administrative borders, decentralized self-organizing algorithms are required, with autonomous capabilities of individual nodes. The ultimate aim is to derive design paradigms for communication networks and distributed computing environments that are capable of providing rapidly adapting services and applications in scenarios where networked devices and users interact in a highly dynamic manner.

This seminar aims for building a scientific community by bringing together researchers and engineers who have gained experience in different emerging network technologies and related fields such as self-organization and peer-to-peer networking, network tomography and measurements, autonomic computing, organic computing and bio-inspired techniques, active and programmable networking, and service creation and management.

The seminar aims for identifying common research challenges and for discussing potential solutions. As an outcome of this seminar, synergies and open issues among different areas will be identified, allowing to better align research in the different communities.

Areas of interest include:

- Theoretical foundations for networks with autonomic elements
- Tools and techniques for designing, analyzing and building autonomic networks and systems
- Network monitoring and measurements for self-managing networks and systems
- Adaptive security and safety mechanisms for self-protection and self-healing of networks and systems
- Advanced information processing techniques for autonomic networking and self-organization, in particular statistical and optimization techniques, policy-based techniques, context-awareness, algorithms from control theory, machine learning, knowledge-based and goal-driven role-based mechanisms
- Models suitable for autonomic networking, including models from related disciplines such as models for automation and control, models of biological systems, or models of economic systems
- Languages, development and securely programmable environments for autonomic communications systems
- Applications and example scenarios
- Human interaction with autonomic networks and systems

8.2 Architectures and Algorithms for Petascale Computing

Seminar No. **06071** Date **12.02.–17.02.2006**
Organizers: U. Rüde, H.D. Simon, P. Sloot

This seminar will focus on *high end simulation* as a tool for computational science and engineering applications. To be useful tools for science, such simulations must be based on accurate mathematical descriptions of the processes and thus they begin with mathematical formulations, such as partial differential equations or integral equations.

Because of the ever-growing complexity of scientific and engineering problems, computational needs continue to increase rapidly. But most of the currently available hardware, software, systems, and algorithms are primarily focused on business applications or smaller scale scientific and engineering problems, and cannot meet the high-end computing needs of cutting-edge scientific and engineering work. This seminar is concerned with *peta-scale* scientific computation, which is highly computation- and data-intensive, and cannot be satisfied in today's typical cluster environment. The target hosts for these tools are systems comprised of thousands to tens of thousands of processors. By the end of the decade such systems are expected to reach a performance of one Petaflop, that is 10^{15} floating point operations per second.

The rapid progress over the past three decades in high performance simulation has recently been facing an increasing number of obstacles that are so fundamental that no single solution is in sight. Instead, only a combination of approaches seems to promise the successful transition to petascale simulation.

- Petaflops systems are necessarily massively parallel. Many simulation codes currently in use (e.g. commercial finite element packages) are hardly scalable on parallel systems at all, but even specifically developed programs cannot be expected to scale successfully to tens of thousands of processing units, as will be used in Petascale systems.

- Achieving a significant percentage of the processor performance has become increasingly difficult especially for many commodity processors, since the so called *memory wall* prohibits better efficiency. The compute speed is not matched by the memory performance of such systems. Mainstream computer and CPU architecture is hitting severe limits which may be most noticeable in a high performance computing scenario (but not only there).

- Further improvements of latency and bandwidth are hitting fundamental limits. At 10 GHz clock rate, light travels for 3 cm in vacuum, but a Petaflop system may be physically 100 m across, so that latencies of several thousand clock cycles are unavoidable for such a system.

- Similarly, a Petaflop system would ideally have an aggregate bandwidth that, if transported on buses of 128 bit width at a clock rate of 1 GHz, would require in excess of a million of such buses operating in parallel. Therefore not only latency, but even the available bandwidth may become a severe bottleneck.

- New and innovative hard- and software architectures will be required, but it will not be sufficient to design solutions only on the system level:

- additionally the *design (and implementation) of the algorithms must be revised* and adapted.

- new latency tolerant and bandwidth optimized algorithms must be invented and designed for petascale systems.

The proposed seminar will focus on develping solutions for concurrent and future problems in high end computing. Specifically, these are:

- innovative hard- and software architectures for petascale computing
- scalable parallel simulation algorithms, whose complexity must depend only linearly (or almost linearly) on the problem size
- scalable massively parallel systems and architectures
- simultaneously using multiple granularity levels of parallelism, from instruction or task level to message passing in a networked cluster
- devising algorithms and implementation techniques capable of tolerating latency and bandwidth restrictions Petaflop systems
- tools and techniques for improving the usability of such systems
- possible alternatives to silicon-based computing

8.3 Peer-to-Peer-Systems and -Applications

Seminar No. **06131** Date **26.03.–29.03.2006**
Organizers: A.D. Joseph, R. Steinmetz, I. Stoica, K. Wehrle

Under the term “Peer-to-Peer”, a very promising paradigm for communication in the Internet arises. Though it was originally used for pragmatic and not always legal file-sharing activities, the Peer-to-Peer technology offers interesting opportunities for highly distributed and scalable systems and applications.

According to recent reports from ISPs, a major amount of Internet traffic is governed by Peer-to-Peer applications. Due to the continuing growth and diversification of the Internet and its applications, it becomes exceedingly difficult to meet the resource demands by traditional Client-Server solutions. These centralized approaches can be hardly realized for, e.g., file sharing applications, distributed file systems, or grid computing environments.

Given this persistent and long-term development, there are three fundamental challenges for current and future Internet applications:

- Scalability is of utmost importance in order to cope with user bases and resource consumption of applications (in terms of bandwidth, storage, processing, etc.) growing by several orders of magnitude.
- Only through security and reliability it is possible to maintain the availability of centralized services in the face of distributed denial-of-service attacks. Data privacy and censorresistance are also of growing concern.

- Flexibility and quality of service allow the rapid deployment of new technologies throughout the Internet, e.g. to realize long-promised multicast and host mobility features.

The Peer-to-Peer paradigm shows the potential to meet these challenges. Peer-to-Peer systems share distributed resources and address services based on content rather than location. Without the necessity for central entities, they organize themselves into cooperating infrastructures of symmetric peers. The two approaches of structured vs. unstructured P2P systems allow for different and sometimes complementary trade-offs and still bear a wide range of ongoing and future research in the P2P area.

The goal of the second Dagstuhl seminar on Peer-to-Peer-Systems and -Applications was to assemble researchers being highly active in the area of Peer-to-Peer mechanisms and networking

- to reflect on recent research activities,
- to identify future research issues, i.e. major challenges and
- to strengthen the Peer-to-Peer community in research.

8.4 Internet Economics

Seminar No. **06132** Date **28.03.–31.03.2006**
Organizers: M. Karsten, L. McKnight, P. Reichl, B. Stiller

Driven by a strong commercialization of the Internet as a networking technology and as a business-enabling platform the use of this network has changed dramatically in the last 5-8 years. The immense range of applications and services, the complex set of management tasks for network operations and business control, and an increase in customer and provider demands has opened up the necessity of combining well-known network principles with economic basics. However, these combinations are still in a quite unbalanced state, as it is unclear how the many roles found in distributed systems providing services in a coordinated manner – considered as a must – do behave in a competitive market. Technical functions as well as economic means considered as a whole only will enable a world-wide, interoperable network with those incentives given for providers and customers to participate, offer, and utilize the information network of the near future.

