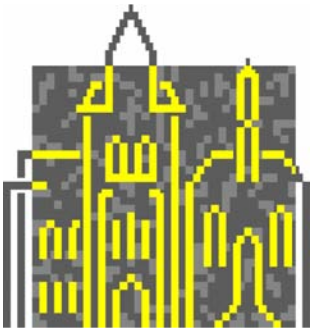

SCHLOSS DAGSTUHL

INTERNATIONAL
CONFERENCE
AND RESEARCH CENTER
FOR COMPUTER SCIENCE

Dagstuhl News

January - December 2003

Volume 6
2004

ISSN 1438-7581

Period: January - December 2003

Frequency: 1 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.

Associates: Gesellschaft für Informatik e.V., Bonn
Technische Universität Darmstadt
Universität Frankfurt
Universität Kaiserslautern
Universität Karlsruhe
Universität Stuttgart
Universität Trier
Universität des Saarlandes

Funding: The state governments of Saarland and Rhineland Palatinate

Address: IBFI Schloß Dagstuhl
Octaviealle
D-66687 Wadern
Tel.: +49 - 6871 - 905127
Fax: +49 - 6871 - 905130
E-mail: service@dagstuhl.de
Internet: http://www.dagstuhl.de/

Welcome

You have in your hands the sixth edition of the "Dagstuhl News", a publication for the members of the Foundation "Informatikzentrum Schloss Dagstuhl", the *Dagstuhl Foundation* for short. As always, we are a bit late, which as always has its reasons in the fact that Dagstuhl keeps us busy. Let me right from the start tell you why we are busy. We have experienced increasing numbers of guests and overnight stays during the last couple of years. We are approaching the capacity limits and look for ways to squeeze more rooms into the building and extend seminars into the weekend. At the same time, submissions of proposals have increased to about twice as many as we used to have. Of course, we are happy about these figures as they can be interpreted as proving the unbroken popularity of Dagstuhl. On the other hand, the scheduling problem becomes more and more difficult.

The decision by the Federal-State Commission (Bund-Länder Kommission) to move Dagstuhl onto the *Blue List* of research institutions with combined federal and state funding also throws some shadows of increased bureaucratic efforts onto our staff.

The main part of this volume consists of collected resumees from the Dagstuhl Seminar Reports. 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 2003 are on the Web under URL: `http://www.dagstuhl.de/Seminars/03/`

As I have told you last time, we are switching to publishing online proceedings of our Dagstuhl Seminars instead of the old Seminar Reports. 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. Our staff member Jutta Huhse is still working on making this project a success. It's not an easy job and needs some motivational work. Some exemplary reports can be found on our web pages.

The State and the Activities of the *Dagstuhl Foundation*

The foundation currently has 45 personal members and 7 institutional members.

In 2003, the foundation has supported a few guests with travel grants and a reduction of the Seminar fees. As usual, the supported guests did not have any budget for traveling expenses and could not be financed by Dagtuhl's normal budget. All supported guests were young researchers aged 20-30 years.

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, December 2004

Contents

Chapter 1

Data Structures, Algorithms, Complexity

1.1 Numerical Software with Result Verification

Seminar No. **03041** Date **19.01.–24.01.2003**
Organizers: R. Alt, A. Frommer, R.B. Kearfott, W. Lutter

Numerical computations are not reliable in the sense that rounding errors affect more or less every result of such a computation. Recently, several techniques have been developed to computationally circumvent these problems. The result of a computation then has the same rigour as a mathematical proof; it is therefore reliable and verified. The idea of the seminar was to bring together those who develop software for verified numerical computation and those who need such computations in their applications.

More than fifty scientists took part in this seminar. Each day started with a highlighted lecture on one of the main topics of the seminar. In order to stimulate interactions and to trigger discussions, participants were assigned to groups sharing a common subject: comprehensive software systems, libraries, enhanced software systems, object orientation, standardization, optimization, algorithms for verified numerical computation, novel approaches to validation, engineering and financial applications, applications in process simulation and control, applications in geometry and geodesics, applications in physics and chemistry.

During daytime, these groups presented their latest results in common sessions; this work was complemented by software demonstrations which took place in the evening.

For detailed information on all talks we refer to the abstracts in the proceedings as well as to the external home page of this conference which contains the slides of most of the lectures. At this point we just mention the major lines of discussion and development which became evident through this seminar.

Validated numerical computation is now supported by a variety of numerical software. The latest developments show that fast validated computation can be achieved, that the high precision evaluation of standard functions is still an exciting area of development, that the

integration with algebraic and symbolic computation becomes increasingly important and is supported more and more, and that techniques from compiler technology get used more and more in this area.

From the algorithmic point of view, the most impressive progress is being made in methods for global optimization, boosted by a European project. One of the strong points of the seminar was probably also the fact that many scientists from various application fields participated very actively. It became clear how validated numerical computation today enters such different areas like control theory, process simulation, mechanical reliability, robotics, chemistry, physics, geodesy and computational geometry.

Finally, it was interesting and stimulating to compare the different approaches to validation relying on interval arithmetic, stochastic arithmetic, static code analysis or artificial intelligence techniques like theorem proving.

1.2 The Propositional Satisfiability Problem – Algorithms and Lower Bounds

Seminar No. **03141** Date **30.03.–04.04.2003**
Organizers: A. Goerdt, P. Pudlak, U. Schöning, O. Watanabe

The propositional satisfiability problem is the basic problem for which efficient algorithms in the classical sense do not exist. However, theoretical and applied computer scientists are clearly interested in this problem. On the applied side the satisfiability problem is seen as a paradigmatic combinatorial search problem. It is a special type of constraint satisfaction problem. And constraint satisfaction problems allow for a natural modeling of real life search problems. On the theoretical side two complementary aspects of the satisfiability problem are the focus of recent research: First, developing algorithms with provable performance guarantees, and second, proving lower bounds of any kind. Recently scientific progress has been made in each of the aforementioned areas.

Due to the diversity of the techniques employed the corresponding scientific groups tend to be in part disjoint. It is the obvious purpose of the seminar to bring these groups together.

The seminar fulfilled its purpose in any respect. Most of the about 20 talks dealt directly with algorithmic aspects of the problem, most interestingly some experimental and theoretical analyses of local search algorithms were presented. Three talks from the applied area are also worth mentioning. In two of them satisfability instances arising from cryptographic applications were presented and one dealt with satisfiability instances arising form the area of configuration (of cars). These talks were particularly interesting to algorithm designers because they made them familiar with complex instances from real life. Experience will show, if this has served as a starting point of a fruitful collaboration.

The understanding of random propositional formulas in conjunctive normalform is still one of the major open problem areas. The relevant "satisfiability threshold conjecture" is based on an experimentally clearly visible phenomenon, but is is still only to a small

part proven by now. The conjecture asserts that formulas with approximately 4.2?n many randomly chosen 3-clauses become suddenly unsatisfiable.

A couple of open problems where discussed by the participants during an open problem session. These are contained in the proceedings.

1.3 Centennial Seminar on Kolmogorov Complexity and Applications

Seminar No. **03181** Date **27.04.–02.05.2003**
Organizers: B. Durand, L.A. Levin, W. Merkle, A. Shen, P. Vitanyi

Public Outreach

Algorithmic information theory (Kolmogorov complexity theory) measures the amount of information in a given finite object (bit string, file, message etc.) and formalizes the distinction between highly compressible objects that contain little information (regular objects) and incompressible objects with high information content (random objects). This idea was put forward in 1960's by several researchers, including the famous mathematician, Andrei Nikolaevich Kolmogorov, and led to fruitful developments. The seminar celebrating the 100th birthday anniversary of Kolmogorov, tried to gather the most active people in the field, including some disciples of Kolmogorov, for discussion.

Scientific Highlights

Several active fields of research were covered in the talks:

Relations between computational complexity and descriptional complexity. The idea of taking into account the computation time (needed for decompression) was clear already in the 1960's. However, only recently this connection became better understood and interesting relations between complexity classes and time-limited random (incompressible) objects were found. This development could be seen also as finding connections between different notions of randomness (randomness in algorithmic information theory, pseudo-random number generators etc.).

Starting with classical works of Martin-Löf, the notion of algorithmic randomness was closely related to measure theory. Recently it was noted that the classical notion of Hausdorff dimension (and similar notions) could be naturally translated to the algorithmic information theory using martingale technique and similar notions.

The first Kolmogorov paper on the subject was called "Three approaches to the definition of the notion of amount of information" and these approaches were named 'combinatorial', 'probabilistic' and 'algorithmic'. Recently some formal links between these three approaches were noted that allow us to translate some results of algorithmic information theory into combinatorial results and statements about Shannon entropy.

Last but not least there has been a recent development clarifying the distinction between "accidental" information (random noise) and "meaningful information" and how to separate the two. This is a central object of statistics and model selection.

Perspectives

Algorithmic information theory belongs to theoretical computer science and does not claim to be immediately applicable to practice (for example, there is no algorithm to compute Kolmogorov complexity of a given string). However, its ideas act as a sort of inspiration for quite practical applications in learning theory, pattern recognition etc. showing that deep theoretical research becomes useful unexpectedly often.

1.4 Fixed Parameter Algorithms

Seminar No. **03311** Date **27.07.–01.08.2003**
Organizers: M. Fellows, M. Hallett, R. Niedermeier, N. Nishimura

An Organic View of Computational Complexity

"How are we able to have this conversation?"

For one thing, the reader of this discussion is processing strings of symbols over an alphabet of 26 distinct kinds (not 10,000) and making new associations between three or four ideas concurrently (not 5000). Our natural intuitions about the complexity of information processing tell us that these relatively small parameters of the situation make a big difference in our ability to accomplish the task (of reading). Similarly, small structural parameters can make an enormous difference in the ability of computer algorithms to process information. These parameters may describe the number of tracks in the layout plan for a microcircuit, the number of genes in an evolutionary family, or the number of processors to be scheduled. Frequently these numbers are also in the range of 10 or 20 or 50 for realistic applications.

It's not just a matter of being clever enough. Some problems, such as factoring an integer into primes, appear to be intrinsically resistant to any kind of efficient information chemistry. (As lemonade can be made from lemons, this is actually useful: internet commerce via cryptographically secure communications depends on computational intractability, the impossibility of any efficient solution). The tragedy of the mathematical theory of computing is that there are thousands of natural and important computational problems that, like factoring, appear not to admit any efficient general means of solution. But wait —

What do we mean by "efficient"? In the theoretical framework for computer science that has emerged over the first few decades of this new discipline, the basic definitions are one-dimensional: attention is focused on the cost of information processing as a function solely of the *overall input length*. This makes sense if the input is random or arbitrary and does not have any hidden or implicit structure. But as we have suggested, we seem

to be able to read and write (and accomplish many other computational tasks) because the information processing problem involved is governed by parameters and structures of modest size, even though the total length of the input to be processed is much longer.

This is the main intuition and issue addressed by the subject of designing fixed-parameter algorithms in the parameterized complexity framework, which introduces a two-dimensional analysis, where one dimension (as classically) is the overall input length, and where the other dimension represents the restricted structure of the realistic situation (the relevant "problem parameter(s)"). The goal is to confine any explosive computational costs to a function only of the (relatively small) parameter(s). This relatively new research program is showing very wide-ranging successes in addressing in this way the computational challenges that face us, in confronting the thousands of computational problems that appear to be (one-dimensionally) quite hard.

Numerous recent working algorithms in computational biology and bio-informatics (e.g., genome and proteome analysis) are based on these new ideas about designing useful algorithms by exploiting natural problem parameters. Bio-informatics continues to be an area of exciting successes for the field. The Dagstuhl workshop brought together algorithms and complexity theorists, as well as implementors and applications-oriented researchers. We anticipate, from the collaborative connections made at Dagstuhl Castle, that the richly developing toolkit of mathematical ideas for designing and analyzing parameterized algorithms, will continue to move quickly into practical deployment.

Scientific Highlights

The seminar on Fixed-Parameter Algorithms brought together researchers from around the world to share their experiences in developing algorithms for a wide range of application areas (e.g. computational biology, graph theory, and motion planning) using diverse approaches. Many of the results presented at the workshop were improved algorithms for classic fixed-parameter problems such as Vertex Cover. The workshop brought to focus how new techniques, such as automated generation of search trees or crown rules and duality, can lead to "feasible in practice" fixed-parameter algorithms.

Several talks discussed implementations capable of solving these problems on graphs with more than 2000 vertices! The design of general schemes for the distributed computation of search trees, automating the identification of good reduction rules, amortized analysis of the behavior of fixed-parameter algorithms, and the relationship between approximation and fixed-parameter complexity were identified as strong areas of interest in the near future.

In order to take advantage of the diversity of expertise and to foster new research collaborations, a portion of the seminar time was set aside for active research, as detailed below in the section on training. At least one of the results presented at the meeting arose from new collaboration that started at the last Dagstuhl meeting on parameterized complexity, two years ago.

Sessions

In order to facilitate working relationships among senior and junior participants, we scheduled the talks to allow for extra active working sessions of two types.

In his opening session, Mike Fellows presented a series of challenges for the participants, both long-term and short-term ideas for future work. Two specific problems were identified for immediate work; the goal was to come up with the fastest possible fixed-parameter algorithms for packing of k disjoint three-stars (a problem known to have fixed-parameter algorithms) and for planar directed feedback vertex set (a problem whose status is still unresolved). Early in the week, we scheduled an afternoon session for work on the packing problem, where researchers shared their partial results and then broke into small groups for further progress.

During another session, researchers were asked to present open problems for further work by small groups. The problems ranged in application area as well as technique. Researchers then gathered in small groups to join forces in solving the problems; groups resulted in new working partners, senior researchers alongside junior ones, each contributing using different approaches. Further "cross-fertilization" took place in the summary meeting on the last day, where progress reports were made by all the groups. The impact of the seminar will be felt in years to come, as results are found and collaborations continue.

1.5 Graph Colorings

Seminar No. **03391** Date **21.09.–26.09.2003**
Organizers: J. Nesetril, G. Woeginger

The seminar was devoted to the most important recent developments in the area of graph colorings. A non-expert definition of graph coloring is the following: We want to color several objects with the smallest possible number of colors, subject to collision constraints that forbid that some pairs of objects receive the same color. The definition for experts is quite similar, but one has to replace the word "objects" by "vertices of a graph", and "collision constraints" by "edges". The basic graph coloring problem is computationally intractable (NP-hard), and for that reason the combinatorics of graph colorings is quite messy and complicated and hard to handle, and it leads to many fascinating questions.

Over the last three decades, researchers in Discrete Matematics, in Combinatorial Optimization, and in Theoretical Computer Science have spent considerable effort on understanding the combinatorics and the computational complexity of various graph coloring problems. There are many reasons for this.

- Graph colorings are ubiquitious in the modelling of real world applications. For instance, they show up as frequency assignment problems in telecommunication; they show up as machine assignment problems in production scheduling; they show up as register allocation problems in operating systems etc, etc, etc.

- Most graph coloring problems are very easy to formulate, very easy to grasp, and very difficult to solve. Some graph coloring problems constitute attractive puzzles.

- Graph coloring problems form a keystone in the testing of various algorithmic approaches, like local search approaches, genetic algorithms, simulated annealing, Markov chain approaches, and so on. Computationally, graph coloring problems belong to the most difficult problems. Hence, if an algorithmic approach works out well for graph colorings, we expect it to work out well for many other algorithmic problems as well.

- Graph colorings show up in an incredible variety of forms. Just to name a few: There are lambda-colorings (= colorings with a condition at distance two in frequency assignment); alpha-colorings, sub-colorings, list-colorings, f-colorings, precoloring extensions, colorings with forbidden subgraphs, graph homomorphisms, Ramsey colorings, role assignments, equitable colorings, etc etc etc. Also all kinds of morphisms from structures into structures fall into this area.

The Dagstuhl workshop on graph colorings was attended by 45 particpants with affiliations in 17 countries (Austria, Canada, Czech Republic, Denmark, England, France, Germany, Malta, Netherlands, Norway, Poland, Slovakia, Slovenia, Spain, Sweden, Switzerland, USA).

All in all there were 29 scientific presentations. We decided right in the beginning to have a so-called "liquid" schedule: There are no fixed time slots; every speaker is allowed to talk as long as s/he likes; if questions come up in the middle of the talk, then the speaker may switch topic and discuss these questions. For every day, we only fixed a rough list of speakers; this list was flexible, and sometimes we had to remove the last speaker of the day and make him the first speaker of the following day. We also moved the coffee-breaks a lot.

On Tuesday evening and on Wednesday evening, we had open problem sessions. We are currently collecting these open problems, and we will add the resulting open problem list to a special issue of the journal "Theoretical Computer Science" that will be devoted to the "2003 Dagstuhl Seminar on Graph Colorings". We expect this special issue to appear in spring or summer 2005.

1.6 Theoretical and Computational Aspects of Matrix Algorithms

Seminar No. **03421** Date **12.10.–17.10.2003**
Organizers: N. Higham, V. Mehrmann, S. Rump, D. Szyld

This seminar attracted forty-six participants from twelve countries. The main theme was matrix algorithms from several perspectives: computer science, information theory,

mathematics, engineering, physics, chemistry, statistics, algorithms, software, control, industrial applications. Many attendees praised the diversity of the presentations. The diversity of matrix algorithm topics was one of the recurrent themes repeated over and over by the participants, who said that this made this meeting particularly special. This breadth in the topics of the conference illustrated the richness of the field usually referred as "Computational Linear Algebra". This field includes the solution of systems of linear equations (ubiquitous in many applications in science and engineering), the design and implementation of preconditioners, solution of eigenvalue problems and combinatorial matrix problems.

The participants appreciated the ability to have discussions with people whom they would hardly meet at other conferences. The interaction between people from different areas of work was very fruitful. There were many such examples of people who have known of each others' publications, but at the seminar they had the chance to interact with each other for the first time. Other researchers who did know each other were able to renew their contacts and collaborations.

The Dagstuhl environment added to the group's sense of camaraderie. There was ample time for informal discussions, and people took real advantage of this. It was not unusual to see people working together in the evenings. In fact, it was hard to find a working area not occupied by two or three participants writing on paper at tables, or on whiteboards.

The program of presentations gives a clear idea of the multiplicity of topics discussed, from wireless communications to Quantum Chemistry. Scientists who work mostly in theoretical aspects of the field contributed ideas to those working in applications areas. At the same time those theoreticians felt inspired by the new problems presented. During and after the talks, there were many questions and discussions. There was a real interaction between speaker and audience. Overall everyone agreed was that it was a very stimulating meeting.

Chapter 2

Verification, Logic

2.1 Verification and Constructive Algebra

Seminar No. **03021** Date **05.01.–10.01.2003**
Organizers: T. Coquand, H. Lombardi, M.-F. Roy

General Presentation

The meeting was an attempt to bring together people from different communities: constructive algebra, computer algebra, designers and users of proof systems. Though the goals and interests are distinct, the meeting revealed that there is a strong core of common interests, the main one may be the shared desire to understand in depth mathematics concepts in connections with algorithms and proofs. An interaction appears thus to be possible and fruitful. One outcome of this week was the decision to create a European group under the acronym MAP for "Mathematics: Algorithms and Proofs". As we said in our proposal: "If there is enough common interests and good interactions during the week, the Dagstuhl seminar could be the starting point of a European proposal on the same topic, with more ambitious goals." This is indeed what happened.

Summary of the meeting

Here are some common themes that emerged in the meeting on constructive algebra and verifications. There is no attempt to be exhaustive.

- **Certificates**

  A first common theme that emerged can be captured by the notion of "certificate", and was exposed clearly by the talk of Arjeh Cohen. This notion unifies some attempts to connect proof systems and computer algebra systems, that were the topic of the talks of Loic Pottier and David Delaye. The idea is roughly that computer algebra should communicate mathematical data together with a *certificate*, which

represents the information needed to complete a proof of correctness of the mathematical data. This notion is reminiscent of the difference NP/P: it may be hard to check that a formula is a tautology but it is easy to check a proof. A simple example is provided by the gcd of two polynomials P and Q. The computer system should communicate not only the answer G, but also a certificate, that may be four polynomials A, B, C, D such that $AP + BQ = G$, $P = CG$, $Q = DG$. To find G may be hard, but to check these equalities is easy. A more sophisticated example was the topic of the talk of Loic Pottier (special cases of quantifier eliminations for reals), who had to program in CAML his own version of a computer algebra algorithm in order to get the desired certificates.