The economic part of this situation is guided by business policy management, the economics of Internet services and applications in a mobile and fixed environment, multi-provider schemes, regulation aspects of countries and continents, cost modeling tasks, and charging as well as pricing support for services and content. The technological and networking basis for such schemes to be offered in an operational manner, covering inter-disciplinary feedback and control, finds its foundation in a policy-based management of Service Level Agreements (SLA) and Quality-of-Service (QoS), monitoring and measurement schemes for high-performance multi-provider networks, security mechanisms, and explicit peer-to-peer as well as grid computing support.

Both areas outlined above need to be focused onto a common target, which will be achievable due to joint discussions and identification of common solution proposals, investigations of their feasibility, and a consolidation of technical and economic mechanisms to enable a fast, guaranteed, and efficient provisioning of differentiated services in multi-provider Internet.

This seminar did proceed with 11 presentations and subsequent discussions, which addressed the following areas:

- Individual Tariffs for Mobile Commerce Services — A Computational Model
- A Decentralized Market Place and its Application to Bandwidth Trading
- Compensation Analysis of Next Generation Services
- Multipath Routing, Coordinated Congestion Control, Efficiency, and Cooperation
- IMS Charging — A Key Step in Realizing Next Generation Networks
- Congestion Pricing and Non-cooperative Games in Communication Networks
- How Economic Models Fail
- A Decentralized Learning Algorithm when Pricing a RED Buffer
- Reshaping the Agenda
- The Impact of Information and Computational Limitations in Mechanism Design
- Licensing Models for the Grid

8.5 Dynamically Reconfigurable Architectures

Seminar No. **06141** Date **02.04.–07.04.2006**
Organizers: P.M. Athanas, J. Becker, G. Brebner, J. Teich

Dynamic and partial reconfiguration of hardware architectures such as FPGAs and XPPs brings an additional level of flexibility in the design of electronic systems by exploiting the possibility of configuring functions on-demand during run-time. This has led to many new ways of approaching existing research topics in the area of hardware design and optimization techniques. For example, the possibility of performing adaptation during run-time raises questions in the areas of dynamic control, real-time response, on-line power management and design complexity, since the reconfigurability increases the design space towards infinity.

This Dagstuhl Seminar on Reconfigurable Architectures has aimed at raising a few of these topics e.g. on-line placement, pre-routing/on-line routing trade-off, power minimization etc., and also at presenting novel ideas on how to overcome the difficulties introduced in dynamic reconfigurable systems.

The first issue in a reconfigurable system is how to manage the dynamic placement of functional modules on the reconfigurable area. Current reconfigurable architectures are restricted on the granularity of the reconfigurable parts, which results in a limited flexibility in the partitioning of the reconfigurable blocks. Novel methods for overcoming these difficulties had been discussed during the seminar, as well as how future alternative architectures can be adapted for avoiding such problems. The highest level of flexibility during reconfiguration would be achieved with on-line routing of functional descriptions in order to dynamically place functions on hardware resources that are currently available. On-line routing would make it possible to dynamically adapt the bit width of communication channels. Such a scenario would, however, lead to an increased response time and in some cases even an increased level of power consumption, due to the increased processing required to perform the re-routing. It seems reasonable to introduce dynamic reconfiguration in the system behavior in a way that enables the dynamic adaptation to changing system requirements on e.g. power consumption, performance, accuracy and resource utilization. For example, the placement and routing of a function can be dynamically adapted to provide either high performance or low power, according to the current system workload and power limitations.

These first investigations on dynamically reconfigurable architectures presented and discussed within this Dagstuhl Seminar dealt with clarifying exactly what level of flexibility can be achieved in current architectures, how to further increase this flexibility by exploiting novel methods, and how to integrate and exploit this flexibility in an optimal manner.

The workshop started on Monday morning with the mutual introduction of each participant and by posing exactly one question to the audience on what he/she sees the most important challenge in reconfigurable computing. Many questions dealt with the problem of dynamic reconfiguration. Issues such as need (killer applications), testability, speed and overhead were questioned initially until Andreas Koch found and summarized on Wednesday evening more than ten application areas requiring or benefiting from dynamic hardware reconfiguration. One of these applications was also presented by Oliver Diessel on Thursday morning, namely an industrial project exploiting dynamic hardware reconfiguration, namely positioning of satellite receivers that exploit the diversity in satellite signals to mitigate the effects of interference. Other questions dealt with reconfigurable interconnect, power consumption problems and competitiveness issues of actual reconfigurable devices with respect to ASIC implementations. On the tool side, questions focused on actual problems about stability and support for design flows enabling dynamic hardware reconfiguration. Finally, questions addressed the usefulness of libraries for reconfigurable computing. These questions formed the basis of the Monday evening breakout session where individual groups discussed application domains and module types (such as processing, I/O, memory, communication, operating systems types) that might be reusable in the reconfigurable computing community. A collection of domains and useful module types was summarized and added to the web page `http://www.r-space.de` under the name ReCoLib. This website is intended to collect library modules as well as benchmarks for reconfigurable computing in the future.

The third day of the seminar (Wednesday) contained just four presentations during the morning (with the presenters invited to be controversial), and a group discussion in the evening. The theme for the day was: “New thought models for reconfigurability and programmability”. The aim was to look beyond immediate aspects of dynamically reconfigurable architectures towards a longer-term research agenda in the general area of the seminar. In an opening presentation, Gordon Brebner presented arguments in support of entitling the next seminar in the series as “dynamically adaptable behaviors” rather than “dynamically reconfigurable architectures”. The intent was to support a move towards a more applications-centric view of the field. That is, given the relative maturity of research on the architectural side, one should seek natural interpretations of reconfigurability in the needs of applications. The presentation did not advocate ad hoc per-application design approaches, but favored research in domain-specific frameworks that support desirable dynamically adaptable behaviors. In feedback, and on other occasions during the seminar, there seemed to be some consensus on the viewpoint. Rather than perennially chasing one elusive “killer application” for general reconfigurable architectures, we can find a variety of uses by studying a variety of applications. Apt choice of applications is important though, and Peter Athanas made this point strongly, with the presentation titled “The (empty?) Promise of FPGA Supercomputing” conveying his general argument: that there may be serious shortcomings when attempting to use current reconfigurable technologies for high performance computing applications. Hartmut Schmeck contributed a very interesting overview of the new field of “organic computing”, where there is a particular need for properties like robustness, adaptivity, and flexibility. His presentation offered a general systems setting – requiring self-organization with some degree of control – that is perfectly compatible with implementing dynamically adaptable behaviors using dynamically reconfigurable architectures. Finally, Reiner Hartenstein discussed the three paradoxes of Reconfigurable Computing: the low power paradox; the high performance paradox; and the education paradox. The last of these is a critical one for further development of the field. In order for a research breakthrough to have widespread impact, broad education in the new possibilities is essential to encourage thinking away from tradition paradigms. This of course has to be backed up by appropriate software tools, not just physical-design technologies. In the evening, a discussion session was held, titled “Dynamic reconfiguration considered harmful?”. This featured interesting and wide-ranging discussion of the achievements in our area and the practical effect of the research activities. One particular concrete outcome was the collection of a “Top Ten” list of applications of dynamic reconfiguration, collated by Andreas Koch. This served as a very useful assurance that the efforts of researchers in this area have not been in vain. It also confirmed that different behaviors of different applications provide a different scope for the use of reconfigurable architectures.

The gap between tools and available technology became the central topic on Thursday with eight very interesting contributions. Half of these talks dealt with operating systems and task concepts in support of reconfigurable computing devices such as FPGAs. Neil Bergmann, for example, advocated the use of standard Linux processes to enable a codesign for reconfigurable hardware/software designs by introducing ideas how to port typical Linux IPC mechanisms into hardware. Vincent Mooney presented a framework called Delta for generating RTOS for FPGAs including possibly multiple processors, and, Marco Platzner and Florian Dittmann presented algorithms for scheduling reconfigurable hardware modules with real-time constraints on dynamically reconfigurable devices such as the

Erlangen Slot Machine (ESM), an FPGA-based platform for supporting dynamic reconfiguration and inter-module communication between modules arranged in slots. The talks by Roger Woods, Andreas Koch addressed questions of compiling data flow application, and C-code specifications respectively onto reconfigurable hardware. Finally, Klaus Waldschmidt introduced interesting perspectives on how reliability affects power consumption for multi-core architectures based on the introduction of novel dynamic power management techniques that avoid thermal cycling, combined with dynamic workload distribution.

8.6 Scalable Data Management in Evolving Networks

Seminar No. **06431** Date **22.10.–27.10.2006**
Organizers: S. Böttcher, L. Gruenwald, P.J. Marrón, E. Pitoura

Data management technologies have been widely used in fixed-wired networks within the last two decades. Scalable technologies for query evaluation, transaction management and data storage have been developed for client-server systems and are widely used in industry. Nevertheless, key data management technologies are limited in their support of evolving networks including mobile ad-hoc networks (MANETs), peer-to-peer (P2P) systems, and sensor networks. More specifically, current database system technologies lack sufficient features for handling the mobility and ad-hoc aspects of many networking infrastructures. This includes cross-layer optimization involving network and data management aspects, location based data management and optimization, and mechanisms for handling error situations, such as node failures, link failures, and network partitioning. Therefore, an adaptation of current data management technologies and their applications to mobility and ad-hoc requirements is essential for them to be useful to mobile users.

During this one week seminar, the 25 participants that came to Dagstuhl were actively involved in discussions about these topics and discussed the requirements and protocols for atomic transactions in mobile environments, the role and definition of P2P databases, and the differences in characteristics of environments like MANETs, P2P, and sensor networks.

One focused working group discussed the application scenarios, the technical requirements, and the open research questions for atomic transaction commitment in mobile ad-hoc environments. After collecting the requirements in different application fields ranging from emergency scenarios to mobile business and mobile gaming, the group investigated how the requirements are met by current atomic commit protocol implementations. One of the conclusions drawn was that although much research has been contributed to this field, the atomic commit problem is still open and unsolved for server-less applications and a demand for further research in this area exists.

Another focused working group discussed key requirements for P2P databases. Since P2P systems allow new forms of data and resource sharing among fast-changing communities having thousands of users, new ways of modelling, indexing, storing and querying data are required and must be adapted to P2P overlay structures.

The third working group discussed and compared a huge variety of extensions of conventional data management technologies that have been proposed for meeting the particular requirements of MANETs, P2P, and sensor networks. The discussion focused on the aspects of "typical application requirements" and "core technologies required for meeting the application requirements". One of the conclusions was that MANETs, P2P, and sensor networks have a lot of common additional requirements beyond standard database technology, but they are used in very different scenarios, which have different requirements.

Additionally, there were four inspiring overview talks on data management in sensor networks, middleware approaches for sensor networks, personalization for the wireless user, and location-based services, which were followed by interesting discussions on further research directions in the whole group.

Finally, a couple of spontaneous ad-hoc organized discussions deepened some aspects of the workshop, for example, a discussion on a comparison of specific atomic commit protocols for MANETs, a discussion on data compression techniques, as well as more general discussions like how to teach the major results in the field.

Altogether, the seminar offered an excellent opportunity to share experiences and compare different contributions to data management in evolving networks from all the above areas, i.e. MANETs, P2P and sensor networks. We are convinced that a better understanding of problems that are common to different areas of data management in evolving networks has been achieved, and that this will result in synergetic effects in further research. All in all, the seminar was very successful and offered researchers of various disciplines from different countries an opportunity to share their experiences and expertise in a setting that is, as always, both inspiring for deep thoughts and pleasurable for idea exchanges.

A Perspectives Workshop where similar topics were discussed was held in Dagstuhl in March 2005: "Peer-to-Peer Mobile Ad Hoc Networks - New Research Issues", Organizers: M. Gerla (Univ. California - Los Angeles, US), C. Lindemann (Universität Dortmund, DE), A. Rowstron (Microsoft Research, GB).

8.7 Naming and Addressing for Next-Generation Internetworks

Seminar No. **06441** Date **29.10.–01.11.2006**
Organizers: B. Ahlgren, L. Eggert, A. Feldmann, A. Gurtov, T.R. Henderson

The basic principles of the original Internet architecture included end-to-end addressing, global routeability, and a single namespace of IP addresses that could serve simultaneously as locators and host identifiers. A second namespace of fully qualified domain names was later added, and a domain name system (DNS) was developed to map between such names and addresses. Additionally, over time, a number of additional namespaces have emerged, many of which include some component of domain names and are also served by DNS. DNS has used caching to scale well, and consequently is not optimized for rapid updating of records.

However, due to the growing trend towards mobility of users, terminals or even whole networks, dynamic naming structures are gradually replacing the static mechanisms of the traditional Internet. Meanwhile, the emergence of network address translation (NAT) has clouded the end-to-end significance of IP addresses. Moreover, during the evolution of the Internet from a small research network to a worldwide information exchange, a growing diversity of commercial, social, ethnic, and governmental interests has led to increasingly conflicting requirements among the competing stakeholders. These conflicts create tensions that the original Internet architecture struggles to withstand. Clark et al. refer to this development as “tussles in cyberspace”.

This evolution has prompted research into different internetworking architectures, such as FARA, Plutarch, Triad, i3, SNF, TurfNet, DOA, IPNL and 4+4, among others. At the core of these next-generation network architectures are naming and addressing frameworks that are significantly more flexible, expressive, and comprehensive than the Internet’s hierarchical IP address space. These naming frameworks are key components that enable advanced internetworking capabilities, such as multi-homed mobility, dynamic composition of networks, or delay and disruption-tolerant communication.