This notion of certificate is also closely connected to the talk of Helmut Schwichtenberg (common to all interactive proof systems with explicit proof objects): a starting point of such work is that it is undecidable in general whether a given program meets its specification. In contrast, it can be checked easily by a machine whether a formal proof is correct. The proof object itself can thus then be used as a certificate.

It is curious that a similar notion of certificate was used in the talk of Dmitrii Pasechnik. There, of course, the goal is completely different, which is to provide interesting strong propositional proof systems with lower bound results. Finally, the talk of Laureano Gonzalez-Vega was concerned with the difficulty of computing algebraic certificates in some geometrical statements in Real Algebraic Geometry.

- **Algorithms in Mathematics, via Proof Theory**

  A second theme is what one may call the relevance of classical mathematics to algorithms. The talks of Henri Lombardi, Marie-Francoise Roy and Ulrich Kohlenbach showed, in very different ways, that mathematical proofs that use a priori highly non computational concepts, such as Zorn lemma, or compactness principles, contain implicitely very interesting computational informations. The talk of Ulrich Kohlenbach presented a way to extract implicit informations in proofs, in such a way that one can even obtain new theorems, surprising to the expert, from these informations (here in the field of metric fixed point theory). One interesting topic is to compare the two approaches: in Lombardi and Roy's talks, to use techniques from geometric logic, and in Kohlenbach's talk, a modification of Gödel's Dialectica interpretation, that is especially well suited to extract bounds from classical proofs. Ulrich Kohlenbach said for instance that it should be interesting to use his methods also for examples on algebra, where the dynamical method of Lombardi-Roy has been used so far. A general feeling, emerging from some talks and discussions, was that the algorithms extracted by the dynamical method from a priori non effective proofs, may give algorithms that are better (even feasible) than the algorithms one can extract more straightforwardly from usual constructive arguments. For instance, in usual constructive mathematics, one requires to have a test of irreducibility for polynomials. While such a test exists in some cases, they are usually quite inefficient. The algorithm corresponding to a proof using this test is thus a priori also inefficient. By contrast the algorithm extracted from dynamical methods does not rely on such tests. It was suggested by Henri Lombardi that some efficient algorithms may be obtained in this way in number theory (dynamical theory of Dedekind domains).

Such claims, if they happen to be verified, are of fundamental importance.

- **Progress on basics**

  Another theme is best expressed by one sentence taken from the presentation of the seminar: "It is remarkable that in constructive and computer algebra, progress in sophisticated algorithms often implies progress on basics". This point was stressed in the talk of Peter Paule on symbolic summation for instance, who provided basic examples that would be welcome additions to basic courses on calculus, and several time in discussions, for instance for algebraic topology. Another example was provided by the talk of Gilles Dowek, who, motivated by quite concrete problems in safety of air trafic control, presented a new form of induction over real numbers that may be interesting for presenting basic proofs in real analysis.

- **Proof Systems and Computer Algebra Systems**

  A large part of the talks was concerned about connections between computer algebra systems and proof systems. Peter Paule reminded us, with some concrete examples, that people in proof system should be more aware of the power of current computer algebra systems. The talks of Renaud Rioboo presented a system aiming at combining proofs and computer algebra computations. The talks of Clemens Ballarin and Julio Rubio supplemented the talk of Francis Sergeraert by presenting an on-going attempt to use techniques from formal methods and interactive proof checking to ensure the correctness of a large sofware system for computations in algebraic topology. One interesting conceptual connection emerged from the talk of Peter Paule, on the concrete example of checking tables of equalities between special functions. But there is a mismatch between this representation and the representation of expressions as *functions* of real or complex quantities. Typically, the functions may have pole, or may involve ambiguities. What interests primarily the user of such tables is of course the interpretation of expressions as functions.

  This suggests a natural place where proof systems may complement computer algebra systems. Such a connection appeared in the talks of Loic Pottier and David Delaye. The simplest example may be provided by the equality $x \times 1/x = 1$. This equality is perfectly valid from the computer algebra viewpoint, since it is interpreted in the field of rational expressions (field of fractions of a polynomial ring). Considered as a function $x \longmapsto 1/x$ has a pole at $x = 0$ and the proof system will have to generate the condition $x \neq 0$.

- **Constructive Mathematics**

  Several talks were given on constructive mathematics. Francis Sergereart presented a way to do algebraic topology constructively, which is actually implemented in Common Lisp. Peter Schuster presented a constructive definition of the notion of scheme, a basic concept in modern algebraic geometry. There are probably deep connections between this presentation, based on point-free topology, and the talks of Henri Lombardi and Herve Perdry on dynamical algebras, that would be interesting to explore further. The talks of Erik Palmgren and Jesper Carlström were about Martin-Löf type theory. Type theory appears to be a potential formalism in which

several concepts that were presented at the workshop could be elegantly expressed. Just to take one example, if we succeed to express constructive algebraic topology, as presented by Francis Sergeraert, in type theory, one would have an algorithm (in a functional programming language) which is correct by construction, thus bypassing the need of a formal verification a posteriori. In the present stage however, this may seem utopic (probably the program obtained in this way would be too inefficient), but this might be an interesting project. The meeting ended by a talk of Bas Spitters on a constructive proof of Peter-Weyl's theorem, and it would be interesting to explore further the algorithmic ideas implicit in this proof.

Impact

The main positive surprise of the seminar was that communication is possible, and in fact highly appreciated, between quite distinct fields of mathematics and computer science. One participant expressed for instance his positive surprise to see in the same talk the name of Jean-Pierre Serre, who made fundamental contributions in algebraic topology, and the name of Turing, one of the founder of the mathematical notion of algorithm. The participants were working in different fields, but were all deeply interested in the interconnections between mathematics, algorithms and proofs, and several participants expressed the opinion that this combination of different topics with a strong common interest allows for a rich interaction. What was positive also was the emphasis, common to many talks, that progress in sophisticated mathematics and algorithms often implies progress on basics. This seminar was also a wellcome occasion to have a beginning of a real dialogue between designers and users of proof systems, and specialists in computer algebra and mathematics. Such dialogues have already started in research groups that were represented (Linz, Nijmegen, Paris VI) but the seminar showed new unexpected research directions (proof theory, constructive algebraic topology).

One outcome of this week was the decision to create a European group under the acronym MAP for "Mathematics: Algorithms and Proofs".

2.2 Objects, Agents and Features

Seminar No. **03081** Date **16.02.–21.02.2003**
Organizers: H.-D. Ehrich, J.-J. Meyer, M. Ryan

There are many ways of structuring software, and the seminar focussed on an established one (object-orientation) and two emerging ones (agent-orientation and feature-orientation).

- The object paradigm is now widely used in software technology (with programming languages like C++ and Java, and OO modelling frameworks such as UML). However, the theoretical foundations of the object paradigm are not settled yet, although clean concepts and reliable foundations are more and more demanded not only by

academia but also by practitioners. In particular, the precise meaning of UML concepts is subject to wide debate.

- Agents are more special kinds of objects, having more autonomy, and taking more initiative. For this reason, agent-oriented programming is sometimes referred to as 'subject-oriented' rather than 'object-oriented', indicating that an agent is much more in control of itself than an object which is manipulated by other entities (objects). There is some work on investigating typical object notions like inheritance in the context of agents. An interesting question is whether this is a fruitful way to go. Typically, agents are thought of being endowed with 'mental states' involving concepts like knowledge, belief, desires and goals, in order to display autonomous and in particular pro-active behaviour.

- Features are optional extensions of functionality which may be added to a software product, in order to reflect changes in requirements. They also cut across the class structure, because implementing a feature typically involves updating several classes or objects. The more complex the system is, the harder it is to add features without breaking something; this phenomenon has been dubbed the 'feature interaction problem'. Because users like to think of a system as comprising a base system together with a number of features on top, features could potentially be seen as a structuring mechanism rivalling objects and agents.

In recent years, concepts in object-oriented modeling and programming have been extended in several directions, giving rise to new paradigms such as agent-orientation and feature-orientation.

The Dagstuhl seminar explored the relationship between the original paradigm and the two new ones. The participants' reaction was very positive, and we are planning a Springer-Verlag book of the proceedings.
See `http://www.cs.bham.ac.uk/~mdr/research/dagstuhl03/cfp`

The main highlight was the exploration of the novel theme which ran throughout the seminar, namely the intersection and interaction between the three concepts of the seminar title. Some of the issues are highlighted in the following table.

Objects	*Agents*	*Features*
basic structuring mechanism	structuring mechanism	secondary structuring mechanism; cuts across existing structure
private data & message passing	private data & communication	violates privacy; invades code
reactive	deliberative, reflective, having belief/desire/intention	"goal oriented" desirable
prescribes behaviour	autonomous	autonomy desirable
monotonic (+ overriding)	non-monotonic	non-monotonic

2.3 Reasoning about Shape

Seminar No. **03101** Date **02.03.–07.03.2003**
Organizers: M. Müller-Olm, H. Riis Nielson, D. Schmidt

The recent theory and practice of computation has been strongly influenced by aspects of the *shape* (topology) of control, data, and communication structures. Instances of this phenomenon are

- the topology of objects in heap storage;
- the topology of secure networks;
- the topology of communication behavior.

The shape of the resulting topologies can affect and even determine program correctness, reliability, and performance. Different approaches have been developed to reason about such shapes. These approaches have similar aims, face similar technical difficulties, and have achieved similar basic successes, but the connections between the approaches are tenuous and vague.

To address this shortcoming, a Dagstuhl seminar on "Reasoning About Shape" was held on 2-7 March, 2003 that focussed on the topic of reasoning on heap-storage shape as those generated by functional, imperative, and object-oriented programming languages. The seminar was attended by 34 researchers from 8 countries. It brought together three distinct groups of people who use different techniques to study the topic:

- those who use static analysis;
- those who use logics;
- those who use model checking and theorem proving.

In order to facilitate communication between the three communities, four one-hour introductory tutorials were presented on the approaches:

1. **An Introduction to Shape Analysis**
   by Thomas Reps, University of Wisconsin
2. **An Introduction to Separation Logic**
   by Josh Berdine, Queen Mary University, London
3. **An Introduction to Model Checking and Flow Analysis**
   by Markus Müller-Olm, Universität Dortmund
4. **An Introduction to Heap-abstraction Methods**
   by David Schmidt, Kansas State University

The topics in the tutorials were developed by 24 technical presentations by the seminar participants. The seminar format provided ample time for discussion and development: each one-hour tutorial was followed by 20 minutes of discussion, and each 30-minute technical presentation was followed by 15 minutes of discussion. (Often, the discussion was intermixed with the presentation.) All talks will appear in the Online Proceedings.

Scientific Highlights

Several significant areas of study were developed by the technical speakers. Noteworthy (but not exhaustive) examples were

- improvement of static heap analysis, as presented by Greta Yorsh, Tel Aviv University ("Symbolic characterization of heap abstractions"), Eran Yahav, Tel Aviv University ("Use of evolution logic for verifying temporal properties of concurrent software"), and Thomas Reps, University of Wisconsin ("Symbolic implementation of the 'Best' transform")
- development of logic-based approaches to reasoning about heap storage, for example, Hongseok Yang, KAIST University, Korea ("Verification of the Schorr-Waite graph marking algorithm by refinement"), Peter O'Hearn, Queen Mary College, London ("Local reasoning and the frame rule"), and Cristiano Calcagno, Imperial College London ("Automatic reasoning of programs in spatial logic")
- application of model-checking and theorem-proving techniques, presented by Helmut Seidl, Trier Universität ("Linear algebra for program analysis"), Anders Møller, Aarhus Universitet ("Program verification with monadic second-order logic"), Patrick Maier, Max-Planck-Institut für Informatik, Saarbrücken ("Bounded model checking of pointer programs"), and Andreas Podelski, Max-Planck-Institut für Informatik, Saarbrücken ("Software model checking for safety and liveness")

Two other significant contributions must be mentioned: Martin Rinard, Massachusetts Institute of Technology, presented a talk on "Data structure consistency checking and repair," and Viktor Kuncak, also of MIT, spoke about "The undecidability of graph matching in monadic second-order logic." The latter talk demonstrated a negative result that impacts one direction of work followed by the static-analysis shape community and was a significant contribution made available by the Dagstuhl seminar.

Perspectives

In addition to the significant scientific contributions presented at the meeting, the seminar provided an important opportunity for the members of the three approaches to be exposed to the work of the other groups and discuss similarities, differences and potential for collaboration. After five days of presentations, discussions, and debates, two meetings were held to summarize the results of the seminar. Briefly stated, the conclusions are the following.

- Shape analysis is a viable research field with substantial intellectual content and significant applications and problems waiting to be solved. There are promising solutions and a community is building around the topic. The Dagstuhl seminar was a significant contribution to the development of that community.

- The field of shape analysis is not mature. As demonstrated at the seminar, there are many approaches, and it is unclear how to evaluate and compare the approaches. Nonetheless, it is important to proceed, because the topic is one of the last important semantical problems in the core imperative programming field.

- Future concerns must include applying existing approaches to larger problems, especially by exploiting abstraction and modularity principles. There is an uncertainty as to the degree of manual annotation versus automated inference that can be applied to solving the problem. Finally, more time must be spent on deciding upon those crucial shape properties that must be solved and developing the technology to deduce the crucial properties. The interactions between the three communities at the workshop were an impressive start, but more collaboration will be required.

2.4 Deduction and Infinite-state Model Checking

Seminar No. **03171** Date **21.04.–25.04.2003**
Organizers: D. Kapur, A. Podelski, A. Voronkov

Q: In 'infinite model checking', what is infinite, the model or the checking?

A: Both.

Model checking is an automated method to verify runtime properties of programs. Finite model checking applies to finite abstractions of software systems. Often, the task of constructing appropriate finite abstractions manually is hard, if not impossible. Therefore, a recent and promising research direction aims at infinite model checking. Here, deduction takes the central role in accounting for the infiniteness that arises from the direct modelling of software systems.

So far, the deduction problems arising in this context have been addressed in an adhoc manner by the model checking community. It is interesting to explore where existing techniques can be applied and where new kinds of research questions are raised.

For finite systems, model checking is based on Boolean logic. For many of the classes of systems with specific characteristics for infinite data and infinite control, the question for the right logic is still open (right in terms of appropriate expressiveness and computational cost). It will be useful to classify the deduction problems corresponding to the different classes of systems.

Data: What classes of formulas are best used to account for classes of operations over classical domains such as integers? What are the new domains to model pointer structures, lists, queues, abstract data types in general?

Control: Advanced control (recursion, concurrency, threads, dynamic objects with changing communication patterns, mobility of computational agents) requires models of process terms with specific algebraic laws (for stack concatenation, parallel composition); which ones exactly?

For safety properties, model checking amounts to automatically synthesising inductive invariants, by fixpoint iteration. For infinite model checking, the application of the fixpoint operator, the fixpoint test and the extrapolation of intermediate results each are theorem proving tasks. What are the demands, the functionality, and the evaluation criteria for theorem provers that are called during fixpoint iteration?

For example, the performance of a possibly incomplete decision tool for the validity of implication (used for the fixpoint termination test) determines a tradeoff where the fixpoint iteration terminates after either few but possibly expensive steps or cheap but possibly numerous steps.

Extrapolation of intermediate results during fixpoint iteration is required for accelerating or enforcing termination. The abstract interpretation framework of Cousot and Cousot formulates abstraction techniques at a semantic level. Their instantiation to syntax-based theorem provers is still not obvious.

There are many more possible topics to be discussed at our workshop...

2.5 Probabilistic Methods in Verification and Planning

Seminar No. **03201** Date **11.05.–16.05.2003**
Organizers: C. Boutilier, B. Haverkort, M. Kwiatkowska, M. Vardi

Introduction

Probabilistic modelling is widely used in the design and analysis of computer systems, decision support and scheduling problems, and has been rapidly gaining in importance in recent years. In a distributed environment, various randomized schemes have been found to act as symmetry breakers, leading to efficient, symmetric solutions to distributed coordination problems, for example leader election and consensus algorithms. Probability also provides means to model unreliable or unpredictable behaviour, aiding in the study of fault-tolerant systems, computer networks and queueing systems, and to predict their behaviour based on the calculation of performance characteristics. In decision-theoretic planning and reinforcement learning, probability is used to represent and quantify uncertainty, and to model computational processes under various scenarios.

Scientific highlights

Probabilistic techniques are extensively used in the following three areas:

1. **Performance evaluation** has its roots in the early 1910's when A.K. Erlang in Denmark developed stochastic capacity planning techniques for telephone exchanges. This so-called queuing theory developed further throughout the 20th century, aided by the efficient solution due to Buzen (1973) and the development of mean-value analysis by Reisen and Lavenberg (1977). The now established field of performance evaluation aims to develop formalisms and tools for modelling systems and analysing their performance measures, as a means to support the process of design and engineering. The analysis involves building a probabilistic model of the system being considered, typically a continuous time Markov chain (CTMC), but often a stochastic process of a more general nature as well. The model serves as a basis for analytical, simulation-based or numerical calculations which result in steady-state probabilities and the associated performance measures (resource utilisation, average call waiting time, etc). Alternatively, transient behavioural aspects, such as the probability of message delivery or quality of service dropping below minimum within a given time bound, can be analysed. The research in the area encompasses a variety of techniques, including measurement and testing, focusing on quantitative characteristics, and covers a broad spectrum of issues, for example designing description languages, formulating efficient numerical methods and tools for solving thus derived models, and queueing theory.

2. **Probabilistic model checking** (or probabilistic verification) is an extension of model checking techniques to probabilistic systems, first introduced by Hart, Sharir and Pnueli (1982). As in conventional model checking, a model of the probabilistic system, usually in the form of a discrete or continuous time Markov chain (DTMC/CTMC) or a Markov decision process (MDP), is built and then subjected to algorithmic analysis in order to establish whether it satisfies a given specification. The specifications are usually stated as formulae of probabilistic temporal logic, which in addition to conventional modalities may include probabilistic operators, whose outcome is true/false depending on the probability of certain executions occurring. The model checking procedure combines traversal of the underlying transition graph with numerical solutions of linear optimisation problems (for Markov decision process models) and linear equation systems and linear differential equation systems (for DTMC/CTMC models). The model checker can either produce a binary answer (yes or no, true or false), by comparing the obtained probability with the given threshold, or simply return the likelihood of the executions instead. Although algorithms for model checking probabilistic systems have been known since the mid-1980's (Vardi, 1985), it is only recently that experimental, tool implementation work has begun. The main thrust of the research in this area is to further the experimental work by learning from and incorporating the successful techniques of conventional model checking, for example by adapting symbolic techniques (such as MTBDDs) for model checking probabilistic systems, or the use of uniformisation (a well-developed technique in performance evaluation) for model checking timed properties for continuous time Markov chains. The foundational work continues to seek out new algorithms, notations and languages, and to adapt them to specific applications which require probabilistic modelling.

3. **Decision support** and **adaptive control** ("planning") heavily depend on adequate modelling of uncertainty due to environmental factors. Markov decision processes, which originated in operations research in the 1950's, serve as a representation for planning and control problems which can be analysed by solving appropriately derived variants of Bellman equations. Since the 1980's, following the pioneering work of Dean and Kanazawa (1989) and Tatman and Shachter (1990) on exploiting structure in representation and solution of such problems, Markov decision processes have been central to the research in automated planning in Artificial Intelligence. In subsequent years, a great deal of progress was made exploring structured versions of earlier algorithms for unstructured problems, and also using the basic technology for model checking, including binary decision diagrams (BDDs). True MDPs, i.e., problems in which the current state of the system is completely observable to the decision maker, are rare in practice and hence the partially observable variant (POMDP) is of great importance. Recently, there has been a resurgence of interest in POMDPs, and a host of new algorithms have been developed, including variational methods which open up the possibility of solving a wide range of problems. Variational statistical methods can in some cases reduce the need for state space exploration using a combination of sampling techniques and a reformulation in terms of a (continuous) parameterized space of actions.

It should be clear from the above that there are commonalities between the main research challenges of the three areas:

- they have to deal with very large state spaces, and therefore have to resort to structure in order to arrive at compact model representations: in model checking variants of BDDs are used, and in performance evaluation the Kronecker representation as well as matrix diagrams;
- they draw on probabilistic techniques and require appropriate efficient numerical solution methods (linear equations, linear differential equations and linear programming) capable of handling very large models.

At the same time, there are differences in their respective focus and research goals. Performance modelling has developed mature analytical, numerical and simulation methods for analysing various probabilistic systems; it can evidently serve as a useful source of expertise in Markovian/non-Markovian analysis techniques and numerical computations not normally employed in conventional model checking. Likewise, planning and scheduling has led to the emergence of sophisticated MDP/POMDP algorithms; since MDPs arise as models of randomized distributed algorithmic schemes, these may well be relevant for probabilistic model checking. In turn, model checking, and probabilistic model checking in particular, can offer advanced efficient techniques for analysing the underlying transition graph.

Seminar programme

The seminar programme included five one-hour tutorials, by Moshe Vardi, Christel Baier and Joost-Pieter Katoen on probabilistic model checking, Bob Givan and Ron Parr on

Markov decision processes, and Bill Sanders on dependability and performability evaluation. The remainder of talks were either long (45 min) or normal (30 min) research presentations.

There was also a panel discussion chaired by Marta Kwiatkowska which involved two researchers from each of the three areas: Christel Baier and Prakash Panangaden from probabilistic verification, Bob Givan and Richard Dearden from planning and Bill Sanders and Gianfranco Ciardo from performance modelling.

Perspectives

It became clear during the seminar that the three areas (probabilistic model checking, decision support and performance evaluation) are indeed closely related. All are concerned with variants of probabilistic models, typically Markov processes of some kind, their efficient representation utilising structure, and automated procedure for their analysis. Whereas the probabilistic model checking and performance evaluation communities were aware of this connection, and indeed some of the researchers straddle the two areas, it was not until the meeting that it was realised how close the decision support field is to the other two fields as well. We saw many examples of this in the talks; for example, conventional MDP algorithms from the decision support field, such as value iteration, are being used in probabilistic model checkers. We expect much future exchanges between the areas.

The panel discussion focused on the question of what each area can offer to the other two, and vice-versa. A common view that emerged is that in the probabilistic model checking field predominantly discrete mathematical theories are used, whereas in the decision support and performance evaluation field continuous phenomena play a major role, in which the notion of approximation is fundamental. It was felt that also in probabilistic model checking the notion of approximation should become more important.

2.6 Language-Based Security

Seminar No. **03411** Date **05.10.–10.10.2003**
Organizers: A. Banerjee, H. Mantel, D. Naumann, A. Sabelfeld

Summary

Modern computing systems are particularly vulnerable to security attacks at the application level. Traditionally, security mechanisms have been based on low-level protection such as OS-based monitoring and access control. However, application-level attacks (e.g., the widely-publicized Lovebug and Melissa viruses executed on behalf of a mailer application) operate at a higher level and circumvent the security mechanisms. Not only is malicious code a threat to security, but also unintended bugs in the specification and implementation of systems can lead to equally disastrous effects.

Application-level security is becoming an increasingly popular area of research because there is an increasing demand for applications to provide high assurance that particular security policies are followed. An effective way to achieve high assurance is to counter security threats at the same level as attacks the application level. Because applications are typically specified and implemented in programming languages, this area is known as language-based security. A direct benefit of language-based security is the ability to naturally express security policies and enforcement mechanisms using the techniques of the well-developed area of programming languages. These techniques facilitate rigorous specifications of security policies as well as their mechanical verification.

Language-based techniques are gradually entering standard security practices. For example, the Java byte-code verifier is a language-based enforcement mechanism of particular integrity properties. As another example, the Java Virtual Machine and the .NET runtime system provide a dynamic access control mechanism that inspects the runtime stack to check whether permissions have been granted to code in the calling chain.

Despite such forays into mainstream security practices, there are a number of open issues in language-based security. One problem is to preserve the confidentiality of data by programs. This involves specification and enforcement of a property that guarantees that a program's public outputs do not (explicitly or implicitly) reveal information about the program's secret inputs. Recent technical advances allow enforcing confidentiality using a variety of language-based techniques e.g., type systems, data-flow and control-flow analysis, abstract interpretation, model checking, etc.

While more and more realistic security properties for more and more expressive languages are being considered, there are critical challenges remaining in the area of language-based security in general and in the area of program confidentiality in particular. To name just a few:

- security in concurrent and distributed systems
- minimization of the trusted computing base
- system-wide security
- security analysis for machine languages
- certifying compilation
- compositionality of security properties
- high assurance in the presence of downgrading
- security protocols

To gain insight on these challenges, and with the ultimate goal to enhance standard security practices with language-based protection mechanisms, a Dagstuhl seminar on "Language-based Security" was held October 5 - 10, 2003.

The seminar was attended by 59 researchers from 10 countries. There were 39 technical presentations and 3 tutorials by the seminar participants. When the organizers sent out

the initial invitations, they had expected between 35 to 45 participants. In the end, there were close to 60, and the seminar was oversubscribed. Except for the hour-long invited talks, most presentations were 30 minutes long. Each full day ran from 9 AM to 6 PM. The large number of talks posed a scheduling challenge for the organizers. But thanks to some participants who chose not to talk and to some who agreed to make short (15-20 mins.) presentations, the scheduling became more manageable. The talks followed a standard conference format with questions/discussions during and at the end. Breaks between talks also facilitated discussions. The busy schedule made it at times difficult to have too many discussions during a talk. Therefore, two provocative discussion sessions were arranged. An evening discussion led by Peter Ryan, focused on providing a definition to language-based security and open issues in the area. Another discussion led by Greg Morrisett, focused on security issues and open problems for next generation virtual machines.

In addition to discussions over drinks, cards and billiards, it was quite common to find groups of participants working past midnight. Many chose to keep working despite diversions like the Wednesday afternoon excursion. The organizers are grateful that it was possible to arrange an organ concert by participant Michael Clarkson at the Dagstuhl chapel. This was an unusual bonus to the busy academic schedule.

Scientific Highlights

The main theme of the seminar was further elucidated by way of three one-hour tutorials on three main application areas of language-based security.

- Language-based Information-Flow Security by Andrei Sabelfeld, Cornell University.
- Typed Assembly Language (Background) by Greg Morrisett, Cornell University.
- Protocol Analysis by Dieter Gollmann, TU Hamburg-Harburg.

Sabelfeld's talk gave a survey of current research on information flow security, particularly focusing on static program analysis to enforce such policies. Four main subareas of research and open problems in information flow security were detailed: enriching language expressiveness, exploring the impact of concurrency, analyzing covert channels and refining security policies.

Morrisett's talk gave a survey of current research on typed assembly language. Typed assembly language is an idealized RISC-style assembly language with a formal operational semantics for a simple abstract machine. The goal is to provide type structure for machine code so that useful abstractions may be supplied to compilers to support specific security policies like memory safety.

Gollmann's talk gave a survey of approaches to protocol analysis and verification that are in some way linked to programming languages. Examples of topics considered were the Dolev-Yao intruder, protocols analyzed using CSP/FDR (e.g., Needham-Schroeder public key protocol), protocols specified using nominal calculi like the Y-calculus, and the use of protocol analysis in the context of secure APIs. The talk emphasized verification of

protocols with respect to specified security goals in specified environments, the Dolev-Yao intruder model was discussed as one such example environment. Thus care must be taken in claiming discoveries of new flaws in a protocol: one needs to make sure that the protocol was indeed analyzed in an environment for which it was intended. The talk suggested that language-based protocol analysis tools are likely to be most useful for developers instantiating established protocol design techniques for use in standard environments, e.g., in web services.

Several significant areas of study were developed by the technical speakers. Noteworthy (but nonexhaustive) examples are:

- *Access control, cryptography, information flow and noninterference* as presented by Eijiro Sumii, University of Pennsylvania (A bisimulation for dynamic sealing), Tamara Rezk, INRIA, Sophia Antipolis (Noninterference for the Java Virtual Machine), Dominic Duggan, Stevens Institute of Technology (Type-based distributed access control), and Roberto Giacobazzi, Università di Verona (Parameterized secrecy by abstract interpretation).

- *Specification and automatic validation of properties of security protocols*, as presented by Bruno Blanchet, MPI, Saarbrücken (Automatic proof of strong secrecy for security protocols), Andre Scedrov, University of Pennsylvania (A probabilistic polynomial-time calculus for the analysis of cryptographic protocols), Flemming Nielson, Danish Technological University, (Automatic validation of protocol narration), Riccardo Focardi, Università Cá Foscari, Venezia (Language-based security in authentication protocols), and Luca Vigano, ETH Zürich (An on-the-fly model checker for security protocol analysis).

- *Formulation of correctness properties for downgrading* as presented by David Sands, Chalmers University of Technology (Controlled downgrading based on intransitive (non)interference), and Reiner Hähnle, Chalmers University (A theorem proving approach to secure information flow).

- *Memory safety* as presented by Drew Dean, Stanford Research International (Definition of memory safety), Michael Hicks, University of Maryland (Safe and flexible memory management in Cyclone), and Gogul Balakrishnan, University of Wisconsin (Analyzing memory accesses in x86 executables).

- *Information flow policies in distributed systems and dynamic security policies* by Mads Dam, Swedish Institute of Computer Science (Information flow control for cryptographic applets), Andrew Myers, Cornell University (Using information flow policies to construct secure distributed systems), and Stephan Zdancewic, University of Pennsylvania (First class principals in the decentralized label model).

A special mention must be made of the talk, "A semantics for web services authentication", presented by Cédric Fournet (Microsoft Research, Cambridge) which considered the problem of specifying and verifying cryptographic security protocols for XML web services. The protocols themselves are based on a faithful account of the XML wire format and are described as Y-calculus processes. The work demonstrates a direct application

of language-based security techniques in an area of practical importance, namely, web services.

Finally, the talks on downgrading presented initial results in an area that significantly impacts the entire direction of work on information flow security and noninterference and was a significant contribution made available by the Dagstuhl seminar.

Besides the tutorials and the talks, there were two open discussion sessions. Peter Ryan led a discussion on providing a definition for language-based security and challenging issues in language-based security. The consequences of a simple definition – application of semantics and programming languages to security – was explored in the following contexts: modeling user behavior, analyzing protocols, developing trustworthy code, formalizing software requirements and guiding software design by providing proper abstractions. Among the challenges discussed were compositionality of protocols and the connection between noninterference and access control.

Greg Morrisett led a discussion on next generation virtual machines. The main question considered was how one might design a secure operating system retaining the good features of abstract machines like the Java Virtual Machine (JVM) or Microsoft's Common Language Runtime (CLR). Some issues focused on were: types for access control (capabilities), transfer of capabilities, resource control, JVM's thread model, exception handling and stack inspection.

Perspectives

The seminar was the first gathering of researchers working under the rubric of language-based security. The vibrant atmosphere at Dagstuhl provided an excellent opportunity for participants to be exposed to each other's research, to compare and contrast different approaches and to seek potential collaborations.

After five days of presentations, discussions, and debates, David Naumann led the discussion summarizing the main results of the seminar. Briefly stated, the conclusions are the following:

- Language-based security is a thriving research area with substantial intellectual content and significant applications and problems waiting to be solved. As the level of participation shows, there is a large community that has built around the topic. Participants deemed the Dagstuhl seminar a success and felt that a significant contribution has been made to further strengthen the community.
- Language-based security has a potential for substantially facilitating *security by design*. Applications are implemented in programming languages, systems are modeled at different levels of abstraction (using different languages), and security policies can be expressed and analyzed at each of these levels (e.g., by static analysis, model checking, formal verification). A closer integration of the various analysis techniques and their underlying security properties is the challenge we have to face for exploiting efficient language-based security techniques (e.g., security type systems) more systematically in a security-by-design approach.

- Critical open problems/questions emerging from seminar discussions include:

  - How to specify differing security goals for different parts of a system? For example, the same resource may have low integrity in one part of the system, hence performing a trusted action based on the resource is potentially dangerous. Yet, in another part of the system it may be safe to proceed with any action based on the resource.
  - How can specialized automated checking tools be integrated with constructive formal methods in high assurance software processes?
  - At what level (source or object) should the analysis and transformation of particular parts of a system be carried out?
  - How to handle dynamic information flow policies, e.g., information flow policies that change over time? How to connect to PKI?
  - How to specify and check security policies in extensible systems where components may be written in different languages? For example, how to specify confidentiality in a system where a webserver written in Java talks to a database implemented in SQL? Technical goals here may involve modular checking of security policies, composition of security policies, compiling to a common intermediate language and agreeing on a semantics for the common intermediate language.
  - Specification and verification of secrecy properties for security protocols (e.g., protocols for secure web services).
  - Integration of security protocols into compilers.
  - Language-based treatment of mutual distrust.
  - Language-based treatment of covert channels.
  - How do language-based abstraction mechanisms interact with information flow?
  - How can language-based techniques impact the design of next generation platforms (e.g., Microsoft's Common Language Runtime, or more speculatively, a secure operating system)? What security properties may one demand of a secure operating system?
  - Development of a toolset to reduce the trusted computing base.
  - What assurances can be provided in a system that employs downgrading?
  - Empirical studies on the efficiency and usability of language-based methods in large systems. For example, for checking confidentiality using a type-based information flow analysis for a system where downloaded applets may call trusted library methods, it is possibly inefficient to annotate all library methods with information flow properties.

2.7 Hardware and Software Consistency Models: Programmability and Performance

Seminar No. **03431** Date **19.10.–24.10.2003**
Organizers: J. Knoop, J. Lee, S. Midkiff, D. Padua

Hardware consistency models define the order that events that occur on one processor, or memory subsystem, appear to occur to other processors or memory subsystems. We use Memory model to refer to the equivalent software concept. A memory model can be defined as part of the semantics of the programming language. The memory model defines the order that memory references in thread of a program, written in the language, should appear to other threads, written in the same language. A memory model defines the order that memory references in a thread of a computer program are mandated by the semantics of a language or other piece of system software to appear to occur in other threads in the computer program. Until recently, these issues were largely the province of specialists who designed memory subsystems and processor cache protocols, implementors of operating systems, and database architects. The design of consistency and memory models was skewed towards providing high performance at the expense of usability or programmability. There are at least two contributing factors for this. First, processors were expensive, and never quite fast enough, requiring performance be maximized. Second, multithreaded programming was used almost exclusively in the design of widely used components such as database systems and operating systems. Thus very labor intensive approaches to programming these consistency models was acceptable. Most ordinary programmers never had to deal with memory consistency issues.

The widespread availability of explicitly parallel programming targeting shared memory systems has changed this equation. In particular, Java, OpenMP, C#, P-Threads, and distributed shared memory systems have forced programmers to be aware of the underlying semantics of the memory model. And, in all of these systems, poor performance, incorrect programs and lack of portability can result from an improper understanding of the underlying model. Thus knowledge that was formerly required of a relatively small number of specialists is now required of large numbers of programmers, in fact, required of the typical programmer. Given that the systems written by these typical programmers are not as widely disseminated as the systems written by the specialists, the cost of coping with the vagaries of consistency models is relatively much higher. Moreover, as the complexity of operating systems and middleware grows, the complexity of hardware and consistency models and software memory models leads to subtle errors in the code, degrading software reliability.

These changes in the tradeoffs between programmability and performance in memory models have sparked renewed research into how to design both consistency and memory models. Topics of intense interest include:

- What are the trends in hardware and software consistency models?
- What is the performance loss associated with moving towards simpler consistency and memory models? How much loss is acceptable?

- How can hardware consistency models be made simpler for programmers with acceptable losses in performance?
- What compiler techniques can be used to mask the complexity of hardware consistency models, or mask the performance costs of simpler hardware consistency models?
- How can memory models be designed to allow programmers to more easily write correct programs? What are the costs of doing this in terms of missed compiler optimization opportunities and additional synchronization overhead in modern out-of-order processors?
- Can compile-time analyses and optimizations mitigate some of these costs, and if so how?
- Are heuristic approximations to expensive compile-time analyses sufficient?
- What idioms and software engineering tools can be used to increase programmability in the face of complex memory models?

We have two large goals for the seminar. First, we would like to foster discussions about the usability and performance requirements of consistency models in the different areas where these are important issues (architecture and hardware, databases, and programming languages) and give knowledgeable members of the fields the opportunity to learn from the experiences of their colleagues in different fields. From these discussions, we hope to come to a better understanding of the tradeoffs and possibilities that can be exploited by researchers and practitioners in each of these areas, and to come up with important research questions that will yield broadly applicable results. Because of Dagstuhl's schedule allowing for mix of unstructured discussion in a congenial environment and more formal presentations, we see it as an ideal setting for bringing together members of these different communities to tackle these difficult issues.

2.8 Applied Deductive Verification

Seminar No. **03451** Date **02.11.–07.11.2003**
Organizers: D. Basin, H. Ganzinger, J. Harrison, A. Pnueli

Summary

Software and hardware systems are increasingly employed in safety or mission critical applications. Deductive verification can be used during development to minimize the risk of their failure. Although the costs associated with verification are often considered high, verification methods have achieved considerable success and there is increasing industrial interest in applying such methods.

The aim of this Dagstuhl seminar was to bring together researchers from academia and industry who are applying deduction to substantial "real-world" problems. We interpret deduction in a broad sense including interactive and automated theorem proving, model checking, program analysis, and the use of decision procedures. Deductive verification is the application of these methods to system analysis; its scope ranges from using theorem provers to carry out full-scale system verification to more light-weight applications that are easier to automate, such as analyzing system properties using model checkers or other decision procedures. Topics relevant for the seminar included research on:

- promising application-oriented foundations,
- method combination (e.g., integrating deduction and model checking), and
- abstraction and other techniques that can reduce the complexity of verification problems.

Applications include:

- Software verification, including protocols, concurrent systems, multimedia applications, and security,
- Hardware verification, including pipelined architectures and cache protocols as well as parameterized verification, and
- Tool verification, i.e., the verification of tools used in safety critical application, such as hardware-targeted compilers.

During the seminar we aimed to achieve a cross-fertilization between theoreticians and practitioners working in the area. This was achieved both by overview talks on the state of the art in the application of deductive methods, and by providing a forum for communication between researchers working on theory with practitioners from industry who are applying verification tools to large-scale applications. The seminar also featured evening tutorials and tool demonstrations.

Scientific Highlights

Formal techniques are increasingly being used to tackle 'real-world' industrial applications, and several speakers provided evidence of this. For example, Thomas Arts gave a fascinating overview of formal verification activity at Ericsson, while Patrick Cousot discussed a rigorous proof that the avionics software used in current Airbus aircraft cannot encounter floating-point overflows.

While we have not yet reached the stage of being able to perform a complete verification of large systems, we have many examples of proving either 'big properties of small systems' or 'small properties of big systems'. Indeed, we can see complete formal verification as one end of a continuum with traditional forms of static checking (type checking etc.) at the other. Thomas Ball and Sriram Rajamani discussed the impressive success of the

SLAM static checker, using theorem proving technology to enhance static checking, which apparently identifies a productive point on this continuum.

One key technique for tackling large and complex problems is *abstraction*, and several speakers discussed the use of abstract interpretation in this capacity. Another powerful technique in real-world problem solving is the identification of certain canonical classes of problems into which many others can be mapped (e.g. propositional satisfiability, linear or semidefinite programming). Armin Biere's talk suggested that quantified boolean formulas (QBF) may become such a class in the near future.

Meanwhile, steady progress on more traditional fronts was reported. For example, Harald Rueß discussed the current techniques and progress made for combining decision procedures for quantifier-free theories, and Ken McMillan surveyed his key idea of using interpolants to allow bounded model checking to be used for complete correctness verification, not merely bug-finding.

2.9 Perspectives Workshop: Design of Systems with Predictable Behaviour

Seminar No. **03471** Date **16.11.–19.11.2003**
Organizers: L. Thiele, R. Wilhelm

Embedded Systems with hard real-time requirements are abundant in our environment, in cars, airplanes, trains, production facilities, household appliances, and entertainment systems. Many of them are found in safety-critical systems whose failures can endanger human life. Verification of functional properties and non-functional properties such as the satisfaction of real-time constraints is mandatory. The Forum will concentrate on the real-time aspects.