The naming architectures of these new internetworking architectures frequently have a more formal framework for naming than do current networks. Many naming architectures provide dynamic bindings between the levels of names and objects. With dynamic bindings at multiple levels, names of objects can become location independent and support different types of mobility, e.g., nodes or services. Some naming architectures also support the notion of indirection or delegation.

To discuss these issues and to advance this field of research, a seminar on naming and addressing for next generation internetworks was held at the Schloß Dagstuhl from October 29 to November 1, 2006 . Researchers from different fields discussed their views and recent results pertaining to naming and addressing problems related to the seminar topic. This article briefly reviews their presentations and discussions, as well as the research questions identified and debated at the seminar.

The seminar brought together a diverse community of researchers from academia and industry, with different research interests including network theory, mobile networks, interdomain routing, networking in challenging environments, privacy, deployment, and peer-to-peer aspects of networking.

Conclusions

The contributions and discussions echoed many similar concepts during the workshop, including cryptographic names, flat names, search as a naming mechanism, DTNs, DHTs, etc. Hence, an important question is: “Are we all working on similar things and is the packaging and terminology an issue?” It appears that a commonly accepted terminology is missing for naming and addressing. There have been several attempts to define a common terminology for addressing and naming, but the results of the workshop suggest that those approaches are not sufficient and more general evaluation of the terminology problems is needed. To start with, a clear problem description for naming and addressing issues would help. For example, it is not clear whether the problems are more about performance and efficiency or about providing new capabilities. Furthermore, additional research has to evaluate which features and characteristics the current APIs are missing that hinder users from offering services with more flexible naming.

Part of the problem with naming seems to be the lack of hard requirements, stemming, among others, from a fuzzy understanding of the naming and addressing infrastructure users. Hence, there is a clear need to better define the problem space and resulting requirements. Questions like “what is being named?”, “what are connectivity properties of the network?”, “what must it inter-operate with?”, “what are the service models?”, and “what infra-structure is needed?” have to be addressed. Participants would also like to see boundaries and tradeoffs in a more analytical and rigorous manner, taking aspects like security, heterogeneity, and mobility into account.

But where are the specific problems in generalizing naming or in finding axiomatic approaches to naming? Routing seems to be one problem, as it is hard to separate from naming and addressing. Furthermore, there are three different groups of stakeholders – operators, terminals and users/services – and it seems to be hard to find a common ground between them. A general understanding of the specific problems is still missing, though. Mobility is a good example; it is an intensively studied aspect in the context of naming and addressing. Yet, it is not clear if the lack of a clear solution for IP mobility is a result of the architecture or a fundamental problem with mobility.

With respect to some of the perceived fundamental problems, there are nevertheless some promising, but not fully tested solutions, such as HIP. Hence, future work should focus more on deployment and implementation to gain more experience with those solutions.

Moreover, short-lived labels and multiple coexisting namespaces are promising areas that should be further investigated. If name-spaces are added to the current architecture, implications on the search, resolution, security, and routing mechanisms have to be considered, resulting in additional engineering and operating costs that have to be taken into account as well.

Current approaches place lots of focus on the network layer. The strong impact that Google and search engines in general have had on the way that people think about names and networking APIs implies that we may need to be more radical in thinking beyond the network layer.

Participants agreed that a follow-up workshop in about 18 months to discuss those topics and other advances would be useful.

Chapter 9

Modelling, Simulation, Scheduling

9.1 Simulation and Verification of Dynamic Systems

Seminar No. **06161** Date **17.04.–22.04.2006**
Organizers: D.M. Nicol, C. Priami, H.R. Nielson, H.Ruess, A.M. Uhrmacher

Simulation is widely used for modeling engineering artifacts and natural phenomena to gain insight into the operation of those systems. Simulation, using a simulator or otherwise experimenting with fictitious situation can show the eventual real effects of some possible conditions. In contrast, formal verification is concerned with proving or disproving the correctness of a system with respect to a certain property, using mathematical and logical methods. Verification techniques include explicit-state enumeration, symbolic simulation, model checking, static program analysis, and theorem proving techniques.

Despite of these different objectives, the fields of simulation and verification address similar research challenges. In particular, the need for identifying and defining suitable models of the dynamic system under study unifies both research fields. Modeling methods have a significant impact on how easily certain phenomena can be described, they influence the acceptance in the application community and the possibilities to be analyzed and simulated. They formed one focus of discussion and the basis for a working group exploring the potentials and limitations of different modeling approaches. Although there are disparate approaches in the fields of simulation and verification for validating timed, probabilistic, and hybrid systems, both fields address component-based and abstraction-based validation techniques. Refinement and abstraction plays a crucial role, both in simulation and verification, but even more so if both approaches are combined. The role of refinement and abstractions has been a second focus of general discussion intensified in a dedicated working group. In application areas like embedded systems and systems biology, researchers of simulation and verification are coming together to see a lot of common problems and interesting solutions that help propelling research in the respective areas. During the seminar, the application area of Systems Biology moved quickly into the focus of interest, which resulted in a working group addressing the question whether the application area of systems biology requires specific modeling, simulation, and verification tools, and how biological systems differ from engineered ones.

The dialog between the simulation and verification community about the abovementioned issues most notably started with the Computational Methods in Systems Biology Workshop series, the first of which took place in Trento in 2003. The Dagstuhl Seminar continued and intensified this dialog between both communities working on simulating and verifying dynamic systems.

Seminar

Dagstuhl is dedicated to working groups. In contrast to traditional conference settings, the schedule offered plenty of time for working groups, discussions, and spontaneous activities. To give an overview about the different areas, state-of-the-art plenary talks were given in the beginning of the seminar. Short presentations provided the opportunity for each participant to present and discuss his or her own work during the morning session. The afternoons were dedicated to working groups. The themes of the working groups formed during the first days of the seminar. In the evening, the results of the working groups were presented in plenary sessions.

Intertwining working groups and plenary sessions helped to work on concrete questions in the different groups and to support a cross fertilization among them. The seminar was a truly interdisciplinary event and all participants played an active role in driving the progress and content of the seminar and the individual working groups. Their results took shape in three documents that formed the basis of the working groups' report: The challenge of combining simulation and verification. As always, Schloss Dagstuhl and its ambiance, its unusual blend of the old with the new, the organization, and the very helpful staff contributed largely to the success of the seminar.

Chapter 10

Mathematics

10.1 Reliable Implementation of Real Number Algorithms: Theory and Practice

Seminar No. **06021** Date **08.01.–13.01.2006**
Organizers: P. Hertling, Ch.M. Hoffmann, W. Luther, N. Revol

Real numbers are objects containing an infinite amount of information. Therefore, they cannot be represented precisely in a computer. This leads to well known problems caused by unverified finite precision implementations of real number algorithms.

There are several scientific communities, not only in mathematics but also in computer science, that are concerned with reliable real number algorithms. Computable analysis is a fast growing subdiscipline of theoretical computer science that analyses real number computation problems in the context of the Turing machine model. Another theoretical approach is domain theory. Here one of the goals is to lay foundations for a programming language for exact real arithmetic. There are many approaches that deal with the reliable implementation of real number algorithms from a practical point of view. The basic idea of interval arithmetic, to start with, is to compute with intervals that are known to contain the real number in question. It is striking that the space of intervals is a special case of a domain. Other, in many cases related approaches are Taylor models, high precision software, exact arithmetic, algorithms using result verification, symbolic representations of part of the data, algebraic computation schemes, perturbation schemes, and many more. These approaches are being applied in order to solve numerical as well as geometric problems. In computational geometry the special problem of implementing real number algorithms reliably is complicated by the interplay of numerical predicates and hidden dependencies between them that arise from geometry theorems that may not be known. This creates opportunities for inconsistent decisions that lead to faulty data structures and, ultimately, to failure of the computation.

It was the goal of the seminar to bring together people who are dealing with the reliable implementation of real number algorithms either from a theoretical or from a practical point of view and to stimulate an exchange of ideas between the different communities that will bring an advance for the reliable solution of numerical and geometric problems. Some particular goals of the seminar were:

- to point out the most urgent current practical problems in the implementation of real number algorithms, to analyze them using the tools and notions from topology, computability theory and complexity theory, and hopefully to understand them better and make progress towards a solution,
- to explore the practical aspects of the computability and complexity notions for various types of continuous objects, based on various topologies and the resulting types of information and of representation methods that are used in order to describe the objects in an approximating way,
- to analyze and compare the various software tools for reliable implementation of real number algorithms, to analyze and compare their advantages and limits, and to explore the need and the possibilities to develop further software tools specially suited for developing and implementing reliable algorithms over the real numbers,
- to integrate reliable functions and algorithms into computer algebra systems as well as recent modeling and simulation software.

The Seminar

Forty eight researchers from many different disciplines attended the seminar: people concerned with constructive mathematics or logic, with computability theory or complexity theory over the real numbers, with interval arithmetic, with robustness problems in computational geometry or solid modeling, with computer arithmetic, and with software for fast, arbitrarily high precision computations. The program consisted of 35 talks of 30 minutes each, and of three evening sessions with additional presentations and discussions. Many presentations showed that there are already strong interconnections between various disciplines. There were also lively discussions about different theoretical models and practical approaches for reliable real number computations.

Topics of the Seminar

- Constructive Mathematics and Computability Theory over the Real Numbers
- Complexity Theory over the Real Numbers and Software for Fast, Arbitrarily High Precision Computation
- Computational Geometry and Solid Modeling, Robustness Problems
- Interval Arithmetic and Software Systems for Reliable Computations
- to integrate reliable functions and algorithms into computer algebra systems as well as recent modeling and simulation software.
- Floating Point Arithmetic, Verification of Software

10.2 Challenges in Symbolic Computation Software

Seminar No. **06271** Date **02.07.–07.07.2006**
Organizers: W. Decker, M. Dewar, E. Kaltofen, S. Watt

Symbolic computation software allows mathematicians, scientists, engineers, or educators to deal with elaborate calculations using a computer. The applications range from introducing the experimental method in fields of pure mathematics to practical applications, for instance, in cryptology, robotics, or signal theory. The software includes mainstream commercial products such as Maple or Mathematica and highly specialized, public domain systems such as CoCoa, Macaulay2, or Singular.

Symbolic computation software implements a variety of sophisticated algorithms on polynomials, matrices, combinatorial structures, and other mathematical objects in a multitude of different dense, sparse, or implicit (black box) representations.

The subject of the seminar was innovation in algorithms and software, bringing algorithm designers, software builders, and software users together. The program consisted of 21 talks of 45 minutes, 3 panel discussions and a presentation of the "Oberwolfach references on mathematical software" (G.-M. Greuel). Supplementing the material directly related to the talks and discusssions, the proceedings contain additional contributions by J. Abbott ("Challenges in computational commutative algebra"), A. Storjohann ("Notes on computing minimal approximant bases"), and Stephen M. Watt ("Pivot-Free Block Matrix Inversion").

Well-known fundamental computer algebra algorithms include Buchberger's Groebner basis algorithm (adressed in the talks by A. Frühbis-Krüger, S. Laplagne, M. Noro), algorithms for system solving via triangular systems (X. Dahan, J.-G. Dumas), for linear algebra problems (E. Schost, A. Storjohann), for sparse interpolation (W.-S. Lee), for GCD computations and polynomial factorization (E. Kaltofen, S.M. Watt, L. Zhi), Fast Fourier Transformation algorithms (J. Johnson), and algorithms for dealing with differential and difference polynomials (X.-S. Gao, M.M. Maza). The variety of problems adressed in the different talks ranged from improving one of the basic algorithms (for instance, Noro) to discussing sophisticated algorithms based on the fundamental algorithms, such as Villamayor's algorithm for resolving singularities which made an algebraic geometers dream come true (Frühbis-Krüger). Also treated were symbolic-numerical methods (Lee, Zhi) and purely numerical approaches based on algebraic ideas (T. Ashby). Various implementation details were given.

A number of talks dealt with system building in all its different flavors, adressing basic problems related to creating high-performance software code and high-level interfaces (J. Abbott, Johnson, R. Rioboo), pen based systems and computer algebra (G. Labahn), and the GNU TeXmacs platform and computer algebra (J. van Hoeven).

Particular applications discussed include applications to real algebraic geometry (Lazard) and to problems arising from the integration of rational functions (P. Paule).

The topics of the panel discussions were "Visions for computer algebra in five years", "What can't, but should be done by computer algebra", and "What will be the next killer application of computer algebra". These discussions also adressed "political" questions such as:

- How to transfer computer algebra knowledge into application areas?
- How to create computer algebra tools for people below university level?
- How to make computer algebra systems more user-friendly?
- How to evaluate the quality of work done in computer algebra?
- How can the computer algebra community and the numerical analysis community be linked?

These themes also entered the lively discussions taking place between the lectures and in the evenings.

10.3 Computational Structures for Modelling Space, Time and Causality

Seminar No. **06341** Date **20.08.–25.08.2006**
Organizers: R. Kopperman, P. Panangaden, M.B. Smyth, D. Spreen

Topological notions and methods have been successfully applied in various areas of computer science. Image processing, programming language semantics and exact computing with real numbers and vectors in Banach spaces are important examples. Computerized geometric constructions have many applications in engineering and physics. The seminar concentrated on computational structures for modelling space, time and causality, which are basic in these applications. Special emphasis was given to connections with physics.

Due to the digital nature of computation, such structures differ from the mathematical structures they model, based on the continuum, that are classically used in these fields. Their typical features include a graph-based digital framework useful in computing algorithms, and also feature asymmetry and partiality. The classical spaces contain only the ideal elements that are the result of a completed computation (approximation) process which involves algorithms based on moving between points (for which a graph structure is used). But spaces that also allow reasoning about such processes in a formal way must contain the partial (and finite) objects appearing during a computation as well, and must consider a limiting process. Only the partial and finite objects can be observed in finite time. The leading example of such a structure is the domain (in Scott's sense). Here, the finitely observable properties of a process are the primary objects of study. The ideal entities which are the only elements considered in classical mathematical structures are obtained as derived objects via the limiting relationship. By a continuous model of a classical space we usually mean a domain, perhaps with additional structure such as a measurement or partial metric to represent the original space, as the subspace of maximal points. This gives a handle on the computational aspects of the classical space.