The trends in processor design and in software development, however, makes this verification more and more difficult. Processor architectures are optimized for average-case performance using components such as caches, pipelines, and all kinds of speculation. They make processor behaviour hard to predict such that often overly conservative assumptions have to be made leading to a waste of hardware resources if real-time constraints have to be satisfied. On the other hand, methods to safely predict processor behaviour for a given program have been developed. Their success strongly depends on certain architectural features, e.g. the cache replacement strategy, and on the software-development discipline. For example, it has been shown that certain cache replacement strategies which are found in heavily-used processors do not allow precise predictions of the cache behaviour. On the positive side, much safety-critical code is automatically synthesized from formal specifications and very easily analyzed.

Larger embedded real-time systems often run on top of real-time operating systems (RTOS). These often take scheduling decisions dynamically. Their behaviour has to be analyzed together with the application.

A new trend is the development of real-time systems using real-time middleware and component based design. Adding real-time middleware further complicates the task of verification, since the middleware has to be included in the verification. Component-based design requires an incremental approach to verification.

Several scientists working on embedded systems, mostly in Europe, have recognized that a serious combined effort by representatives from several fields is necessary to establish a discipline "Design of Systems with Predictable Timing Behaviour". These areas are: Processor Architecture, Compiler Construction, Timing Analysis, Real-Time Operating Systems, Code Synthesis.

The goals of the Forum are

- to exchange ideas between the different groups about design principles for predictable systems,
- to exploit synergies resulting from the combination of such principles originating in different groups, e.g., the combination of processor-design principles with insights gained in the Timing-Analysis group,
- clarify the relation between average-case performance and predictable worst-case performance,
- lay the groundwork for a discipline "Design for Predictability".

Chapter 3

Geometry, Image Processing, Graphics

3.1 Computational Geometry

Seminar No. **03121** Date **16.03.–21.03.2003**
Organizers: D. Halperin, G. Rote

Geometric computing is present in virtually every corner of science and engineering, from computer-aided design and manufacturing to cartography and structural molecular biology. For over two decades, Computational Geometry has supplied the solid foundation for the study of algorithms which are relevant to all these areas.

Traditionally, Computational Geometry has treated linear objects like line segments or polygons, occasionally also circles and ellipses or other special shapes. For many novel applications, it is important to handle more general curves or surfaces that might be given as splines or in a general parametric form. Such shapes should be handled by algorithms directly, not only by piecewise linear approximation as has been done so far. Examples of applications that will benefit from extending the Computational Geometry repertoire to curved objects are: robot motion planning with many degrees of freedom (as has been demonstrated in the seminar), advanced manufacturing techniques involving micro manipulation and assembly, and computer-aided surgery, to name a few. Considerable portion of the seminar was dedicated to presenting and discussing recent progress in geometric computing with curves and surfaces.

Twelve talks dealt with this topic, ranging from special number types to support the robust handling of curved objects through algorithmic techniques to implemented systems (talks by Yvinec, Teillaud, Emiris, Mehlhorn, Wolpert, Wein, Halperin, Morin, Demaine, Fortune, Calatayud, and Schirra).

Another perspective on similar issues has been provided by several invitees from the area of Geometric Modeling and CAGD (talks by Brüderlin, Brunett, Hagen). This was nicely complemented by talks on modeling techniques in curve and surface reconstruction by Giesen, Dey and Ramos. Straddling both Computational Geometry and Computer-Aided

Geometric Design was the talk by Morin, who described the joint work with Knauer on geometric filtering for parametric curves.

Additional topics were applications in wireless communication (talks by Smorodinsky and Funke), meshing (talk by Shewchuk), geometric optimization problems with applications to cartography, metrology, and other areas (talks by Barequet, Brönniman, van Kreveld, Har-Peled, Cabello, Mitchell, and Efrat), the geometry of lines in space (talks by Cheong, Koltun), and large kinematic structures with applications to molecular simulation and robot motion planning (talks by Agarwal, Knauer, and Guibas).

Novel perspectives on algorithms for geometric problems were proposed by Bernard Chazelle, who presented an approach for solving geometric problems in sub-linear time, without looking at the whole data, and Chee Yap on a new general framework of pseudo-approximation algorithms.

There was an unusually large number of participants (67), many of whom gave presentations about their latest results (46 presentations), lasting 10-30 minutes. Still, there was ample time for scientific discussions and social interaction during the extensive lunch breaks, in the evenings, and during the excursion on Wednesday afternoon. Special care was taken to give younger participants the opportunity for presentations and to get them involved in the discussions.

During the seminar, there was a meeting of representatives of the CGAL (Computational Geometry Algorithms Library) group to coordinate work on arrangements of curves. Several key issues in implementing arrangements of curves were clarified and resolved by the shared experience of all sites.

An open problem session was held on Monday night. It lasted two hours and additionally stimulated the discussions during the workshop. Some problems were solved during the week or even right at the problem session. A list of open problems was collected.

3.2 Scientific Visualisation: Extracting Information and Knowledge from Scientific Data Sets

Seminar No. **03231** Date **01.06.–06.06.2003**
Organizers: G.-P. Bonneau, T. Ertl, G.M. Nielson

Scientific Visualization is a research area that is having great impact on how computers are used in research. Scientific Visualization is concerned with techniques that allow scientists and engineers to extract knowledge from the results of simulations and computations. Advances in scientific computation are allowing mathematical models and simulations to become increasingly complex and detailed. This results in a closer approximation to reality thus enhancing the possibility of acquiring new knowledge and understanding. Tremendously large collections of numerical values, which contain a great deal of information, are being produced and collected. The problem is to convey all of this information to the scientist so that effective use can be made of the human creative and analytic capabilities.

This requires a method of communication with a high bandwidth and an effective interface. Computer generated images and human vision mediated by the principles of perceptual psychology are the means used in scientific visualization to achieve this communication. The foundation material for the techniques of Scientific Visualization are derived from many areas including, for example, computer graphics, image processing, computer vision, perceptual psychology, applied mathematics, computer aided design, signal processing and numerical analysis.

The methods of visualizing data developed by Scientific Visualization researchers presented at this seminar are having broad impact on the way other scientists, engineers and practitioners are processing and understanding their data from sensors, simulations and mathematics models.

This particular seminar focused on the topic of Segmentation. Segmentation is a key issue in extracting information and knowledge from scientific data sets. The problems of developing techniques for segmentation are extremely difficult but the benefits to the fields of engineering and medicine are tremendous.

Scientific Highlights

The topic of this seminar is Scientific Visualization. This type of research has impact on how other researchers and practitioners process data obtained by collection, simulation or modeling. This area of research is approximately a dozen years old. From the very beginning of Scientific Visualization, it has been recognized that segmentation is a key issue in extracting information and knowledge from scientific data sets. The problems of developing techniques for segmentation are extremely difficult but the benefits are tremendous. Intelligent segmentation involves the qualitative understanding of scientific data and the support for qualitative enquiries about particular features or attributes. The creation of systems that identify and segment features and attributes and produce useful resulting scientific images will remain a dream until we have widely applicable automation tools for specifying, detecting, and extracting knowledge from scientific data sets. Specific areas of active current research covered during the presentations include:

- Feature and knowledge property preservation through implicit, wavelet and other methods for building hierarchical, multiresolution models.
- The description of meta features and attributes such as patterns of vortex cores found in certain characteristic complex flows for vector fields.
- Segmentation approaches to the automatic determination of transfer functions of several dimensions.
- Segmentation, feature extraction and region of interest determination with multidimensional curvature schemes applied through watershed techniques.
- Statistical and probability based segmentation and feature extraction techniques.

- Complex geometry representation, Morse theory and other approaches to the inference, determination and preservation of genus and other topological attributes inferred from sampled data.

3.3 Hierarchical Methods in Computer Graphics

Seminar No. **03271** Date **29.06.–04.07.2003**
Organizers: M. Gross, D. Manocha, H. Müller, H.-P. Seidel

Over the last decade hierarchical methods, multiresolution representations and wavelets have become an exceedingly powerful and flexible tool for computations and data reduction within computer graphics. Their power lies in the fact that they only require a small number of coefficients to represent general functions and large data sets accurately. This allows compression and efficient computations. They offer theoretical characterization of smoothness and coherence, insights into the structure of functions, and operators, and practical numerical tools which often lead to asymptotically faster computational algorithms. Examples of their use in computer graphics include geometric modeling, mesh simplification, multiresolution surface viewing with automatic level of detail control, image and video editing, compression, global illumination computations, volume visualization, and animation.

There is strong evidence that hierarchical methods will become a core technique in computer graphics in the future. The seminar was a follow-up to a Dagstuhl-Seminar with the same title which we have organized in 1998. The development since then confirms this impression.

The idea of this Dagstuhl Seminar was to provide again a forum for the leading researchers in this area to present their ideas and to bring together applications and basic research in order to exchange the requirements of systems, interfaces, and efficient algorithmic solutions to be developed.

Another goal of the seminar was to provide an opportunity for discussing ideas and work in progress. International conferences with their densely packed schedules usually leave little room for this sort of scientific exchange. There is a requirement of events like Dagstuhl Seminars. This was demonstrated by the number of participants from different European countries and abroad.

Scientific Highlights

The majority of the presentations of the seminar can be assigned to one of five main topics: high-quality interactive graphics, data acquisition for realistic rendering and image-based modeling rendering, parametrization of meshes, adaptive/dynamic/deformable meshes, and processing/rendering of point data.

High-quality interactive graphics can be achieved in different ways. One aspect is to get the necessary throughput of data by algorithms adapted to the capabilities of graphics

processing units, PC processors, and parallel processing. One contribution of the seminar to this aspect concerned the possibility of interactive ray tracing which has been a grand challenge over the past twenty years, but now seems close to a solution. Interactive rendering of point sets has been another issue. Similar to ray tracing, a huge amount of operations have to be performed. As demonstrated at the seminar, hierarchization is a helpful approach to cope with this problem.

In order to achieve the necessary speed, the algorithms are usually implemented close to hardware, similar to assembler programming, using sophisticated instructions sets of e.g. graphical processing units (gpus). From the view of software production, this development is not satisfactory. One contribution concerned a language for gpu-programming, called Sh, which might be a tool for more efficient implementation.

Another approach to achieve interactivity is to develop simplified or adapted models of simulation for image generation which still yield visually satisfying results. Several examples have been presented at the seminar. An interesting question might be how to bring all those specialized approaches together.

The presentations to next topic of the seminar, data acquisition for realistic rendering and image-based modeling rendering, has shown the necessity, but also the achievement, of cooperation between the fields of computer graphics and image processing/computer vision. Themes of the presentations have been wavelet-based lightfield compression, wavelet environment matting, accurate light source acquisition and rendering, free viewpoint video and 3D TV, scanning large-scale articulations and learning and application of class-specific information for facial modeling and animation. Hierarchization, in particular by wavelets of different sort, helps to overcome the problem of the huge amount of data that has to be processed and stored.

The third topic, parametrization of meshes, is a central issue of mesh processing. New contributions concerning improved parametrizations have been presented, over the sphere, as well as for parametrization of meshes of arbitrary genus.

The fourth topic, adaptive, dynamic and deformable meshes, is less settled than the topic of static meshes. Themes of particular interesting presentations on the algorithmic level have been interactive animation of objects represented by surface meshes and collision detection for deformable objects. An interesting talk on an application which demonstrates the power of the methods treated in the seminar concerned a geometric data base for gene expression.

The fifth topic, modeling and rendering of point data, currently finds intensive interest in the research community. It was represented in the seminar by two contributions which reported on recent important advances. The contributions were entitled “Hierarchical Splatting of Point Data” and “Multiscale Modeling of Point Sample Geometry”.

To each of those topics, both senior and young researchers have given presentations. It could be observed that most of the young researchers had already achieved an exceptional level in their specific topic of research. The wide variety of applications of hierarchization in computer graphics presented at the seminar offered the opportunity to them to recognize possibilities of application of their methods to applications somewhat besides their main interest. Many questions and discussions after the talks have shown that the opportunity has been used.

Nearly around three quarters of the about 50 participants came from European countries, two third of them from Germany. Most of the participants from abroad came from the USA (8), others from Canada (2) and Brazil (1). The seminar was successful in strengthening the contacts, in particular, to the American researchers. But also several discussions on possible future joint research projects on all levels (world-wide, European, nation-wide) could be observed outside the regular program of the seminar.

3.4 Computational Cartography and Spatial Modelling

Seminar No. **03401** Date **28.09.–03.10.2003**
Organizers: P. van Oosterom, M. Sester, J. Snoeyink, M. Worboys

Background

The Dagstuhl seminar on "computational cartography and spatial modelling" is the fourth in a series of seminars bringing computer scientists and spatial scientist together. This started with the first seminar (then called Computational Cartography) where Computational Geometers and Cartographers did meet and discuss their problems and potential solutions. With the third seminar, the multidisciplinary aspect has become even larger by adding "Spatial Modelling" to the scope. Scientist and developers with a broader geo-science background on the one hand (geography, geodesy) and a broader computer science background on the other hand (modelling, DBMS) were added to the multidisciplinary group of participants. This fourth seminar, without changing the title, became even more multidisciplinary as the spatial and temporal aspects of mobile computing (including topics such as location-based services and sensor networks) were included in the program. The group of participants was diverse both w.r.t. to 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 technological advances of the recent past, for example, increasing graphics capabilities, multimedia technology, multimodal interaction possibilities, distributed computing, the Internet, wireless communication, new sensors, and efficient geo-data collection techniques, have lead to many new possibilities for interaction with and visualization of spatial data. These advances are currently hampered by lack of suitable algorithms as well as limited understanding of the possibilities of human interaction with spatial data. In the spatial modelling and analysis domain, the field is lacking an integrated approach to deal with (3D) space, time, attributes and their interrelations. Multi-scale issues complicate matters even more, because certain patterns or processes only show up or play a role at specific scales. Most studies so far have concentrated on at most two of the issues: (3D) space, time, attribute, and scale. During the seminar several presentations tried to address more

of these issues at the same time; e.g. 3D generalization (scale) based on the classification and attributes of the object (and task/context of the user). As stated above, this year's seminar also covered the application to portable computing: e.g. location based services and mobile GIS, where low-bandwidth, limited display capabilities require new thinking on aspects such as computational support and human computer interaction. Furthermore, cognitive aspects, context awareness as well as user preferences and privacy issues have to be respected and integrated.

Program

The presentations were organized into a number of sessions with related topics. Some of the presentations covered a single aspect of the theme for the seminar and others addressed several aspects within one presentation. Both types of presentations were very interesting and contained the statement of (new) problems and solutions in a single or multidisciplinary context. The presentations focused on:

1. computational geometry (3D Hilbert curves, new dynamic spatial indexing, definition of valid polygons, equal polygon subdivision, conveyor belt-assisted path planning, etc.),

2. geographic reasoning (mathematical concepts, matroids, cognitive aspects, geo-processes),

3. database and topology issues (topology rules, persistent topology storage),

4. 3D models (TIN/TEN-based, reconstruction/manipulation terrains and objects; e.g. buildings including roofs),

5. generalization (also in 3D, buildings), and

6. mobile/kinematic GIS (sensor networks, LBS, privacy aspects, context-aware map agents).

Due to the efforts of both the presenters and the audience, the disciplinary boundaries were crossed many times and this resulted in refreshing discussions. This was directly after the presentations, but also during the breaks in the pleasant environment of the Schloss Dagstuhl, there was sufficient time to go into more detailed discussions. It has been a very fruitful meeting for all participants. The meeting place traditions (problems/ challenges sessions, ample time for questions and interactions between talks, environment: library, computer room, common rooms, etc.) indeed helped to break down barriers imposed by academic disciplines. Some of the new research results presented were obtained in collaborative projects (and cooperations) which started after the previous Dagstuhl seminar.

Outcomes include

- A collection of abstracts, presentations (slides) and a number of papers surveying the current state of the art in this field and latest research initiatives (available on the Materials page).
- 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 **Open Problem List** (`ftp://ftp.dagstuhl.de/Proceedings/03/03401/03401.OpenProblemSession.pdf`).

3.5 Cognitive Vision Systems

Seminar No. **03441** Date **26.10.–31.10.2003**
Organizers: H. Christensen, H.-H. Nagel

Early attempts to integrate AI and Computer Vision failed due to lack of robust vision techniques for the derivation of symbolic descriptions of the 'meaning' of images, and the lack of AI techniques to handle information with associated uncertainty. Over the last decade, significant progress has been achieved in Computational Vision, AI, and computer platforms.

Regarding Computational Vision, the basis in terms of generating a representation of the system environment through use of robust methods is not yet particularly strong. At the same time, the AI community has established new paradigms for handling uncertain information and scalable models. In parallel to these developments, the progress in the design and production of highly integrated circuits and computer programming systems has resulted in a system performance that facilitates real-time generation and processing of information even from video input streams.

The seminar discussed models for Cognitive Vision Systems (CVS) in terms of system layout and components. In addition, both Computer Vision and AI techniques as components of systems were presented. This seminar also involved discussions on the conceptual basis for Cognitive Vision and the feasibility of constructing computational systems that have 'cognitive' functionality.

Chapter 4

Artificial Intelligence, Computer Linguistic

4.1 Plan-Based Control of Robotic Agents

Seminar No. **03261** Date **22.06.–27.06.2003**
Organizers: M. Beetz, J. Hertzberg, M. Ghallab, M. Pollack

Background

In recent years, autonomous robots, including Xavier, Martha, Rhino, Minerva, and Remote Agent, have shown impressive performance in long term demonstrations. In NASA's Deep Space program, for example, an autonomous spacecraft controller, called the Remote Agent, autonomously controlled a set of scientific experiments in space. At Carnegie Mellon University, Xavier, another autonomous mobile robot, has navigated through an once environment for more than a year, allowing people to issue navigation commands and monitor their execution via the Internet. In 1998, Minerva acted for thirteen days as a museum tour guide in the Smithsonian Museum, and led several thousand people through an exhibition.

These autonomous robots have in common that they perform plan-based control in order to achieve better problem-solving competence. In the plan-based approach robots produce control actions by generating, maintaining, and executing a plan that is effective and has a high expected utility with respect to the robots' current goals and beliefs. Plans are robot control programs that a robot can not only execute but also reason about and manipulate. Thus a plan-based controller is able to manage and adapt the robot's intended course of action – the plan – while executing it and can thereby better achieve complex and changing goals. The use of plans enables these robots to flexibly interleave complex and interacting tasks, exploit opportunities, quickly plan their courses of action, and, if necessary, revise their intended activities.

Content

The first Dagstuhl seminar on "plan-based control of robotic agents" took a technological view and provided us with an overview of recent developments in the plan-based control of autonomous robots. We identified a number of computational principles that enable autonomous robots to accomplish complex, diverse, and dynamically changing tasks in challenging environments and seen a variety of ways to incrementally advance the existing techniques.

Unlike the first seminar, the primary focus of the second seminar has been target problems in the hopes that by investigating these problems thoroughly and identifying the challenges and issues implied by them we will get a better understanding of how the field of plan-based robot control should advance in the next decade. Thus, key questions that we have seeked to answer included, what are the big gaps?, what can we do to close them?, and what are the promising techniques? Our main target problem has been the plan-based control of autonomous household robots.

Thus in the seminar we have considered recent developments in the plan-based control of autonomous robots and identified computational principles that enable autonomous robots to accomplish complex, diverse, and dynamically changing tasks in challenging, uncertain environments. These principles include plan-based high-level control, probabilistic reasoning, plan transformation, formalizations of robot control programs, and context and resource adaptive reasoning.

In the seminar we have worked towards comprehensive and integrated computational models of plan-based control that consider different aspects of plan-based control (plan representation, reasoning, execution, and learning) together and not in isolation. Our hope is that such integrated approaches will enable us to exploit synergies between the different aspects and thereby come up with more powerful computational models.

To achieve these goals we have invited leading experts from areas such as AI planning, plan execution, probabilistic robotics, intelligent control theory, cognitive robotics, robot perception and state estimation, robot learning, and verification of embedded control systems. To focus discussion, we plan to investigate selected applications, such as an autonomous household robot, for which we will provide informal descriptions well in advance.

Main Results

Besides the talks, discussions, and joint research plans started and intensified as part of the seminar two main results have been accomplished:

A roadmap for research in plan-based control of robotic agents.

Michael Beetz was the editor of the roadmap of research in plan-based control of robotic agents. Substantial parts of the roadmap were discussed and produced in the seminar. The roadmap gives an introduction to the field, a framework for integrated plan-based control, and an outline of the projected and suggested lines of research for the next decade. Indeed one of the reviewers of the roadmap (who was not

at the seminar) stated that he could not imagine that such a comprehensive and coherent roadmap could have been produced without the impact of the sessions at the Dagstuhl seminar.

Challenge scenario.

A second important result is a challenge scenario for plan-based robot control. We consider a humanoid robot, such as the Sony SDR-3 or the Honda Asimov, with additional manipulation skills that is to do household chores as an interesting challenge for the field of plan-based control of autonomous robots. The challenge is to develop a plan-based controller for such a robot that enables the robot to be put in another household, to operate in this household for some months, and do a substantial part of the household chores satisfactorily.

The topic "Plan-based Control of Robotic Agents" has become a field of steadily and impressively growing research interest. In particular, the NASA has initiated several well funded research programs that cover our field. Their interest are mainly autonomous space explorers that should be controlled by plan-based control mechanisms. A much bigger program is planned for the next five years. Three of the seminar organizers give academic advice for this program that is to be launched next year. It is vital for our research community in Germany and Europe to continue doing research at an internationally competitive level. The Dagstuhl seminars on "Plan-based Control of Robotic Agents" are important meetings that help us to achieve these objectives.

4.2 Embodied Artificial Intelligence

Seminar No. **03281** Date **06.07.–11.07.2003**
Organizers: V. Hafner, F. Iida, Y. Kuniyoshi, R. Pfeifer, L. Steels

One of the most significant achievements of the Dagstuhl Seminar on Embodied Artificial Intelligence is that a productive worldwide scientific community of the field has been formed, which consists of researchers from many different disciplines such as biology, neuroscience, robotics, computer science, and psychology. The diversity of backgrounds of the community member provides a broader perspective of the most significant open problems and challenges. In addition, the active contributions of the young researchers to the conceptual discussions confirm a promising development of the field toward the future.

One of the main highlights of the seminar was the oral and poster presentations followed by active discussions. The topics covered a range from the "low-level" topics such as materials for robot construction, neuron-cell activities, to the "high-level" conceptual issues of representation and consciousness. Particularly, the trend such as open-ended development of embodied agents leads to highly stimulating debates. Owing to the thoughtful, constructive and original comments and feedbacks by the leading researchers of various fields, every speaker and participant had a unique opportunity to look into not only technical details but also important conceptual issues underlying each research area. From the

dynamics of the participants' interactions, a number of interesting ideas have been generated, which provided excellent materials for the organized discussion sessions reported below.

In addition to the presentations, several emergent discussion sessions had been organized, where conceptual issues of design principles of intelligent agents, methodology/evaluation, and challenges toward the future were discussed. Although the result of these discussions is still somewhat inconclusive given the current developing stage of the field, we have reached a considerable level of consensuses toward establishing the "Theory of Embodied AI".

One of the central discussion topics for the design principles was the notion/definition of embodiment. The main questions were centered around how we could deal with the concept of "embodiment" in the simulated/real-world research, and how to conceptualize it in a theoretical domain.

Very briefly, some of the important aspects can be summarized as follows:
(1) An embodied agent should physically interact with environments.
(2) An embodied agent should exploit the laws of physics.
(3) An embodied agent can structure the sensory information from the environment by properly exploiting the physical interaction.
For the future work, therefore, the design principles of "cheap design", "ecological balance", "sensory motor coordination" and "redundancy" should be the crucial issues. In this sense, a non-modularized design strategy of artificial systems, in other words "everything is connected to everything" fashion design, could be another aspect to be considered. This approach would lead to a more comprehensive understanding of the concept of embodiment.

Methodology and evaluation of the research contributions is particularly important for a successful development of the field. One of the essential strategies for a plausible methodology/evaluation is to preserve the diversity of the studies, whereas some intensive research targets should be explored in parallel.
Examples are:
(1) Repeatable robotic experiments
(2) Comparative studies of embodied agents
(3) Quantification of environments
(4) Performance evaluation for the multiple-task systems.
Moreover, these methodology/evaluation criteria should be considered in the review process of the publications.

Some challenging research issues toward the future have been discussed. Given the common goal of the field is to understand the design principles for a general-purpose system, the current interest of research is mainly how to proceed the research in the direction of a significantly more complex agent which adaptively interacts with a dynamic environment. As one of the approaches toward tackling this problem, the open-ended development of embodied agents has been extensively discussed.
More clearly,
(1) Open-ended evolution for the design of an artifact;
(2) Evolution versus development (i.e. time-perspectives);

(3) Rewards/value-systems that go beyond basic values such as pain and pleasure;
(4) The role of embodiment in the context of developmental process;
(5) Quantification of "ecological balance" (information theoretical and statistical methods).

In addition to an abstract booklet which has been published before the seminar, we have agreed on a few follow-up activities for publicity on the basis of the constructive discussion during the seminar. The publication plan includes a proceeding book from Lecture Note in Computer Science of Springer (LNCS), and making a documentary film of the seminar.

4.3 Robot Navigation

Seminar No. **03501** Date **07.12.–12.12.2003**
Organizers: R. Fleischer, R. Klein, A. Lopez-Ortiz

Autonomous robots are supposed to perform well, even without complete information about their environment. Frequently occurring subtasks include

- to search an environment for a goal,
- to explore an unknown environment, and
- to determine their own position, given a map.

Depending on the type of the robot's sensors, and on its a priori knowledge about the environment or the position of the target, these – and other – tasks give rise to a variety of on-line navigation problems.

A solution for a problem P consists of a strategy S that solves correctly all instances of P. It is called competitive with factor c if it solves each instance p of P at a cost not bigger than c times the cost of solving p optimally (for example, we would compare the length of the robot's path from the start to the goal against the length of the shortest such path that exists in the given environment).

Given a navigation problem P, two questions arise. Can P be solved by a competitive strategy with a constant factor c? And, if so, what is the competitive complexity of problem P, that is, the smallest possible factor c? These questions are mostly of theoretical nature. They can be studied independently of the more technical problems in robotics, like systems design, fusion of sensor data, or dealing with inaccuracies. But we hold that gaining a better understanding of these navigation problems is not only an interesting theoretical challenge; it might also provide solid ground work for the development of future robots.

Due to their particular nature, on-line navigation problems have been studied independently by researchers in three scientific communities: by some more theoretically-oriented electrical engineers in robotics, by people in on-line analysis, and by computational geometers. This is reflected by the corresponding Dagstuhl seminars. But in none of these

seminars have on-line navigation problems received the attention they deserve. In this seminar, we brought together leading experts of all three groups. We reviewed and discussed the state of the art, and we tried to identify important problems of common interest.

25 researchers with affiliations in Canada (3), Germany (9), Hong Kong (2), Israel (3), the Netherlands (4), Slovenia (1), and the USA (3) participated in the meeting. Seven participants were graduate students or postdocs. Four keynote speakers, Rudolf Fleischer, Vladimir Lumelsky, David Mount, and Mark Overmars, gave one-hour survey talks. The remaining 19 presentations given by participants of the meeting covered a wide range of topics, ranging from robot path planning, search and exploration problems, to algorithmic issues in practical robotics. In an evening session we discussed important open problems.

Chapter 5

Programming Languages, Compiler

5.1 Emerging Technologies: Can Optimization Technology meet their demands?

Seminar No. **03071** Date **09.02.–14.02.2003**
Organizers: T. Conte, C. Eisenbeis, M.L. Soffa

The talks at this seminar on optimization generally focused on 5 topics: program analysis, code optimization, dynamic analyis and optimization, new infrastructures for compilers, and embedded systems/architectures. The underlying questions during the seminar were (1) whether the issues and techniques presented were actually new or only rephrasements and rediscoveries of prior work and (2) what is realistically feasible in the next 5 – 10 years.

With this respect three panels were organized:

Program analysis:

static? dynamic?

Embedded processors:

Do they provide the compiler community with new challenges? Some participants argued that new criteria have now to be taken into account such as power saving, code size, WCET. Hand coding is common practice in embedded systems. The fact that embedded architectures are less symmetric makes the code generation process tightly coupled with the optimization process. Also data transfers between memories must be managed by the compiler. But are new optimization techniques needed or can we modify existing ones to accommodate the needs of embedded system software? There seems to be a lack of transfer of information or know-how between different generations of researchers and the compiler and embedded communities. There was also a question of whether optimization had developed enough that a scientific foundation could be laid and used with embedded systems. There was concern expressed about the difficulty of teaching optimization because material is spread across many different venues. One participant argued that researchers should write books on optimizations.

Dynamic program optimization:

Is dynamic program optimization the best way and the only way to go beyond the limits of the traditional static optimization? Also, what form should dynamic optimization take? The discussion was fairly negative about the overhead of on the fly analysis and optimization. It was argued that its utility clearly depends on the mobility and adaptability requirements of the code and on the behavior of a program. Which are the actual performance improvements that can be expected from these methods? Performing run time optimization is actually designing a new run time process that could be considered as a "software processor" very much like out-of-order processors that have "dynamic optimization" features. The central question is, therefore, where to place the frontier between the program and the runtime system.

Apart the panels, the trends that we can identify after this seminar are:

- a trend to consider the system (program, processor, compiler) from outside, analyse it very much like a biological entity (by running systematic experiments and analyse performance curve or graphics), observe its reactions to stimuli (iterative compiling methods are such an observation system)
- great need and trend to go towards formal methods, not only for safety critical systems but also for generating stable (bug free) optimizing compilers for ordinary processors.
- great need that compilers and architects communicate, especially in the domain of embedded systems.

Chapter 6

Software Technology

6.1 Software Architecture: Recovery and Modelling

Seminar No. **03061** Date **02.02.–07.02.2003**
Organizers: A. v. Deursen, R. Kazman, R. Koschke

Description of the seminar topic

Software architecture is recognized as a critical element in the successful development and evolution of software-intensive systems. In 1995, the Dagstuhl Seminar No. 9508 on Software Architecture (organizers: D. Garlan, F. Paulisch, W. Tichy) was held. In retrospection, the seminar has had a tremendous impact on both research and practice of software architecture. Today, we have workshops and conferences as well as many books on software architecture. The IEEE has recently set a standard on recommended practice for architectural description of software-intensive systems (IEEE-Std-1471-2000). Methods and notations exist to model software architectures during system design. Techniques are being developed to reconstruct architectural views of existing legacy systems in the postdelivery phase of a system's life cycle. However, despite the many achievements, there are still many open research questions. In this Dagstuhl seminar, we will concentrate on the relation between modeling an architecture of a new system and reconstructing and evolving an architecture of an existing system. To a large extent, these two different aspects have been examined separately in two hardly overlapping research communities of forward and reverse engineering.

The designed architecture is used to validate whether all requirements can be fulfilled and is then implemented. The architecture description helps to communicate with customers and programmers. When the system is built, new requirements may arise and the system needs to evolve. The necessary changes and their potential impact of these changes on the system are analyzed based on the designed architecture. Typically, the stakeholders involved during a system's evolution are different from those during development and they have different requirements. Maintainers need a detailed view of the architecture as-built, i.e., the one that the system actually has, as opposed to the architecture that was originally

designed. The designed architecture and the architecture as-built far too often diverge due to ad-hoc changes and changes that are not properly documented.

Software architecture recovery aims at reconstructing views on the architecture as-built. Architecture recovery research issues include many topics, such as system browsing, dealing with multiple perspectives, visualization, usability evaluations, reference architectures, interface description, re-modularization, and so on.

The overall goals of the Dagstuhl seminar on Software Architecture Recovery and Modeling is to bring together researchers as well as practitioners from the two domains of modeling and recovering software architecture to exchange experiences, discuss new application areas, discover areas of mutual collaboration, and to envision future trends in the field of software architecture recovery and modeling. More specific topics are:

- progress of the last six years since the Dagstuhl seminar on software architecture
- architecturally relevant information for development and evolution
- methods, techniques, and tools to design software architectures
- methods, techniques, and tools to recover architectural views
- notations to capture and specify models of designed architectures and architectures as-built
- means to resolve or capture and tolerate differences between designed architecture and architecture as-built
- aspects that ease architectural reconstruction and evolution to be considered during development

Expected Results

The expected results of this seminar are a report on the comparison of the state of the art in architecture recovery and architecture modeling, an identification of the discrepancies between constructing and re-constructing architectures, a research agenda on how to overcome these discrepancies, and established partnerships of mutual collaboration in the recovery and modeling communities to tackle this agenda.

6.2 Program Analysis for Object-Oriented Evolution

Seminar No. **03091** Date **23.02.–28.02.2003**
Organizers: F. Tip, G. Snelting, R. Johnson

Maintenance and restructuring are activities that have traditionally been associated with "legacy" languages such as Cobol and PL/I. However, with the increasing use of object-oriented languages for large-scale industrial projects, the same activities are now often required in the object-oriented domain as well. But due to the complexity of advanced

object-oriented software development, existing techniques for maintenance and restructuring procedural programs are not adequate.

In order to tackle the challenges of object-oriented maintenance, the Dagstuhl seminar "Program Analysis for Object-Oriented Evolution" brought together two groups of scientists: the program analysis community and the refactoring community. Program analysis has a long tradition and has recently been used extensively to support maintenance activities. Refactoring is a new approach to improve object-oriented designs by applying a sequence of semantics-preserving transformations.

The workshop featured a series of presentations about state-of-the-art program analysis and refactoring technology, as well as extensive discussions about mutual benefit. As an overall result,

- Program analysis researchers now do understand current problems in evolution and restructuring of object-oriented programs. They do understand that the principle of conservative approximation, which is essential in traditional program analysis, can be softened in a refactoring context.
- Researchers in evolution and refactoring now do understand the possibilities provided by the state-of-the-art in program analysis. They do understand that program analysis can provide the semantic guarantees needed for successful refactorings.

As a consequence, we expect many new research projects utilizing these insights. Some such projects have already been started. These projects open the door for safer and more powerful refactorings, providing more reliable and efficient evolution of object-oriented systems.

The workshop featured 30 presentations. There were 4 outstanding keynote presentations on program analysis and refactoring. The other talks presented ongoing research. Two discussion sessions culminated in a collection of open research topics. Half of the talks were given by young scientists (graduate students or post-docs). These young researchers had excellent opportunity to discuss their work with the more senior participants, thus obtaining many valuable insights. About half of the participants (and more than half of the young researchers) were from Europe.

6.3 Domain-Specific Program Generation

Seminar No. **03131** Date **23.03.–28.03.2003**
Organizers: D. Batory, C. Consel, C. Lengauer, M. Odersky

Public Statement

Generative approaches have the potential to revolutionise software development as automation and components revolutionised manufacturing. This technology is particularly

effective when applied with domain-specific techniques, since compact, domain-specific notations make programs easier to write and maintain and domain-specific knowledge allows for a more efficient implementation. The purpose of the seminar was to promote scientific cooperation on the topic of domain specific program generation. This topic has been pursued so far in a number of research communities which had insufficient contact with each other:

Domain-specific languages:
Language developers in a specific application domain have often been unaware of the domain-independent aspects of their domain-specific work. Vice versa, researchers that do not work in a specific domain are often unaware of the experiences made in application work.

High-performance parallelism:
This is one application domain, which has led to the development of a particular form of domain-specific language (so-called skeletons). Researchers in this community have been quite unaware of the wider aspects of domain-specific program generation.

Program generators:
This domain is concerned with the fast and reliable generation of members of a program family (so-called "product lines"). Researchers in this community are often in industry.

Metaprogramming:
Rersearchers in this community develop a technology that can be used for customizing compilation and translation systems for domain-specific purposes. As a main result of the seminar, initial steps were taken to form a working group. Also cooperation on an individual basis was fostered. A compendium of papers presented at the seminar is in preparation for the Lecture Notes in Computer Science series of Springer-Verlag.

Scientific Highlights and Perspectives

The seminar contributed significantly to the formation of a new community around the seminar topic:

- People from different language backgrounds – notably imperative languages (main representative: C++) and declarative languages (main representative: Haskell) – got to know each others' work. People were very receptive of each other: everybody went away with a wider horizon; new cooperations were formed at several levels; some led to newly formed joint authorships for the compendium.
- Talks on applications made a special impact. In particular, the community on high performance parallelism (the "skeletons" community) was recognized as a domain with a special need for optimization. The integration of this domain in the wider context of the research community is a major result of the seminar.

- An outcome of this seminar was the foundation of a new IFIP WG 2.11 "Program Generation" http://www.cs.rice.edu/~taha/wg2.11/

Presentation and Publication

Talks altogether: 35; Keynotes: 3; Talks of young researchers: 4.

23 submissions have been announced by participants for a planned compendium on the seminar topic. The submissions are based on talks given or discussions conducted at the seminar. Springer-Verlag has been approached for publication. The book will appear in May 2004: Domain-Specific Program Generation; Christian Lengauer, Don Batory, Charles Consel, Martin Odersky, eds.; Lecture Notes in Computer Science 3016, Springer-Verlag, 2004.

Working Group on Domain-Specific Program Generation: Mission Statement

Generative approaches have the potential to revolutionize software development as automation and components revolutionized manufacturing. Such approaches are particularly effective when combined with domain-specific techniques, since compact, domain-specific notations make programs easier to write and maintain, and domain-specific knowledge allows for a more efficient implementation.

The aim of this Working Group of researchers and practitioners is to promote innovation in

- foundations
- design
- engineering
- techniques
- tools
- applications

for domain-specific program generation.

The scope of this Working Group covers all aspects of design, analysis, generation, and quality control of generative programs and the programs that they generate, with emphasis on the use of domain specific knowledge. Specific research themes include (but are not limited to the following areas):

- Foundations: language design, semantics, type systems, formal methods, multi-stage and multi-level languages, validation and verification.

- Design: models of generative programming, domain engineering, domain analysis and design, system family and product line engineering, model-driven development, separation of concerns, aspect-oriented modellig, feature-oriented modelling.
- Engineering: practices in the context of program generation, such as requirements elicitation and management, software process engineering and management, software maintenance, software estimation and measurement
- Techniques: meta-programming, staging, templates, in-lining, macro expansion, reflection, partial evaluation, intentional programming, staged configuration, stepwise refinement, software reuse, adaptive compilation, runtime code generation, compilation, integration of domain specific languages, testing.
- Tools: open compilers, extensible programming environments, active libraries, frame processors, program transformation systems, program specializers, aspect weavers, and tools for domain modelling.
- Application: IT infrastructure, finance, telecom, automotive, aerospace, space applications, scientific computing, health, life sciences, manufacturing, government, systems software and middle-ware, embedded and real-time systems, generation of non-code artefacts.

The objectives of the Working Group are:

- Foster collaboration and interaction between researchers from domain engineering, and on language design, meta-programming techniques, and generative methodologies.
- Demonstrate concrete benefits in specific application areas.
- Develop techniques to assess productivity, reliability, and usability.

6.4 Product Family Development

Seminar No. **03151** Date **07.04.–10.04.2003**
Organizers: G. Böckle, P. Knauber, F. van der Linden, L. Northrop, K. Pohl

Product family engineering is a new paradigm in software engineering research, which promises high quality software products at lowered cost and shorter time schedules. The key idea is to emphasize proactive reuse, interchangeable components, and multi-project planning cycles, similar to practices applied for a long time for example in car manufacturing. Product family engineering has recently gained much interest in various application domains including electronic commerce, information systems, medical systems, and telecommunication systems. Product family engineering focuses on the creation and maintenance of a whole set, i.e., a family, of software products and software-intensive systems. A distinction is made between development *for* reuse (called *domain engineering*) dealing

with planning, creation, and maintenance of system assets (development artifacts) common to the various application systems and development *with* reuse (called *application engineering*) where the parts that are specific to particular applications are handled.