This, from a computational perspective, is some of the motivation for developing alternative models, in which a partial ordering (of the approximation of ideal elements by partial or finite ones) is fundamental.

What is remarkable is that very similar order-theoretic models are being developed for (apparently) entirely different reasons in theoretical physics.

The singlarity theorems e.g. show that in classical general relativity the basic geometric assumptions break down (singularities develop) so one seeks alternatives: Ashtekar's quantum geometry, string theory or Sorkin's causets.

Rafael Sorkin and his collaborator in combinatorics, Graham Brightwell, working in a program towards quantum gravity where the causal structure is taken as fundamental, use causal sets as basic structure, which are nothing more than locally finite partial orders. Keye Martin and Prakash Panangaden showed that globally hyperbolic spacetimes (studied in Kronheimer and Penrose's 1967 classic, "On the structure of causal spaces", Proc. Camb. Phil. Soc. 63, 481–501) are special continuous domains.

There are several consequences of the work of Martin and Panangaden. The topology of the spacetime manifold can be reconstructed from the causal structure; indeed from a countable dense subset of the spacetime. The result relates the areas of domain theory in computing and causality in physics, and provides new tools for deriving results relevant to quantum gravity, but it is only a beginning and much needs to be done.

In most work in physics of the kind just mentioned, one views space (or space-time) as a continuous manifold. But by using domains, we gain a clearer view of ideas derived from computer science being applied in the direction of physics.

There are reasons for wanting to consider also discrete models of space and time. Philosophically quantum mechanics suggests that one should look to discrete structures rather than continuum structures. There are no experiments that can probe arbitrarily deeply into the structure of spacetime (as that would require unboundedly high energies) so there can never be any experimental support for a true continuum.

We can now compare causal sets and other event structures with process models in computer science, so that posets and graphs will figure extensively in"discrete" models.

After very successful predecessor seminars in 2000, 2002 and 2004, the seminar in 2006 was the fourth in this series of Dagstuhl seminars which aim to bring together people working in fields like domain theory, topology, geometry, formal topology, and now causal spaces in physics, and to foster interaction between them. A further goal has always been to encourage communication and, hopefully, collaboration between computer scientists and those mathematicians and now physicists who work on similar problems but from a different perspective and who, often, are not aware of what their work has in common.

We are actively looking for people in more fields that involve related ideas of digital approximation of continuous structures.

This time the seminar attracted 49 participants representing 16 countries and 5 continents, among them 8 young researchers working for their master or PhD. The atmosphere was very friendly, but discussions were most lively. During the breaks and until late at night, participants also gathered in smaller groups for continuing discussions, communicating new results and exchanging ideas. Again the seminar led to several new research contacts, collaborations, and at least one successful application for a new Dagstuhl seminar.

As with the seminars in 2000 and 2004, the participants are again invited to submit a full paper for a special issue of *Theoretical Computer Science.*

The great success of the seminar is not only due to the participants, but also to all the staff members, both in Saarbrücken and Dagstuhl, who always do a great job in making everything run in such an efficient and smooth way. Our thanks go to both groups!

Chapter 11

Data Bases, Information Retrieval

11.1 Spatial Data : Mining, Processing and Communicating

Seminar No. **06101** Date **05.03.–10.03.2006**
Organizers: J.-R. Sack, M. Sester, P. van Oosterom, M. Worboys

This workshop has been organized as a successor to four preceding ones. The major goal has been to bring together experts from digital cartography, spatial modelling, computational geometry and cognitive science to meet with professionals from data mining and data interpretation. This has lead to a fruitful exchange of different – but very close – disciplines and hopefully to the creation of new collaborations. The Dagstuhl seminar has not only posed R&D problems, but provided crucial incentives and directions shaping the entire field. The group of participants was diverse both w.r.t. their academic discipline and their professional background. Researchers and developers from within industry, government, and universities (senior and young) shared their latest topics, problems, doubts, and investigations.

Challenges

The importance of spatial data in our daily lives is rapidly increasing and so are the challenges and demands on the research and commercial communities to address the different facets of spatial data. In these communities, spatial data have generated tremendous interest over the last decade.

Interpretation of spatial constellations or situations is a process, that is closely linked to human capabilities and can be formalized using formal semantics (OWL, ODM, etc.). Making implicit information explicit is needed not only for many spatial analysis problems, but also for aspects of information communication: in digital cartography a "hot topic" today is adaptive visualization. A user is presented exactly the information that is needed for a specific purpose in a dedicated specific situation. This presumes that first of all such a situation is identified, and secondly an adaptive presentation is generated from it. A typical adaptation is made according to the scale or resolution of the data – the traditional generalization problem. However, adaptation can be interpreted in a much wider way by also adapting according to the personal profile, behaviour and wishes of the user. Such adaptive representations are especially important in the context of mobile or wireless GIS, where the spatial information has to be transmitted via possibly limited bandwidth channels.

Spatial data also pose exciting questions for the algorithms and data structuring communities. It is vital that computational geometrists meet with the spatial data community to exchange ideas, pose problems and offer solutions. Most algorithmic problems arising in that field are indeed geometric.

Many different application areas arise from the general availability of spatial data, e.g. using data in mobile applications, integrating data for complex tasks (e.g. traffic monitoring, risk management), inferring behaviour patterns from data sets. The problems relating to data interpretation and data mining with respect to these applications have been discussed in the seminar. Other integrated topics are ubiquitous spatial processing and formal spatial semantics.

Outcomes

Outcomes of the seminar include a collection of abstracts, presentations (slides) and some papers surveying the current state of the art in this field and latest research initi-atives (available on the website `http://www.dagstuhl.de/06101/Materials/`). Similar to the previous seminar on ‘computational cartography and spatial modeling’, it is expected that new partnerships and collaborations between multi-disciplinary groups (reinforced and established during the current seminar) will further advance this field with the inclusion of emerging topics.

Another important result of the seminar is the ‘Challenges of GIScience - green + red topic list’, which can be found on the seminar’s website. The idea was to identify topics for the next 5 years that are worth and challenging to work on – and at the same time also state, which topics should not be treated.

11.2 Content-Based Retrieval

Seminar No. **06171** Date **23.04.–28.04.2006**
Organizers: T. Crawford, R. Veltkamp

Images, music, video, and 3D scenes play a crucial role in Visual Information Systems and Multimedia. There is an extraordinary number of applications of such systems in entertainment, business, art, engineering, and science. Such applications often involve huge collections of media, so that efficient and effective searching in databases of these media is an important operation.

The emphasis of this fourth seminar lies on the PERCEPTUAL and COGNITIVE aspects of all kinds of content-based retrieval. Fundamental questions such as the role of perception and cognition in feature extraction, pattern similarity rating, indexing large collections etc. play an important role.

We strongly believe that content based retrieval needs an integrated approach from fields such as image processing, shape processing, psychology, database indexing, visualization, querying, etc. The purpose of this seminar is to bring together people from the various fields in order to promote information exchange and interaction among researchers who are interested in various aspects of accessing the content of images, music, video, and 3D data.

For this seminar, we have invited internationally known as well as young researchers from various disciplines with a common interest in content-based multimedia retrieval. We have been together with a group of 26 researchers for a week, away from the rest of the world, and certainly good interaction and exchange of ideas took place during the sessions as well as in the very "gemütliche" wine cellar, enjoying the cheese platter.

There was a total of 26 presentations, a demonstration session, and a discussion session. The discussion session was about the challenges we face in the coming phases of research in content-based retrieval; to a large extent this discussion was about common problems, especially in the difficulty of evaluation, across all the domains represented at this seminar — we hope that some common solutions will present themselves before too long. The presentations in this seminar can be grouped thematically as follows:

- At a cross-media level, there were presentations about indexing and web issues
- On 3D model retrieval related topics
- The presentations related to music retrieval
- A number of presentations were on video and motion retrieval
- The other presentations were largely in the area of image retrieval

11.3 XQuery Implementation Paradigms

Seminar No. **06472** Date **19.11.–22.11.2006**
Organizers: P.A. Boncz, T. Grust, J. Siméon, M. van Keulen

Only a couple of weeks after the participants of seminar no. 06472 met in Dagstuhl, the W3C published the Final Recommendation documents that fix the XQuery 1.0 syntax, data model, formal semantics, built-in function library and the interaction with the XML Schema Recommendations (see W3C's XQuery web site at `http://www.w3.org/XML/Query/`). With the language's standardization nearing its end and now finally in place, the many efforts to construct correct, complete, and efficent implementations of

XQuery finally got rid of the hindering “moving target” syndrome. This Dagstuhl seminar on the different XQuery implementation paradigms that have emerged in the recent past, thus was as timely as it could have possibly been.

From the beginning, XQuery has been designed as a declarative language in the style of modern functional programming languages. For the query author, declarativity means that the formulation of queries solely depends on the desired input and output – efficency concerns should not have any impact at all. For XQuery implementations, declarativity provides a sheer endless pool of alternative strategies to consume and represent data model instances as well as to compile, optimize, and execute queries. In principle, all of these strategies are acceptable as long as they respect the language’s formal semantics.

This freedom has led to a plethora of, sometimes radically different, approaches to the implementation of XQuery. It is characteristic for most of the implementation projects in this “zoo”, that a specific set of XQuery features drove their initial development, e.g., the evaluation of XPath location steps or the efficient implementation of nested FLWOR expressions and the derivation of equivalent database-style join strategies. To this end, our colleagues out in the field applied existing techniques and devised new approaches rooted in the programming language and database query language domains. Still, XQuery implementations which excel in both, completeness and efficiency, are rare (if available at all) today.

It was the foremost goal of this seminar to bring together a vivid group of academic and industrial researchers who are representatives of the distinct implementation camps that can be currently found in the XQuery landscape. In particular, the organizers tried to make sure that the native, relational, and streaming implementation camps all had their fair share of participants. We are happy to report that a total of 31 colleagues found their way to Dagstuhl – in effect, for three days the castle saw a concentration of expertise in the XQuery language and its implementation that goes unmatched even when compared to the major global scientific conferences in the field.

Concluding Remarks and Future Plans

The functional nature of the XQuery language makes it particularly amenable to implementation techniques developed in the functional programming languages domain (this point was made by Kristoffer Rose, Philippe Michiels, Jérôme Siméon, Maurice van Keulen, and Torsten Grust). It is indeed perceivable to define faithful reformulations of the XQuery semantics in terms of combinator languages or variants of monad comprehensions, two expressions forms from which efficient database-style algebraic plans can be derived. A group of seminar participants will engage in an effort to further develop and study a (unified) algebraic representation for XQuery. Ideally, this will lead to interoperability between some of the many promising XQuery implementation efforts.

We hoped that the participants were prepared and willing to teach each other in a constructive fashion and we were lucky to find exactly that during the seminar days. Dagstuhl greatly helped to create an atmosphere in which the formerly separate camps collaboratively worked on the syntheses of proven XQuery compilation and evaluation techniques.

The organizers would like to sincerely thank the Dagstuhl Scientific Directorate of Dagstuhl castle and are looking forward to put forward a follow-up seminar proposal which will reflect the then current developments around the XQuery language. Quite possibly this will include XQuery 1.1, whose initial requirements analysis phase has started just as we write this, and the forthcoming XQuery Scripting Extension which will bring the worlds of functional XML querying and stateful programming even closer together.

11.4 Digital Historical Corpora – Architecture, Annotation, and Retrieval

Seminar No. **06491** Date **03.12.–08.12.2006**
Organizers: L. Burnard, M. Dobreva, N. Fuhr, A. Lüdeling

The seminar brought together scholars from (historical) linguistics, (historical) philology, computational linguistics and computer science who work with collections of historical texts. These texts or digital libraries or corpora are collected for a number of different purposes such as lexicography, history, linguistics, philology etc. This, naturally, leads to different decisions in their design and architecture.

The purpose of this seminar was twofold: First we wanted to inform each other about the decisions each of us had taken in building a historical corpus and discuss the options. Second, we wanted to build an international network of people working with historical corpora and explore the options for further partnerships or projects. We think that both goals were reached.

The seminar was very interesting and stimulating. In the final discussion of the workshop, a 'grand picture' of the research issues in the area of digital historic corpora was developed. A researcher's workbench should support personalization, collaboration as well as problem solving. It must be complemented by tools for the annotation and the analysis of corpora, as well as providing functions for visualization, browsing and retrieval (especially for spelling variants). These methods should first be applied to and tested on small corpora, before they can be used for large corpora. In this context, evaluation also plays a major role. For large corpora (stored in digital libraries), the choice of an appropriate architecture is a crucial issue.

Another issue that was of interest to all participants is quality control and standardization.

Chapter 12

Machine Learning

12.1 Combinatorial and Algorithmic Foundations of Pattern and Association Discovery

Seminar No. **06201** Date **14.05.–19.05.2006**
Organizers: R. Ahlswede, A. Apostolico, V.I. Levenshtein

The focus of this seminar has been on the completely new scenario and on the wild paradigm shift that are forced by the recent progresses of ICT (information and communication technology). The new scenario is that data and information accumulate at a pace that makes it no longer fit for direct human inspection. The paradigm shift is that, in contrast to a primeval, persistent tenet of traditional information science and technology, the bottleneck in communication is no longer represented by the channel or medium but rather by the limited perceptual bandwidth of the final user: more and more often, the time and resources that need to be invested in order to gain access to information happens to be disproportionate to fruition time and value, thereby defying the very purpose of access. Consequently, the challenge of maximizing the throughput to the final user has taken up entirely new meanings. The implications brought about by such a dramatic change in perspectives have barely begun to be perceived. A science and engineering of discovery is developing to meet these challenges, which promises to revolutionize many facets of human activity beginning with the basic notions and practices of scientific investigation itself.