The seminar built on the results of the very successful Dagstuhl Seminar on Product Family Development held in April 2001 (Seminar No. 01161). The focus of the first seminar was on the technical aspects of product family engineering, while the focus of this seminar was on management and organizational aspects. The objective of the seminar was to cross-fertilize and synthesize the work done by the different universities, research institutes, and industrial research groups working on product family engineering. The topics of the seminar were strategies for product family adoption, organizational forms that support product family development, testing of product family assets and application specific extensions, production of customer-specific applications, product family maturity, and lessons learnt from industrial product family development.

This Dagstuhl Seminar brought together twenty-five leading practitioners and researchers from various disciplines to cross-examine the effectiveness and the efficiency of product family based software system development. The seminar was organised by Günter Böckle (Siemens AG, München), Peter Knauber (University of Applied Sciences, Mannheim), Frank van der Linden (Philips, The Netherlands), Linda Northrop (Software Engineering Institute (SEI), USA), and Klaus Pohl (University of Duisburg-Essen).

The seminar was structured into two parts: an overview talk part over half a day and a working group part that took 2-and-a-half days.

Overview Talks

At the beginning of the seminar, a set of **plenary talks** provided overviews on various aspects of software product family engineering:

- Frank van der Linden (Philips, The Netherlands) explained the view and the latest results of the European major software product family initiative, the ITEA project CAFÉ and FAMILIES.
- Linda Northrop (SEI, USA) complemented this overview talk by providing an overview on the results and actual research plans of the software product line initiative by the Software Engineering Institute (SEI).
- Jan Bosch (University of Groningen, The Netherlands) gave a talk on software variability management. In his talk he discussed the problems and issues of variability management and distinguished different levels of maturity in variability management.
- Kari Känsälä (Nokia, Finland) held a talk on maturity assessment and the specific practices at Nokia. An extension of the CMMI called CMMI-SFE (System Family Engineering) was suggested, which specifically aims at V&V activities.

- John McGregor (Clemson University, USA) provided in his talk an overview on product family testing. The specific problems of testing product families were discussed and practices were presented for system, integration, and component testing.
- Gary Chastek (SEI, USA) dealt in his talk with production plans, which are descriptions of how core assets are to be used to develop a product in a product line. A product plan ensures that product developers can make effective use of the core assets.

Working Groups

After a brainstorming session and further discussions, the participants identified four main topics for parallel working groups, namely:

- Economic Models for Software Product Lines
- Software Product Family Variability
- Product Line Adoption
- Software Product Family Maturity

Within the parallel working groups these topics where discussed. The groups identified common grounds and synthesized their views on these topics.

Each working group gave an intermediate and a final presentation of their results in a plenary session. Moreover, a 1-page summary was written by each group, and outlines for conference and journal papers have been sketched. After the seminar, four papers have already been finalised and submitted for publication.

6.5 Perspectives Workshop: "Software Optimization"

Seminar No. **03351** Date **26.08.–29.08.2003**
Organizers: S. Graham, R. Wilhelm

The area of Software Optimization, in the context of compilers called *Code Optimization*, is not in a good state. Although there is continuing research on this topic, it is largely incremental in nature. There has been little progress in the foundational areas. The relationship to language semantics has not been substantially clarified, and metrics have not been developed to determine the profitability of program transformations, except in very specific instances. The same holds for attempts to mechanize the program transformation task – the needed specification and generation mechanisms are lacking. New architectural concepts undermine traditional separations between machine-dependent and machine-independent optimizations, casting doubts upon established transformations and

requiring the creation of new ones. New languages with dynamic program reconfiguration shift tasks from compile time to run time.

The time has come to step back from current research and to lay out a longer-term research agenda that identifies both the nature of the contemporary and future contexts for optimization and the important problems that need to be addressed. To identify that agenda, this Vision Seminar on Software Optimization will be held in Dagstuhl.

Areas to be represented are the following:

Semantics preservation
Program Analysis

Theory of Program Transformation
Transformation Mechanisms
Program Representations

Metrics, Profitability
Architecture Awareness
OS Awareness
Profiling, Benchmarking
Feedback-Directed Optimization

Influence of Language Design
Interaction with SW-Engineering

Run-Time Adaptation
Run-Time Optimization

Transform. SW Development

Industry Needs Assessment

General Expertise

6.6 Scenarios: Models, Transformations and Tools

Seminar No. **03371** Date **07.09.–12.09.2003**
Organizers: F. Bordeleau, S. Leue, T. Systä

Background

The Dagstuhl seminar on "Scenarios: Models, Transformations and Tools" was organized as a continuation of a series of workshops that were co-located with larger conferences such as International Conference on Software Engineering (ICSE) and Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) since year 2000. In these workshops we had only a limited amount of time (one day) to discuss the various research problems in this field, which motivated us to apply for arranging a Dagstuhl seminar on the theme. One of the main reasons for the increased interest towards the workshops was the wide spectrum of application domains of scenario-based software modeling techniques.

Scenarios are used in telecommunications as message sequence charts, in object-oriented system design as sequence diagrams, in reverse engineering as execution traces, and in requirements engineering, e.g., as use case maps or life sequence charts. These techniques are used to capture requirements, in particular on reactive systems, to capture use cases in system documentation, to specify test cases, or to visualise runs of existing systems. They are often used to represent concurrent systems that interact via message passing or method invocations. In telecommunications, for more than 15 years the International Telecommunications Union has standardised the Message Sequence Charts (MSCs) notation in its recommendation Z.120. More recently, with the emergence of UML as a predominant software design methodology, there has been special interest in the development of the sequence diagram notation. Both MSC and UML 2.0 interaction diagrams, in addition to other scenario notations, were intensively discussed in the seminar.

Program

The presentations were organized into a number of sessions of related topics. The presentations in each of the Sessions can be summarized as follows:

UML:

Interactions in different forms are part of UML 2.0. The problems of agreeing on a common semantics as well as insufficiencies in the semantics of UML Sequence Diagrams were discussed.

Formal Analysis and Semantics:

This session presented a wide range of formal analysis technique either addressing scenario notations, or using scenario notations to capture requirements. These included work on checking the compliance of the behaviour of UML models with a given set of scenarios; a review of syntactic and semantic analysis techniques for sequence diagrams, including decidability and complexity results for different model checking problems; a presentation of the play-in/play-out approach to using collections of sequence charts; an overview of different temporal logics for sequence charts and the complexity of their model checking problems; the use of sequence charts in the visualization of security protocols; and finally an introduction into the concept of shared variable interaction diagrams.

Design:

This session described an approach to synthesizing state machine models from scenarios given as UML use case diagrams; the use of scenarios presented as use case maps in the derivation of performance models; the synthesis of performance models and test cases from message sequence chart specifications; a method to derive behaviour trees from sets of requirements given as scenarios; an approach to infering class behaviour from instance descriptions given as message sequence charts; a characterization of temporal interval relationships as expressed by scenario diagrams; a game theoretic approach to the synthesis of operational models from sequence chart specifications; and a synthesis approach based on the scenario based description of coordination patterns between software roles.

Testing:

Work on the modeling and testing of systems based on ASML specified scenarios as well as the joint use of MSCs and TTCN3 for the specification of tests were discussed in this session.

Synthesis:

This session presented work on specifying dynamic, mobile systems using extensions of sequence and activity diagrams; on refining MSC specifications into models of communicating finite state machines under special consideration of the resulting communication channel structure; and on algorithms for the synthesis of operational models from service specifications given as sequence diagrams.

During one afternoon session and one additionally scheduled night session a requirements capture and documentation case study was performed. Based on a number of requirements on an autonomous shuttle transport system four groups of participants were formed to apply their favourite scenario based modeling technique and, if possible, a related tool to capture and formalize these requirements. One participant with particular familiarity in the case study acted as the oracle and answered additional questions during the sessions. As expected and hoped for, the case study helped participants to understanding the essence of the tools and methods and their intended use and created lively discussions on the results. The intent of the on-line case study was not to aim at a competitive comparison but rather learn from different methods and tools and open lines for future collaboration. A number of groups arrived at synthesizing executable models from the requirements during the case-study sessions.

The audience was very active during all the sessions, creating an interactive atmosphere. Since sessions had time limitations, although somewhat relaxed ones, the discussions were continued over the coffee and lunch breaks, as well as during the evenings in the customary pleasant atmosphere of Schloss Dagstuhl.

Outcomes

The outcome of the Dagstuhl seminar 03371 includes the following:

- A collection of abstracts, presentations (slides) and a number of papers surveying the current state of the art in this field and latest research initiatives (available on the Proceedings page).
- A list of main open research problems.
- A plan for future work.

The list of main open research problems in the field proposed during the seminar contains the following items:

Integration of scenario based synthesis in an iterative software development process.
Incorporating component structure in scenarios and related synthesis approaches.

Clarifying and reserving intention throughout the software development process.
Integrating the different notations, semantics, and approaches.
Synthesis for testing and simulation in the presence of data models.
Scalability to real world complexity.
Taking advantage of the structure of scenario model (with temporal ordering) in the synthesis and verification process.
Enable the description of system dynamics (different dynamic system aspects) in scenario models.
Enable the use (reuse) of existing components in scenario models.
Introduce QoS in the overall design process, starting with scenario models.
Integration of temporal modalities expressiveness in scenario notations while maintaining intuitive, visual appeal.
Roadmap for scenario technique usage in the context of the overall development processes.
Round-trip tool support.
Tool integration (development of a common tool integration framework).
Improve the algorithmic aspect (improve existing algorithms and develop new ones).
Traceability and consistency between models.

A plan for future work was discussed during the final day of the seminar. It was agreed that a summary of the results of the case study would be useful, possibly accompanied with the application of other tools not presented in the seminar. The exact form of the summary was left open. It was also decided that another workshop on the theme was proposed to be held in connection with ICSE 2004.

The local Web page of Dagstuhl seminar 03371 includes the final program of the seminar. Springer Verlag has meanwhile agreed to publish a peer reviewed post seminar proceedings volume in its Lecture Notes in Computer Science series. It is expected that this volume will appear during the second half of the year 2004.

6.7 Software Intensive Embedded Systems – with Special Emphasis on Automotive

Seminar No. **03461** Date **09.11.–14.11.2003**
Organizers: M. Broy, R. Ernst, U. Goltz, L. Lavagno

In a modern car a network of up to 80 electronic control units (ECUs) realises several hundred functions that range from power train control, active and passive safety systems, body electronics and driver assistance to infotainment applications. High demands on quality, reliability, the increasing complexity, and the rapidly growing number of interactions between subsystems, as well as time-to-market and cost constraints lead to challenging requirements for new processes, methods and tools. The automotive area is a particular interesting field for the application of design processes and methods being developed for software-intensive embedded systems. The seminar covered a wide range of topics (see

below). As aspects from such a wide field are relevant for the development of software-intensive embedded systems for a mass market, the intensive interdisciplinary exchange of ideas between experts from electrical and mechanical engineering and computer science is the most promising approach to progress.

The seminar had a dense program of presentations followed by active discussions. Topics were:
design and development processes
tool support (framework and chains)
software integration (technical and legal)
software layer techniques (middleware)
performance analysis (timing constraints)
modelling, analysis, and validation methods
diagnosis and test
model based software and system design
model driven architecture
model checking and verification
requirements on safety critical and embedded systems
programmable networks

The seminar provided a good overview of the international activities in industry and universities in (automotive) embedded software design. As a result, the participants could establish contacts and coordinate their activities. Finally the necessity for further meetings was affirmed. Only by an intensive knowledge exchange it is possible to master the increasing challenges.

6.8 Understanding Program Dynamics

Seminar No. **03491** Date **30.11.–05.12.2003**
Organizers: J.-D. Choi, B. Ryder, A. Zeller

Public Outreach

Understanding what is going on in a program run has been a problem for decades. Classically, program analysis has been divided into two areas:

Static analysis deduces from program code what can (and what cannot) happen in all possible program runs.

Dynamic analysis observes facts in a concrete program run and possibly checks whether these facts meet specific expectations.

However, the gap between "static" and "dynamic" is no longer as wide as it used to be. Both techniques are being extended to incorporate each other's strengths:

- Coming from the static side, more and more analysis techniques make use of symbolic execution (which makes them "dynamic" in some sense) and thus may restrict their range to a specific set of runs in order to increase precision.

- Coming from the dynamic side, analysis need not be restricted to a single run, but to a multitude of runs (possibly even conducted by the analysis process), thus broadening the applicability of its results.

The goals of this Dagstuhl Seminar were to further *bridge the gap* between "static" and "dynamic" analysis – and to explore new directions that would help *integrating* the strengths of the different approaches.

Scientific Highlights

The seminar succeeded in both goals:

Bridging the gap.
At the end of the seminar, all researchers, whether working on "static" or "dynamic" methods, agreed that any information about programs can (and should) be exploited to improve their understanding. This information includes the program code and its semantics, of course, but also program traces, test results, test coverage, program usage in the field, version histories, and other accessible data. The seminar participants demonstrated an impressive range of techniques to exploit these information sources.

Technique integration.
There is an enormous wealth of information about programs that is accessible today. All this data needs to be filtered, combined, and distilled – a task only possible by integrating various "static" and "dynamic" techniques. The integration of the participants' approaches opens up several opportunities to improve program understanding – and this seminar was an excellent starting point to make people meet and work together.

Perspectives

All in all, this seminar has exceeded the organizers' expectations by far – both in terms of creativity and in interaction. Yet, we have only begun to exploit the wealth of information about programs. Several questions offer opportunities for further research, including:

- How do we gather abstractions from concrete runs - from test runs or runs in the field?
- How can such abstractions guide static analysis?
- How can we distinguish the facts that are relevant for a specific behavior?

In addressing these questions, computing power is no longer the limit. Instead, we must find out how to make the best of our tools and techniques. This seminar has turned out several promising approaches.

The spirit of this seminar will live on in future events dedicated to integrate various approaches to program analysis. In particular, we expect the Workshop on Dynamic Analysis (WODA) and the Workshop on Program Analysis for Software Tools and Engineering (PASTE) to show up first integration results. In a year from now, we shall send an informal questionnaire to the participants, asking them how the Dagstuhl seminar has influenced their later research.

Chapter 7

Applications, Interdisciplinary Work

7.1 Information and Process Integration: A Life Science Perspective

Seminar No. **03051** Date **28.01.–31.01.2003**
Organizers: R. Apweiler, T. Etzold, J.-C. Freytag, C. Goble, P. Schwarz

This seminar brought together scientists and industrial developers and researchers to discuss the challenges of integrating bioinformatics/life science data in a meaningful way. Despite the technological advances many open problems and issues persist and need to be addressed. This workshop focused on the main issues of data and process integration in the life science domain.

The result of the seminar showed that integration is still wide open field based on the differences in technology, the expectations by the users, and the kind of problems that biologists and life scientists try to solve. It became apparent that often the integration task is driven by the specifics of the application (lab protocols and their mapping onto computer systems). The discussions also made clear that integration must include semantic integration, in particular the meaningful integration of different space and time scales (microseconds vs. millions of years) and the presentation of discrete and continuous data (the former is well understood, the latter is an open area). Another open (biological) issue is the use of measurements which are often not reproducible, thus making it difficult to compare and to use. Finally it became apparent that biologists and computer scientists must cooperate much closer to solve the complex problems that exist in life science and are about to appear on the (scientific) horizon.

Detailed Agenda for the Workshop

Over the last fifteen years the amount of data in the area of Life Science/Bioinformatics has grown exponentially. This data is stored and is available in an ever increasing number of data collections (also often referred to as databases), each focusing on specific aspects

of life science, such as nucleotide or protein sequences, functional motifs, metabolic pathways, specific organisms, or information related to specific diseases. At the same time the bioinformatics community has developed hundreds of tools to visualize, to analyze, and to process that data, with the goal of turning raw data as produced by sequencing machines into knowledge applicable to drug design and to the development of new therapies. Examples include gene prediction, motif recognition, the computation of phylogenetic relationships, and the deduction of pathways from gene expression arrays. However, almost all of these tools use proprietary, non-standard data formats thus making it (almost) impossible to change those or to introduce new tools without recognizing the need for bridging the gap between the existing world of data and processing conventions and new promising approaches.

With the advent of middleware technology, the focus of research and development in data integration has begun to shift. While many previous efforts have addressed the syntactic integration of data collections, the real challenge now, and for years to come, will be the development of new approaches, techniques, methods and algorithms for performing **semantic integration**. What will be needed are systems that bring together data that belong together, making this determination on the basis of both structure **and** meaning. To achieve this goal, current middleware technology will need to be extended so that it can take advantage of ontologies, semantic networks and other metadata (e.g. information about data quality) to gain a deeper understanding of the primary data.

The problems described are present in both academic and research institutions as well as in pharmaceutical, drug design, medical, and health care businesses. Only the use of modern technology promises the users a platform to bring diverse data, information, knowledge, and processing software together to advance science and to satisfy business needs. If the current time necessary for the development of a new drug, which is estimated to be at app. $10 - 15$ years, is to be reduced fundamentally, the process from molecular biology evidence to clinical studies has to be highly streamlined, which requires a tight yet flexible intertwining of a multitude of databases and applications.

This seminar should bring together scientists and practitioners from the fields of bioinformatics and information technology, in order to better understand the new challenges as well as existing approaches and relevant technologies. Solutions to the new problems will most likely be driven by extending existing technology (e.g. Object-Relational DBMS) to meet new needs (e.g. federated database management, highly-parallel distributed problem-solving on a grid), emerging tools and standards for managing semi-structured data (e.g. XML, XQuery, XSchema) and process technologies (e.g. CORBA, Java Beans, message-driven workflow using Web Services).

New technology areas such as the onlotogies, the **Semantic Web** and the Grid are highly applicable to a more meaningful integration of data, information, and processes for Life Sciences. It becomes important that mutual understanding in both the research and business world arises to make the necessary advances in bioinformatics. Still, it is time to evaluate the current solutions and approaches to drive future research and development directions by the pressing needs of the bioinformatics/life science community.

The areas to discuss include:

- Achieving semantic integration

  - What are today's approaches for semantic integration? Are those sufficient for the life science domain?
  - What are the necessary concepts such ontologies that are necessary to perform semantic integration?
  - What are the languages required to specify the various forms of biological and medical knowledge that is required for bioinformatics research? Are relations and attributes really enough?
  - Which knowledge management techniques (personalization, community building, knowledge sharing, text mining) are appropriate to the Life Science area?
  - How to ensure data quality, data consistency, and completeness? How can data quality be compared, assessed, measured, combined?

- Information discovery and publication
  - What is the optimal access form to the various data collections that are important to scientific organization and business in the different life science areas?
  - Can XML be used as the "universal language" for describing the integrated information base? How to capture "navigational access" based on hyper-linked HTML pages performed today in many application areas?
  - Version management for data collections and metadata that change daily/weekly? Are there compression schemes that can reduce the large amount of repeated (redundant) data? How can we efficiently store the relationships between new or changing evidence and new versions of data?
  - How is information described? What are approaches to handle the description of data (metadata)? Which metadata is relevant (schema, ontologies)? How to store and access it? How to keep it current?
  - What is a federated schema if structured and unstructured data are brought together? Which schema integration techniques, federated query and search technologies are applicable?
  - What are possible system structures in a highly dynamic world that constantly changes and that makes constant progress?

- Information processing paradigms
  - Which processing/transaction models are appropriate?
  - How can ontologies and other meta data support more meaningful processing techniques? Are current techniques adequate for distributed query processing? What are new requirements coming from Life Science?
  - How to represent and manage derived data, data quality and data provenance?
  - How do Semantic Web and Grid technologies contribute?
  - Which federated database technologies can be used in which context? Are the trade-offs that provide the bases to decide which approach to choose in a particular situation?

- Information technologies and standardization
  - How to use different technologies like SQL/MED wrappers, J2EE connectors, EAI adapters, and Web Services for virtual or physical integration. Which technology should be used under which circumstances?
  - Which role will database systems, application server, workflow systems, messaging systems, portal servers, etc. play? How do they relate and cooperate?
  - Does Web Database Technology suffice?
  - What is the query/retrieval interface for the future?
  - What must be standardized in the storage, access, and processing for better information integration?
  - What is the minimum in standards one needs for improved 'cooperation' and 'collaboration' of applications?
  - How can XML-based meta data help to improve to understand the semantics of data to perform challenging tasks such as information integration?

As cross fertilization is important, the major goal of the seminar is to bring representatives from the different communities (from research, from vendors, and from users) together for 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. As a major driving force we plan to use case studies coming for life scientists to discuss many of these issues from a user's (i.e. Life Science) perspective.

7.2 Conceptual and Technical Aspects of Electronic Learning

Seminar No. **03191** Date **04.05.–09.05.2003**
Organizers: C. Haythornthwaite, W. Stucky, G. Vossen

Electronic learning and in particular web-based learning is a topic that has been attracting various communities for many years already. Both in Europe and overseas we see it becoming a major industry and applied both in educational institutes (such as schools and universities) and in companies for the initial or continuous training of employees. E-learning initiatives are increasingly being implemented to support education and workforce enhancement. It is estimated that in the United States alone the e-learning 'industry' will grow from 2.3 billion dollars in 2001 to 23 billion in 2004; Western Europe has one of the highest per capita spending rates on continuing education and training. The online-learning share of training will grow from 20% to 40% against traditional classroom methods. Beyond that, universities are getting 'virtual' and are discovering e-learning as a central paradigm for life-long education and learning.

Besides all the hype that the topic has recently received, there are aspects in the field which can already be considered 'mature' (e.g., the decomposition of a learning platform into an

authoring system, a learning management system, and a run-time system), standardization is underway (e.g., LOM, SCORM), and there are various conceptual issues that are worth considering in appropriate depth: First, e-learning, although fundamentally based on the use of computers, originally emerged in communities other than computer science. Indeed, e-learning has its roots in such fields as performance improvement, education, psychology, and others for which the use of a computer has long been of secondary importance, and for which even in times of the Web a computer remains merely one tool among others. Second, it is well recognized that e-learning, when applied in a company, can yield more than just learning effects; it can also contribute to knowledge preservation and thus to the development of an organizational memory. Third, there is a technological side of the picture which is where computer scientists can mostly contribute: For example, databases are used for storing, retrieving, composing, and configuring learning content, XML is under discussion as an exchange format for such standards like IMS and SCORM (as schema or DTD specification language or in new markup languages such as LMML or EML), and the processes that are involved in an e-learning scenario are sometimes already modelled as workflows.

A successful implementation of an e-learning system relies heavily on building the appropriate infrastructure and selecting the proper tools and technologies that work for the learner and the organization. Thus, it is also worth observing recent products and services, delivery methods, standards, and systems used today. Developing courses for e-learning requires more than technology and creativity. Is the particular topic at hand suitable for remote learning? What are the right electronic elements for the topic and for a student? It is reasonable to explore how to design effective course content, follow up with useful assessment and tracking approaches, and to learn to foster ongoing learner and teacher support and match learning styles with various delivery methods. It is also important to match the tools with the goals of the e-learning environment, recognizing that goals and outcomes can vary, e.g., whether the system is designed to broadcast information only, provide an individual stand-alone learning environment, or create a learning community of collaborative peers.

Given these premises, the seminar brought together a small, but nicely composed collection of people from computer science (databases, knowledge representation, algorithms, multimedia etc.) with experience, ongoing projects, or proven interests in e-learning as well as web-based learning, and blended these people with participants with less technological focus such as library sciences. As a result, talks were given on a wide range of topics that clearly showed the span which the field is currently having. The talks were as follows (in chronological order):

1. Rudi Studer: E-Learning and the Semantic Web
2. Hartmut Schmeck: Scenarios for Computer-Assisted Instruction
3. Radha Gupta: Web Teaching of Computing for Business
4. Peter Westerkamp: xLx a Platform for Graduate-Level Exercises
5. Gottfried Vossen: Learning Objects, Processes, Workflows: A Technical View of E-Learning

6. Jörg Desel: Activities of the GI SIG on E-Learning
7. Thomas Ottmann: Presentation Recording
8. Ralf Klamma: Multimedia Semantics for Electronic Learning Environments
9. Carsten Ullrich: ActiveMath
10. Wasim Sadiq: Workflow-Driven E-Learning Services
11. Stephan Diehl: Collaborative Learning and Distributed Experimentation
12. Christopher Hoadley: Design-based Research and Distributed Cognition in Socio-Technical Systems for Learning
13. Wolfgang Nejdl: E-Learning 2003 ff
14. Jörg Desel: Pros (and Cons) of E-Learning Approaches in Universities
15. Caroline Haythornthwaite: Social Networks and Distance Learners
16. Daniel Sommer: Quality Information Systems for E-Learning Applications
17. Gerald Friedland, Lars Knipping: Electronic Chalk
18. Peter Westerkamp: E-Learning as a Web Service
19. Victor Pankratius: E-Learning Grids
20. Cornelia Seeberg: Courses based on Modules
21. Martin Stein: VISUM
22. Kirsten Keferstein: Process-based Learning Object Management
23. Rob Koper: Learning Networks and Standardization Issues
24. Bernd Krämer: Education a la Carte

Due to the variety of e-learning related aspects that could be presented and discussed, the week served its purpose of crossing borders very well. In spite of the small number of participants, lots could be learned from each other, and fruitful clarifications be obtained. It remains to be seen what benefits such an open forum can drive home in the years to come.

7.3 New Optimization Algorithms in Physics

Seminar No. **03381** Date **14.09.–19.09.2003**
Organizers: H. Rieger, A. Hartmann, K. Mehlhorn

Nearly three years earlier, in December 2001, the Dagstuhl Seminar "Algorithmic Techniques in Physics (II)" took place. Researchers from Computer Science, Mathematics and Physics came together to discuss about algorithmic problems occurring in physics and physical concepts that might be useful in computer science. Bringing together people from three different areas was an experiment that, as all participants agreed in the end, turned out to be a success and, more importantly, should be repeated in the future.

In the field of optimization, the interactions between computer scientists and physicist are strongly growing. This is due to an increasing number of optimization methods applied to problems from physics and, on the other hand, due to concepts and methods from statistical physics which are recently being applied to study optimization problems occurring in theoretical computer science. Still, many algorithms or problems are only known in one field. Hence, computer scientists as well as physicist could profit greatly by participating in this workshop, which aims to spread knowledge to other fields, respectively and to encourage new projects and cooperations.

In recent years, several very efficient exact optimization algorithms have been developed in the computer science community. Examples are maximum flow algorithms, minimum-cost flow techniques, matching methods, which all are graph theoretical approaches or sophisticated branch-and-cut methods, originating in the field of linear optimization. These algorithms have now been applied to problems from physics like for random magnetic materials (random-field systems, spin glasses), in surface physics (solid-on-solid models) and many other disordered systems. The system sizes which can be treated are now much larger than ten years before, allowing to obtain now more reliable and higher significant data.

Also several heuristic approaches have found applications in physics. An example are genetic algorithms, which mimic the optimization of species in an evolutionary process to find very good approximation of the global minima of complicated functions. Genetic algorithms have been recently applied to study systems ranging from the largest sizes, from galaxies to quantum systems. Recently, simple but nevertheless very efficient variants of genetic algorithms have been developed and were presented in the seminar.

The range of problems treatable with exact and heuristic optimization algorithms and the number of algorithms applicable to problems from physics is much larger than it has been realized so far. Hence, the physics community will profit a lot from learning more about recent algorithmic developments. On the other hand, computer scientists, who are looking for real-world applications of sophisticated algorithms, will benefit strongly by finding out about physical problems which can be solved using optimization methods.

In the field of inventing new algorithms, conversely computer scientists can profit from developments in the physics community. Several techniques, which originated in physical

problems or physical techniques, have been applied recently in different areas. The prototypical example is the simulated annealing method, which simulates the slow cooling of an experimental sample to find low energy states. This technique has been applied to many problems from other fields, like the traveling salesman problem or optimization of production schedules. Recently several enhancements of simulated annealing have been developed. Examples are the parallel tempering approach, where several systems are kept in parallel at different temperatures, and the multicanonical ensemble, where the temperature of the sample is allowed to fluctuate according a certain problem-adjusted recipe. Also other concepts from physics have led to the development of new algorithms. One example are renormalization-group based approaches, where the target function is optimized iteratively on different length scales. All these new methods will strongly enhance the efficiency of physics-based algorithms and enlarge greatly the range of applications.

A second emphasis of this workshop was the study of optimization problems from theoretical computer science using concepts and methods from statistical physics. Widely studied problems are the satisfiability problem (SAT), where one asks whether for a given boolean formula there exists an assignment of the variables satisfying all constraints, and the vertex-cover problem (VC), where one seeks for the distribution of marks in a graph such that each edge is adjacent to at least on mark. Both SAT and VC exhibit, like many other problems, phase transitions in a suitable parametrized ensemble of random instances. Thus, many methods invented in statistical physics to study phase transitions can be applied to problems from theoretical computer science, leading to results which could not be found before using traditional methods from mathematics. For example, SAT and the VC have been treated using the replica method, which was originally used to study the aforementioned spin glass problems analytically. Since there are more than 50000 NP-complete problems, many of them unknown to physicists, much work has still to be done in this field.

Interestingly, these phase transitions coincide very often with peaks of the running time or with changes of the typical-case complexity from polynomial to exponential. Hence, from studying these problems, one learns also a lot on the typical time complexity of algorithms. Recently, using the physical approaches, the complexity of simple complete SAT and VC algorithms could be analytically computed for the first time. In this area significant progress has been reported in various presentations in this seminar.

Finally, a part of this workshop was dedicated to bioinformatics. In this field, researchers from biology, computer science and physics cooperate in a most fruitful way. Algorithms provided by computer science and analytical methods and concepts from physics help to elucidate many problems from molecular biology. Examples are the study of protein structures and their dynamics or the prediction of secondary structures. Recently, using a mapping onto a physical system and by applying optimization algorithms, the rare-event statistics of sequence alignment could be studied, a method used to compare DNA and proteins stored in huge data bases.

All the examples given above show that by combining the efforts from computer science and physics substantial progress has been made in the recent years and more can be expected in the future. The participants as well as the organizers had the impression that this workshop contributed to this development and gave all participants many opportunities

for cross-community work and interdisciplinary collaborations.

The scientific highlights of the seminar were represented by the following key note speakers:

- Marc Mézard (LPTMS - Orsay): **Statistical Physics of the Satisfiability Problem: Survey Propagation**, where a new and very efficient algorithm for satisfiability problems was presented.
- Remi Monasson (CNRS, Paris): **Towards an Analysis of Average Case Properties of Backtrack Algorithms for Random Decision Problems**, where the performance of backtracking algorithms was analyzed with tools from statistical physics.
- Frauke Liers (Universität zu Köln): **Exact Ground States of Ising Spin Glasses**, where the recent remarkable progress in the exact computation of 3-dimensional spin glass ground states using branch-and-cut algorithms was reported.
- Martin Weigt (Universität Göttingen) **Solving Satisfiability Problems by Fluctuations: An Approximate Description of Stochastic Local Search Algorithms**, where the latter were analyzed with methods from statistical physics.
- Jean-Christian Angles d'Auriac (Grenoble) **Minimization of Sub-Modular Function: Application to the Potts Model**, where a polynomial algorithm for calculating the partition function of the infinite state random bond Potts model was presented.
- David Saad (Aston University, Birmingham) **A statistical mechanics based analysis of coded CDMA with regular LDPC codes**, where communication codes were analyzed again with tools known from statistical physics.

Chapter 8

Distributed Computation, Nets, VLSI, Architecture

8.1 Adaptivity in Parallel Scientific Computing

Seminar No. **03211** Date **18.05.–23.05.2003**
Organizers: I. Banicescu, K. Pingali, T. Rauber, G. Rünger

Although progress in parallel and distributed methodologies for scientific computing have been quite remarkable during the past years, this area of computer science remains still active, especially in topics concerning the relationship between performance and aspects such as: irregularity of applications and algorithms, adaptive characteristics of software and hardware, heterogeneity of hardware platforms, and flexibility of programming environments. Recent research activities include development of complex hardware architectures, including storage hierarchies or heterogeneous (parallel and distributed) computing platforms with large numbers of processors, as well as irregular applications that involve complex domain decomposition and hierarchical, adaptive and multi-level organization of computation and data structures. The corresponding irregular algorithms comprise applications with sparse, block-structured or adaptive data structures, as well as applications with irregular, runtime-dependent computation and control structures.

Over time, to improve scientific applications' performance on sequential machines, several techniques in hardware and algorithm design, such as storage hierarchies and hierarchical domain decomposition, have been introduced. However, the simulation of large irregular problems still requires the use of parallel and distributed environments. The irregular and dynamically changing runtime behavior makes an efficient parallel realization difficult, since the memory access patterns and the evolution of dynamic structures cannot be determined a priori, and therefore, cannot be planned statically. Consequently, an efficient parallel implementation of this class of problems necessitates the exploitation of flexible programming environments as well as techniques to improve scalability.

This seminar was a forum that brought together researchers working in different areas of parallel scientific computing and its applications, to solve scientific and industrially

oriented problems. It provided a fertile environment for the participants to meet and exchange ideas, as well as to foster future research collaborations.
Of particular interest was the exchange of experiences in interdisciplinary research projects. Topics covered by this seminar included:

- parallel numerical algorithms
- parallel implementation of irregular applications
- algorithms for memory hierarchies with enhanced locality of memory access
- libraries for supporting parallel scientific computing
- mixed task and data parallel executions on large parallel machines
- performance analysis evaluation and prediction
- compiler transformations for increasing the locality of memory references
- dynamic load balancing techniques
- partitioning and scheduling strategies
- heterogeneous computing (cluster and grid computing)
- combination of different programming models for heterogeneous parallel machines

During the seminar, a number of presentations lead to formulation of interesting open questions followed by discussions on optimal integration of adaptivity at various levels of technology in application, algorithms and system development. In the following paragraphs, we summarize a few concepts and ideas for new approaches, methodologies, and future directions that spawned from various talks and discussions.

In recent years, research in modeling and simulation is becoming increasingly important for a wide variety of scientific and engineering disciplines. It addresses the need for developing a safe, dependable and effective information environment, as well as the one for expanding of basic research in revolutionary fields which are of vital importance to our society. As a result, the research community is now faced with new challenges, such as the ones to incorporate additional physics, length scales and time scales, into models for adaptivity, higher fidelity and resolution, or to process variable amount of data from distributed datasets, which in turn place significant demands on software design and hardware implementation. Therefore, there is a need to explore and devise the design of a flexible, robust, and effective *vertical integration strategy*, for advanced development of scientific applications. This integration is expected to facilitate an effective fusion of advances in application algorithms, with the ones in programming environments, system software and hardware capabilities, for the purpose to enable terascale modeling and simulation.

The "Tinker-toy Parallel Programming", an interesting approach to building scientific applications via an aggregation of multiple, light-weight toolkits has been introduced during the seminar presentations. A solution to one possible drawback of such an approach, its

limited support for adaptive computations, has also been proposed using "Zoltan" – a tool that provides support for adaptive, parallel scientific computations, and easy development for dynamic and adaptive simulations.

The rapid development of an emerging technology in "cluster and grid computing" suggests a need for dynamic distribution of work and data that can be adapted to the runtime behaviour of the algorithm. A solution to that has been proposed, and its design, implementation, and evaluation have been presented using "task pools". In this approach, tasks are dynamically distributed to different processors (within nodes of a SMP, or among nodes on clusters of SMPs), and each task specifies computations to be performed and provides the appropriate data.

Some interesting presentations focussed on improving performance of irregular parallel applications via addressing sources of load imbalance at all levels of irregular behaviour (related to problem, algorithm or systemic factors).
A general purpose tool for dynamic loop scheduling to address the stochastic load variations from a range of sources has also been introduced.

A number of interesting discussions took place regarding recent advances in cluster and grid computing through a successful migration of parallel programs (via checkpointing and fault tolerance). In the future, the migration of parallel programs will allow parallel applications to "surf" the grid and adapt dynamically to its changeable environment.

A few interesting contributions presented challenges in BSP algorithm design, programming and software engineering to address adaptivity in scientific computations. Moreover, there were a few novel ideas and original concepts introduced on language support for irregular problems and adaptivity. The audience was delighted to discuss during the talk, as well as during our evening pleasant moments of get-together, some of the possible breakthroughs that could evolve from these ideas.

In conclusion, this seminar presentations and discussions addressed many complex issues including application requirements for adaptivity in space and time, as well as requirements for improving the capacity to effectively use resources in heterogeneous environments. The seminar topics span and integrate the work of many research areas: from irregular scientific applications, to adaptive algorithms, programming models and tools, problem solving environments for cluster and grid computing, and others.
We believe that these contributions, in addition to talks and many interesting discussions, will inspire the participants to continue their research efforts towards an integrated view of adaptivity, allowing them in this way to make significant contributions to the advancement of science.

8.2 Algorithmic Game Theory and the Internet

Seminar No. **03291** Date **13.07.–18.07.2003**
Organizers: M. Karpinski, C. Papadimitriou, V. Vazirani

The seminar was devoted to the most important recent developments in the area of the Algorithmic Game Theory connected to the problems arising from, and motivated by, the

Internet and other *decentralized* computer networks. The most defining characteristic of the Internet is that it was not designed by a single central entity, but emerged from the complex interaction of many economic agents, such as network operators, service providers, designers, users, etc., in varying degrees of collaboration and competition. The major questions that arise in that context are in analysis of its performance and in evaluation of its long term equilibria. They include all sorts of completely new questions that lie on the interface of the fields of networks, algorithms and game theory.

The focus of the workshop was on the following specific topics:

- design of efficient algorithms for game theoretic problems connected to the Internet,
- inherent complexity of game theoretic problems,
- resource allocation and stability,
- Nash equilibria,
- market equilibria,
- mechanism design,
- economic aspects of the Internet,
- combinatorial auctions and
- cost allocations, network design.

Some new broadly applicable techniques have emerged recently in the above areas and the workshop has addressed those developments and new fundamental insights. The workshop has also addressed and formulated important open problems of the area and identified most challenging research directions for the future.

The 47 participants of the workshop came from various research areas connected to the main topic of the workshop. The 31 lectures delivered at the workshop covered wide body of recent research in the area. In addition, a special evening session was devoted to presentation of open problems.

8.3 Dynamically Reconfigurable Architectures

Seminar No. **03301** Date **20.07.–25.07.2003**
Organizers: P. Athanas, J. Becker, G. Brebner, H. ElGindy

The Dagstuhl seminar on "Dynamically Reconfigurable Architectures" has been a very successful meeting of people from different research areas – algorithms, hardware architectures and circuits as well as optical communication.

The seminar showed that technological advances have opened up new ways of implementing complex systems in a way that blurs the barriers between hardware and software

components development, and that existing design tools do not seem to be adequate for the necessary new design styles. Furthermore, new advances in optical communication lead to feasible implementations of interconnection structures which are getting not only theoretical value nowadays.

In recent years a rapidly growing interest in using reconfigurable computing architectures for realizing and developing application-specific computer systems has been observed. The advances in reconfigurable technologies, in algorithm implementation methods, and in automatic mapping methods of algorithms into hardware and processor spaces form together a new computing paradigm of computing and programming, e.g. "Computing in Space AND in Time". This requires different and new approaches in engineering for developing reconfigurable systems and implementing complex algorithms, including theory, architecture structures, algorithms, design systems and industrial applications that demonstrate the benefits of this promising way of computing.

The fast pace of development is leaving industry not enough time to develop the necessary theoretical foundation that underpins CAD tools, OS, designs, architectures and circuit technologies. Traditional hardware and software design processes and the tools to support them are not adequate for the design of run time reconfigurable systems. Therefore, the plan for this seminar is to focus on the issues relevant to the development of support for the meanwhile also in industry attractive reconfigurable technologies. A special focus will be given to dynamically run-time reconfigurable (RTR) solutions, since here system adaptivity and advantages of this technology are highly visible.

The seminar will cover: architecture structures, circuit technologies, system architecture, tools for RTR, general/special purpose system, and of course, existing and new application domains, where (dynamically) reconfigurable computing is more effective than traditional and parallel/distributed architectures. This includes also an appropriate set of models for reconfigurable systems which open ways for application designers and industry to develop efficiently their systems using appropriate high level languages. Especially the risk minimizing factors (time-to-market!) and adaptivity features (multipurpose/ multistandard possibilities!) are important arguments for industrial companies now, e.g. in (mobile) communication technologies, automotive area, etc. to integrate this flexible technology into their product strategy. Here Configurable Systems-on-Chip (CSoCs) solutions for embedded systems are giving valuable perspectives. We also think that the challenges posed by integrating optical technology with RTR should remain a considerable aspect of this seminar.