Above all, the problem of data overload looms ominously ahead in almost every field of our society. Databases in the Tera byte, even Peta byte range are now not uncommon. Our ability to analyze and understand massive datasets lags far behind our ability to gather and store the data with the ever advancing computer technology. Thus, as unprecedented volumes of information are amassed, disseminated and shared at an increasing pace in the emerging information infrastructures, the effective access to, and manipulation of information will depend no longer only on the efficiency with which information itself is structured, compressed, transmitted, stored and retrieved. A new generation of computational techniques and tools is required to support the extraction and the discovery of useful knowledge from the rapidly growing volumes of data. Raw data is rarely of direct benefit.

Its true value is reflected by our ability to extract information useful for decision support or for exploration and understanding of the phenomena exhibited in the data source.

Huge amounts of scientific and social data are being produced and some have been made public in various databases or have been rendered commercially available. These data include experimental/observational data in Physics and Chemistry, DNA and amino acid sequences in Biology, Marketing data, financial data, etc. Thus the scope of data ranges from the microscopic world as to the global and cosmic world. Facing with these "data with hidden values", however, the current status of technology for discovering new scientific laws and knowledge useful for decision making is still immature. As said, a new era of challenges is opening with knowledge discovery technology in most areas in sciences and social activities. Our aim is to develop formal and practical methods for knowledge discovery from large compilations of data in various areas, and simultaneously, systematize the methods so far developed and applied in practical fields toward a creation of knowledge discovery paradigms. The task of analyzing data to extract useful information behind it is becoming more and more difficult because of the huge volume of data and limitations in computational resources.

At some core level in these endeavors, it comes natural to identify the need for novel techniques supporting the automated discovery of patterns and their associations or "rules" in disparate contexts and media. The techniques developed along these lines find ad hoc incarnations in diverse fields but also feature a distinctively unifying flavor. For instance, searching for identical or similar substrings in strings is of paramount interest to software development and maintenance, philology or plagiarism detection in the humanities, inference of common ancestries in molecular genetics, comparison of geological evolutions, stereo matching for robot vision, etc. Checking the equivalence (e.g. identity up to a rotation) of circular strings finds use in determining the homology of organisms with circular genomes, comparing closed curves in computer vision, establishing the equivalence of polygons in computer graphics, etc. Finding repeated patterns, symmetries and cadences in strings is of interest to data compression, detection of recurrent events in symbolic dynamics, genome studies, intrusion detection in distributed computer systems, etc. The techniques for these problems have coalesced into an established core of Optimization, Pattern Matching, and other specialties of Algorithmics.

It is therefore a worthwhile effort to try and extract from the application areas crisp formulation of primitives, and study them in a coordinated fashion. Both theory and practice benefit from such an experience, as an increased degree of awareness and unification is fed back to both sides. This seminar thus concentrated on combinatorial and algorithmic techniques of pattern matching and pattern discovery that are regarded as the enabling machinery for such a revolution.

Chapter 13

Evolutionary Algorithms

13.1 Theory of Evolutionary Algorithms

Seminar No. **06061** Date **05.02.–10.02.2006**
Organizers: D. V. Arnold, T. Jansen, J. E. Rowe, and M. D. Vose

The 2006 Dagstuhl Seminar "Theory of Evolutionary Algorithms" carried forward a series of Dagstuhl seminars that started in 2000 and has become an established event in the community. In the week from from 05.02.2006 to 10.02.2006, 56 researchers from 12 countries discussed their recent work and recent trends in evolutionary computation.

Evolutionary algorithms (EAs) are randomized search heuristics. Introduced in the 60s and having come to great popularity in the 80s, they have been applied successfully in many different areas. Borrowing ideas from natural evolution, these algorithms operate on a population (multiset) of candidate solutions to a task. Promising candidates are selected from the population based on their fitness (objective function value) to become parents. Offspring are generated as variants of parents by means of stochastic crossover and mutation operators. The population for the next generation is chosen from parents and offspring, and the process then repeats until some stopping criterion is met. A wealth of different EAs have been developed which vary and embellish this basic theme, including the use of stochastic operators adapted to the search space and task, sophisticated methods for adapting parameters during a run, co-evolutionary paradigms, and explicit estimation of distributions.

Evolutionary algorithm theory is three decades old, but only recently has theory migrated to provably correct foundations. In addition, the vast majority of practitioners are collectively generating EA variants at a rate which far outpaces the speed with which the comparatively few theoreticians can analyze them. Making the situation worse, what most practitioners say they want — an inexpensive answer for how best to optimize — is in general impossible to achieve. All of this has contributed to a chasm between proven theoretical results and evolutionary algorithms as applied in practice.

Evolutionary algorithm theory is comprised of diverse approaches from various perspectives having differing objectives. Facet-wise analyses concentrating on one-step behavior of EAs (schema theory being the best known approach of this kind), analyses based on Markov chains, infinite population models, heuristic analysis borrowing ideas from statistical mechanics, run time analysis in the spirit of the analysis of randomized algorithms, and other approaches, have been developed separately and almost independently.

The 34 talks given during this seminar were organized in eight sessions that centered not around different approaches but central themes. Presenting different points of view and competing approaches to solve central open problems stimulated fruitful discussions. It became apparent that while different fields continue to contribute their methods and perspectives the central open questions are consistent.

The continuing stream of new variants of evolutionary algorithms was represented in two complete sessions, one dealing with co-evolutionary algorithms, the other centered around estimation of distribution algorithms. While both developments are recent in comparison to standard evolutionary algorithms, the presentations revealed that theory is able to adapt to such new developments.

An open discussion session on Wednesday night turned out to become a forum for a lively and controversial discussion about the course of FOGA ("Foundations of Genetic Algorithms"), the other important bi-annual event focusing on the theory of evolutionary algorithms.

One of the most central and important issues in evolutionary computation theory is the way such algorithms solve optimization problems and, in particular, the role crossover plays in solving such problems. In one of the last talks of the seminar, Riccardo Poli presented work that was partially developed during the week in Dagstuhl. He presented an example function, called OneMix, where on the one hand his specific perspective on the issue becomes concrete and, on the other, other approaches can deal with a concrete example where different explanatory statements can be presented and compared.

A secondary focus is more forward looking. While making significant progress with the development of analytical methods and the achievement of results within the different areas of EA theory, there are problems within areas that have been identified early, but have so far been too difficult to be solved from within the areas in which they arose. In some instances there may be hope that the application of methods from a different area of EA theory will prove useful in the near term. This may contribute to a longer term unification of separate theoretical approaches.

Another important aspect is the connection to other fields of theoretical computer science. Since EAs share some properties with other randomized search heuristics, analytical approaches developed within EA theory may be useful there, too. One example is the analysis of simulated annealing.