8.4 Internet Economics

Seminar No. **03321** Date **03.08.–07.08.2003**
Organizers: B. Stiller, L. McKnight, M. Karsten, P. Reichl

1 Introduction

The Dagstuhl Seminar on "Internet Economics" brought together two groups of international experts on networking and economists for the Internet. While the underlying em-

phasis on today's technology for an end-to-end provisioning of Quality-of-Service (QoS) – covering the Internet as a network as well as the end-system – determines the networking aspects of Internet Economics, the business policy management, the economics of service differentiation, and incentive structures required for a charging support of transport and content defines the key economic aspects. Both areas target the joint discussion and identification of solutions, investigations of their feasibility, and a consolidation of technical and economic mechanisms to enable a fast, guaranteed, and efficient provisioning of differentiated services in the Internet.

2 Public Outreach

Internet Economics outline a key aspect of a commercialized Internet, which address a.o. the pricing problem for Internet services and various management as well as resource allocation problems under economic perspectives. The combination of technical mechanisms, Internet protocols, and economic models determines the best possible methodological approach for optimizing the commercial operation of Internet services in heterogeneously wired and wireless networking technology environments. This Dagstuhl Seminar on "Internet Economics" emphasized the economic modeling of technology problems in the Internet and considered the network, its technical mechanisms, and some areas of the Internet's application domains. While the combination of technical protocol and distributed system aspects cover security, efficiency, and load control, the economic view points included modeling of content pricing and mobile ad-hoc network pricing.

3 Scientific Highlights

The seminar was organized in four sessions, addressing the following topics:

- Pricing,
- ISPs and Internet Economics,
- Architecture and Peer-to-peer, and
- Auditing and Load Control.

3.1 Pricing

The first topic covered the problem of selling e-con-tent, e.g., video by auction mechanisms. While the work investigated suitable mechanisms and effects of this pricing mechanism, the applicability in a wide range of e-con-tent remained debated. In addition, the modeling of correct incentives for a collaboration of users and devices in mobile ad-hoc networks was presented. An underlying model and definitions for ad-hoc under economic perspectives were given.

Discussions covered technical details, which indicated the problem of which parameters to include into a viable and realistic model. Therefore, two working groups have been set up

to identify and determine the key problems and tasks in the pricing domain with respect to auctions as well as wire-less ad-hoc local area network. Key results show that auctions require further investigations in terms of user acceptance and technical practicability in distributed systems. In addition, the ad-hoc group determined an initial level of low-level parameters to be incorporated into an applicable model for incentive investigations.

3.2 ISPs and Internet Economics

Internet Service Providers (ISP) operate successful only in an interconnected manner, therefore, ISPs with those mandatory interconnections see costs due to up to1000 peering partners to be managed. This problem has been investigated by mathematical optimization mechanisms and an architecture has been defined. The broader area of technical issues in networks, a seamless operation for the user, the economic success of such a network including its applications and services offered, has been combined with law enforcement perspectives in a new project and study program on information economic, computer science, law, and economics with interdisciplinary issues.

Finally, a proposal termed Contract and Balancing Process (CBP) was presented, which addresses the problem on how an owner of a communication network shall sell bandwidth to users; e.g., for business with Entertainment-on-demand. Having explicit congestion notifications and their marks in an Internet enables the operator to provide the right incentives and prices to charge his customers a fair price corresponding to their initial statements.

3.3 Architecture and Peer-to-peer

Suitable architectures for commercially applicable networks require service components in an all-IP networking environment in support of charging. However, problems arise from technical faults within the network. Therefore, the key issue is: can these types of services be charged? Depending on the accounting infrastructure and the details being accounted for various different charges may be applied. In case of peer-to-peer (P2P) systems this problem increases, since a group of people is working together without any controlling entity with any type of permanent privileges. Even more, some areas may show a conflict of interest. P2P systems show currently market failures, which are based on the fact that current P2P applications make a contribution to a public good, rather than a marketable good.

The approach presented introduces market management mechanism in P2P systems and currently develops a prototype. Finally, the technical networking details in the network effect the service quality extremely, which in turn shall be charged. Therefore, scheduling matters for non-cooperative multi-class QoS provisioning and business models for assured services are essential. The three types of functional, performance, and organizational challenges have been presented and proposals for end-to-end QoS in legacy operating systems in the local area access have been shown. In consequence, two working groups have been established in the seminar to discuss trust as well as intelligent end systems and

externalities. In case of charging there was a consensus that any type of trust is required, while the problem on conflicting information has to be considered.

Though, either established through means of an infrastructure, e.g., a full set of PKIs and security mechanisms, or by means of reputation mechanisms remains an open debate. With respect to intelligence of endsystems and their mechanisms, P2P economics need to cover nonexclusive goods, e.g., content with digital copies at no cost. Based on the categorization of private and public goods as well as natural monopoly or common resource goods a P2P network may work in a rivalries or nonrivalrous fashion, which may be excludable or nonexcludable goods' offers.

3.4 Auditing and Load Control

Once data of service usage has been accounted for in a subsequent step the validity of these data has to be verified, which requires auditing mechanisms in place. While the Service Level Agreement (SLA) auditing has been focused on in this work, related mechanisms have been investigated in terms if security auditing, e.g., denial-of-service attack or intrusion detection. Though, the key problem in SLA auditing is the violation detection. The approach presented offers a framework and initial mechanisms to specify and describe those actions to be undertaken, many of them automatically, to verify the compliance degree of a service delivered with its original specification.

Binary packet marking has been suggested earlier as an economic signal to enforce cooperation from end systems in times of overloaded network resources. Load control gateways at edge gateways allow building a network system that uses the load signal embedded in a packet stream for connection admission control. The design and implementation of a prototype system has been presented. Thorough performance investigations in various scenarios and with different mechanisms showed that edge based load control can be performed effectively and can efficiently provide reliable service guarantees.

4 Perspectives

In the mid-term range Internet Economics will be effected by law and policy guidelines, which will vary depending on the region of the world. This regionalized view point simplifies the understanding of the problem areas, though, defines an obstacle for worldwide and open markets operating under the same set of rules. However, the need for interdisciplinary research work, especially the effects of incentives and legal aspects for service delivery, proof, and provisioning will determine the key problems to look into soon.

8.5 Algorithmic Aspects of Large and Complex Networks

Seminar No. **03361** Date **31.08.–05.09.2003**
Organizers: M. Adler, F. Meyer auf der Heide, D. Wagner

One cornerstone of our modern society is the use of different kinds of networks. Our cities are connected by a network of streets and railways, telecommunication networks including their wireless, mobile components, the internet, and the World Wide Web build the most important infrastructure for communication and information worldwide. Designing and managing such networks pose challenging algorithmic problems.

The second Dagstuhl Seminar on "Algorithmic Aspects of Large and Complex Networks" brought together 45 researchers (32 Germany, 4 USA, 3 Switzerland, 2 Italy, 1, Slowenia, 1, Poland, 1 Israel, 1 Greece) to discuss recent advances on a huge variety of network problems as described above. Most of the German participants were members of the corresponding DFG research cluster. The purpose of the workshop was to give the opportunity to exchange ideas between researchers working on different areas of complex networks. Interesting talks, fruitful discussions between researchers on different fields and with different background, and the wonderful working and living environment of Schloss Dagstuhl contributed to the success of the workshop. Below we give some examples for the topics considered at the workshop.

Traffic networks.
We discussed the modelling and computation of time tables for large traffic networks. This included the computation of time tables for trains and airplanes as well as models for individual traffic.

Time dependent networks.
Related to the computation of time tables is the area of time dependent networks. Here we discussed network algorithms that solve variants of standard network problems (e.g., shortest paths and network flow) on networks that change over time. Basic network services. One of the topics of the workshop was the question how to provide efficient basic services (e.g., routing) for large computer networks.

Mobile and wireless networks.
The design of algorithms for mobile ad hoc networks and sensor networks is one of the challenges at the beginning of the new century. Dangerous jobs (e.g., exploring contaminated terrain) may in future be performed by robots instead of humans. These robots will be connected by wireless ad hoc networks. We discussed models and algorithms for these kinds of networks.

Chapter 9

Modelling, Simulation, Scheduling

9.1 Challenges in High Performance Simulations for Science and Engineering

Seminar No. **03111** Date **09.03.–14.03.2003**
Organizers: U. Rüde, F. Hoßfeld, P. Langtangen, Ch. Johnson

Enormous growth in computing power and advances in parallel algorithms are enabling the realistic simulation of complex systems of the physical world. Computer simulations – that is high accuracy virtual models of the real world – have begun to replace expensive or dangerous experiments. Computer simulations even allow to experiment with systems and processes which are not open to real experiments (like cosmological, economical, or sociological systems).

Computer simulation is quickly becoming a universal methodology. Examples include weather prediction, climate modeling, astrophysics, turbulence, combustion, biomedical technology, financial engineering, material sciences, environmental modeling, and waste management. Other strategic fields are protein folding, macromolecule and drug design, quantum chemistry, reactive fluid flow, logistic systems, plasma and fusion physics, aerodynamics, superconductivity, string-theoretical problems, and quantumchromodynamics.

The seminar has focussed on simulation as a tool for computational science and engineering applications. To be a useful tool, such simulations must be based on accurate mathematical descriptions of the processes and thus they involve mathematical formulations, like partial differential equations or integral equations. Scientific simulations require the numerical solution of such problems and thus will use enormous resources in both processing power and storage. Even more computing power is needed when the simulation is used only as a component within a more complex task. This happens, e.g. when an engineering design is automatically optimized. In this case a simulation run must be performed within each iteration of the optimization algorithm.

Despite rapid progress over the past three decades, the practical use of high performance simulation and its applications will be facing several severe obstacles within the next decade. Desirable, realistic models are still too compute intensive for current processing

technology. While the fastest computers today may be able to handle simulations with at most 10^9 – 10^{11} degrees of freedom and perform in the range of a few Teraflop (10^{12} operations) per second, the next generation of models will require up to three orders of magnitude more computing power. Current roadmaps predict the availability of petaflops systems, capable of 10^{15} operations per second by the end of the current decade. Such systems are necessarily massively parallel.

The seminar talks have covered topics including

- scalable parallel simulation algorithms
- numerical methods
- the architecture of scalable massively parallel systems
- multiple levels of parallelism, from instruction or task level to message passing in a networked cluster
- devising algorithms and implementation techniques capable to tolerate latency and bandwidth restrictions of future petaflop systems
- software engineering techniques for computational science and engineering applications
- problem solving environments
- handling the complexity of multi-physics models
- validation and verification of large scale simulations
- alternatives to silicon-based computing

The seminar has brought together researchers from across the disciplines who are involved in all aspects of high performance simulation and dealing with the challenges of future petaflops simulations. The discussion across the disciplines, including the hard- and software architecture of the next generation of supercomputers, but with an emphasis on the design of new algorithms, tools, and programming techniques has been especially fruitful. Even more interdisciplinary collaboration will be necessary for efficiently exploiting such systems and managing the enormous complexity of current and future scientific simulation problems.

The results of the seminar will be published in book form.

Chapter 10

Data Bases

10.1 Perspectives Workshop: "Multimedia Retrieval"

Seminar No. **03112** Date **10.03.–13.03.2003**
Organizers: M. Clausen, R. Klein, I. Witten

Content based retrieval of multimedia documents containing text, image, audio, or video is of fundamental importance for a number of applications. This seminar brought together 22 participants from different areas like computer graphics, database and information systems, applied mathematics, audio retrieval, and computational geometry.

Altogether 17 talks were given, on a variety of topics. Yet, it was possible to identify some core problems that kept occurring in different contexts.

1. How to define, and measure, the similarity between objects; how to implement efficient retrieval algorithms for finding the objects most similar to a given query.
2. How to define semantics of multimedia documents, and how to enable semantic-based retrieval.
3. How to decompose documents into segments, and how to perform segment-based retrieval.

These core problems should receive high attention in future research.

Key Words: Multimedia retrieval, shape analysis, feature extraction, metric design, invariants, segmentation, mesh generation, knowledge representation, wavelets, geometric matching.

10.2 Inconsistency Tolerance

Seminar No. **03241** Date **09.06.–13.06.2003**
Organizers: L. Bertossi, P. Besnard, A. Hunter, T. Schaub

Introduction

Database, Knowledgebase and Software systems, or their logical specifications, may become inconsistent in the sense of containing contradictory pieces of information. Since these types of technology are at some level based on classical logic, there is the major problem that in classical logic, any formula is implied by a contradiction. This therefore raises the need to circumvent this fundamental property of classical logic whilst supporting as much as possible of classical logic for these technologies. To address this, several new logics, with new formalisms, semantics and/or deductive systems, that can accommodate classical inconsistencies without becoming trivial, have been proposed. These logics are starting to be used in databases, knowledgebases and software specifications.

In addition, we need strategies for analysing inconsistent information. This need has in part driven the approach of argumentation systems which compare pros and cons for potential conclusions from conflicting information. Also important are strategies for isolating inconsistency and for taking appropriate actions, including resolution actions. This calls for uncertainty reasoning and meta-level reasoning. Furthermore, the cognitive activities involved in reasoning with inconsistent information need to be directly related to the kind of inconsistency. So, in general, we see the need for inconsistency tolerance giving rise to a range of technologies for inconsistency management. We are now at an exciting stage in this direction. Rich foundations are being established, and a number of interesting and complementary application areas are being explored in decision-support, multi-agent systems, database systems, and software engineering.

The seminar brought together specialists from the communities of knowledge representation, databases, software specification, and mathematical logic, with the aim of exchanging research results, ideas and experiences around logic based approaches to inconsistency tolerance in computational systems.

The Seminar and its Projection

The seminar concentrated on inconsistency handling in basically five areas: non-classical logic, knowledge representation and non-monotonic reasoning, logic programming, databases, and software specification.

Whenever some sort of formal logic is used to specify a system, to write down a theory, to represent data or knowledge, etc., inconsistencies may naturally arise. The problem consists then in finding the way of reasoning in the presence of such inconsistencies without trivializing the whole process; or in being able to solve the inconsistencies, e.g. passing to a new, unifying theory, representation or specification; or in being able to isolate the inconsistencies, possibly detecting and using the consistent part of the database, theory, specification, etc.

Problems around inconsistency handling, their conceptualization, solutions, techniques were presented and discussed from different perspectives. Most important in this direction was the heterogeneity of the audience and presenters, who benefited from the different kind of expertise and points of view of other participants. Illuminating discussion were carried out, and research interaction naturally started.

The area of inconsistency handling has received considerable interest from the logical and computer science communities in the last, say three years. This seminar appeared in the right moment. It attracted many participants (and was difficult to accommodate all those who wanted to present), and there was clear interest among them in organizing a second version of it in the near future.

The organizers have already contacted Springer Verlag to publish a book as a natural follow-up of the seminar. The publishing house accepted this proposal and several of the participants (and a few other experts on the field) have already been invited to contribute with a chapter that should both survey his/her area of expertise in inconsistency handling and present some state of the art research. Around sixteen chapters are planned, several of them will be written by more than one author, since the editors have tried to encourage synergy and collaboration in this community. The invitation has been positively received by all the potential authors. The chapters have to be submitted in December 2003. After that they will go through an anonymous review process, that will determine which of them will be accepted, possibly subject to changes. The editors will be L. Bertossi, A. Hunter, and T. Schaub.

10.3 Data Quality on the Web

Seminar No. **03362** Date **31.08.–05.09.2003**
Organizers: M. Gertz, T.M. Özsu, G. Saake, K.-U. Sattler

Although techniques for managing, querying, and integrating data on the Web have significantly matured over the last few years, well-founded and applicable approaches to determine or even to guarantee a certain degree of quality of the data are still missing. Reasons for this include in particular the lack of common, agreed-upon models of quality measurements and the difficulty of handling quality information during data integration and query processing. The problem of data quality arises in many scenarios, e.g., during the integration of business or scientific data, in Web mining, data dissemination, and in particular in querying the Web using search and meta-search engines. Furthermore, it affects various kinds of data, such as structured and semistructured data, text documents as well as streaming data. Information about data quality is becoming more and more important since it provides some kind of yardstick describing the value and reliability of (possibly heterogeneous) forms of distributed or integrated data.

The aim of this seminar was to foster collaboration among researchers from different areas working on problems related to data quality. This included but was not limited to data integration, information retrieval (particularly search engines), scientific data warehousing and applications domains from the computational sciences and bioinformatics. In all these areas, data quality plays a crucial role and therefore different specific solutions have been developed. Sharing and exchanging this knowledge could result in significant synergy effects.

The seminar focused on the following major issues:

- Criteria and measurements for quality of Web data,
- Representation and exchange of quality information as metadata,
- Usage and maintenance of data quality in Web querying and data integration.

The intention was to clarify terminologies and models, analyze the state of the art in the different areas, discuss problems, approaches and applications of quality-aware Web data management and to identify future trends and research directions in the above mentioned areas.

For this purpose, the seminar was organized in four working groups

- Metadata & Modeling,
- Information Quality Assessment and Measurement,
- Do you Trust in Data Quality?,
- and Data Integration,

where participants discussed the special issues and presented their results to the other group members afterwards.

Chapter 11

Other Work

11.1 e-Accessibility: new Devices, new Technologies and new Challenges in the Information Society

Seminar No. **03481** Date **23.11.–27.11.2003**
Organizers: M. Jarke, A. Kobsa, K. Miesenberger, C.A. Velasco

As business and society become more and more dependent on information and communication technologies as well as embedded software systems, the impact of the Digital Divide caused by differences in accessibility to the new technologies is growing in terms of widening differences in educational chances, job market situation, interaction with public administration and government, and – last not least – life quality as a consumer, tourist, patient, etc. To counter this adverse trends, accessibility initiatives worldwide aim at making ICT address the special needs of a much broader group of the population than the traditional technologies which were addressed to an 'average' user. Several important regulations and guidelines in the accessibility area underline this trend. Especially for research, it is, however, also important to look into new challenges that will face accessibility in the future, such as the following:

- We are witnessing a rapid deployment of new devices and technologies that implement the paradigm of ambient Intelligence and ubiquitous computing to allow access to information in different environments. These devices are increasing the risk of Digital Divide for people with special needs (disabled and elderly), as neither Design-for-All methodologies, nor interfaces with assistive devices and software are implemented.
- The new devices challenge accessibility because of their smaller displays, their lack of keyboard – or a small embedded keyboard – and their size reduction. People with motor, visual or hearing impairments are defied by these characteristics and realize that traditional assistive devices are not tackling their needs as in the standard desktop environment. New interaction paradigms and new interfaces must be devised to facilitate access to the new gadgets in "ambient intelligence" scenarios, including biofeedback sensors as input systems.

- The Internet is no longer a set of static HTML pages. Multimedia elements, new XML-based languages and complex Content Management Systems that allow publishing to different environments require a different approach to accessibility for authors and end-users alike.

However, accessibility should not just be seen as enabling use despite ICT innovation. Equally important is the potential for increased participation in society through ICT. The Semantic Web, Web Services, JXTA, RDF, CC/PP and location-sensitive awareness services will help design smart proxy-tools able to react to the needs of the user, the device she is using, and her environment, providing the information requested in an appropriate way, including accessibility considerations.

The year 2003, highlighted as the International Year of People with Special Needs, appears thus as a good opportunity to take stock of the accessibility solutions achieved so far, and to identify the interdisciplinary challenges for accessibility research in the next years. The Dagstuhl seminar will bring together leading researchers from universities, research institutes, and industry to address in particular issues such as: analysis of needs and opportunities for accessible innovations; challenges and solutions for mass customization which allows addressing special needs much more deeply than today at radically reduced costs; research methods, validation, and cooperation between research, government, and industry. In addition to plenary presentations by selected participants, the seminar will comprise a number of working groups with the aim of summarizing major challenges with the aim of informing both the research, user and industry communities, and the public.