DAGSTUHL
REPORTS

Volume 3, Issue 9, September 2013

 ISSN 2192-5283

ISSN 2192-5283

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

Publication date
January, 2014

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at `http://dnb.d-nb.de`.

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

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

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

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

Digital Object Identifier: 10.4230/DagRep.3.9.i www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 13361

Crowdsourcing: From Theory to Practice and Long-Term Perspectives

Edited by

Tobias Hoßfeld[1], Phuoc Tran-Gia[1], and Maja Vukovic[2]

1 **Universität Würzburg, DE,** `[hossfeld,trangia]@informatik.uni-wuerzburg.de`

2 **IBM TJ Watson Research Center, US,** `maja@us.ibm.com`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13361 "Crowdsourcing: From Theory to Practice and Long-Term Perspectives". Crowdsourcing is a newly emerging service platform and business model in the Internet. In contrast to outsourcing, where a job is performed by a designated worker or employee, crowdsourcing means to outsource a job to a large, anonymous crowd of workers, the so-called human cloud, in the form of an open call. Current research in crowdsourcing addresses the following issues: crowdsourcing as a novel methodology for user-centered research; development of new services and applications based on human sensing, computation, and problem solving; engineering of improved crowdsourcing platforms including quality control mechanisms; incentive design and gamification of work; usage of crowdsourcing for professional business; theoretical frameworks for evaluation. The topic on crowdsourcing may have a huge impact on the Internet and its technical infrastructure, on society, and the future of work. In short, crowdsourcing will be a guiding paradigm and form the evolution of work in the next years. Therefore, this seminar helps coordinating research efforts in the different communities. In five presentation and discussion sessions, the diverse aspects of crowdsourcing were elaborated. The topics of the sessions covered (S1) crowdsourcing in general, (S2) industry use cases, (S3) crowdsourcing design and engineering, (S4) programming and implementing crowdsourcing, (S5) applications of crowdsourcing.

The major interests of the seminar participants were then focused in four different working groups on (W1) long-term perspectives & impact on economics in five years, (W2) theory – taxonomy and dimensions of crowdsourcing, (W3) industry use cases, (W4) crowdsourcing mechanisms and design. In parallel to this seminar, a topically related seminar on "Cloud-based Software Crowdsouring", organized by Michael N. Huhns, Wei Li, Martin Schader and Wei-Tek Tsal,(Dagstuhl Seminar 13362) took place. Therefore, a joint late night session was organized to discuss crowdsourcing with respect to ethics and its relation to social computation.

Seminar 01.–04. September, 2013 – www.dagstuhl.de/13361

1998 ACM Subject Classification H.1.2 User/Machine Systems, H.4 Information Systems Applications, H.5 Information Interfaces and Presentation, H.5.1 Multimedia Information Systems, H.5.3 Group and Organization Interfaces, J.0 General Computer Applications, K.4 Computers and Society, K.4.3 Organizational Impacts

Keywords and phrases Crowdsourcing, Human Computation, Human Cloud, Applications, Industry Use Cases, Crowdsourcing Design, Mechanisms, Engineering, Practical Experience

Digital Object Identifier 10.4230/DagRep.3.9.1

Crowdsourcing: From Theory to Practice and Long-Term Perspectives, *Dagstuhl Reports*, Vol. 3, Issue 9, pp. 1–33

Editors: Tobias Hoßfeld, Phuoc Tran-Gia, and Maja Vukovic

Dagstuhl Reports

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

1 Executive Summary

Tobias Hoßfeld
Phuoc Tran-Gia
Maja Vukovic

Over the past several years crowdsourcing has emerged as a new research theme, but also as a new service platform and Internet for harnessing the skills of the large, network-connected crowd on-line. Whilst the research community has not just yet recognized crowdsourcing as an entirely new discipline, many research challenges remain open and need to be addressed to ensure its successful applications in academia, industry and public sectors. Crowdsourcing research intersects many existing domains and brings to the surface new challenges, such as crowdsourcing as a novel methodology for user-centered research; development of new services and applications based on human sensing, computation and problem solving; engineering of improved crowdsourcing platforms including quality control mechanisms; incentive design and gamification of work; usage of crowdsourcing for professional business; theoretical frameworks for evaluation. Crowdsourcing, as a new means of engaging human capital online is increasingly having an impact on the Internet and its technical infrastructure, on society, and the future of work.

With crowdsourcing gaining momentum and becoming mainstream, the objective of this Dagstuhl seminar was to lead coordination of research efforts in the different communities, especially in US currently leading the crowdsourcing market and in Europe. The seminar engaged experts from the different research fields (e.g. sociology to image processing) as well as experts from industry with a practical background on the deployment, operation or usage of crowdsourcing platforms. From industry, real-world problem statements, requirements and challenges, position statements, innovative use cases, and practical experiences are tackled and discussed. The collection and analysis of practical experiences of the different crowdsourcing stakeholders were key outcomes of the Dagstuhl Seminar. The seminar was structured so that the participants use existing use cases, as a driver in the discussion to envisions future perspectives of this domain. To move forward, we identified the need for a common terminology, classification and taxonomy of crowdsourcing systems, as well as evaluation frameworks; and have already proposed a blueprint of the same. The impact of crowdsourcing from different perspectives has been discussed, by participants' viewpoints stemming from societal, business, economic, legal and infrastructure perspectives.

From platform provider side, Nhatvi Nguyen (Sec. 3.11) showed the actual challenges in operating a crowdsourcing platform. As industry use case, the example of enterprise crowdsourcing was presented by Maja Vukovic (Sec. 3.14), where the rapid generation of a snapshot of the state of IT systems and operation is conducted by means of crowdsourcing. This allows for massive cost savings within the company by uncovering knowledge critical to IT services delivery. Crowdsensing is another industry use case presented in the seminar by Florian Zeiger (Sec. 3.15). Environmental sensing in the area of safety and security was discussed from industry point of view along with the challenges and open questions, e.g. user privacy, data quality and integrity, efficient and reliable data collection, as well as architectural decisions and flexible support of various business models. A concrete application for crowdsensing is radiation sensing as shown by Shinichi Konomi (Sec. 3.7).

Beyond this, there were also discussions on multimedia related use cases. Crowdsourcing can be efficiently used for describing and interpreting multimedia on the Internet and allows to better address other aspects of multimedia with meaning for human beings. Martha

Larson (Sec. 3.10) provided examples of these aspects like the emotional impact of multimedia content, and judgments concerning which multimedia is best suited for a given purpose. Klaus Diepold (Sec. 3.6) applied crowdsourcing to move subjective video quality tests from the lab into the crowd. The resulting ratings are used to train mathematical model for predicting subjective quality of video sequences. Multivariate data analysis tools are recommended to incorporate contextual information to further validate the mathematical model. Vassilis Kostakos (Sec. 3.8) showed that the data quality of appropriate subjective tests may be increased by using public displays and touch screens in cities compared to online surveys. While gamification pops up as buzzword aiming among others at increased data quality, Markus Krause (Sec. 3.9) mentioned that the player should be put first i.e. the desires of player are paramount. In particular, task and game ideas need to be able to be linked, while fun has to be the main motivator for the game.

General approaches to improve crowdsourcing and the resulting data quality were a topic of interest by several participants. Gianluca Demartini (Sec. 3.5) proposes to model workers in the crowd as basis for quality assurance mechanisms. Alessandro Bozzon (Sec. 3.2) demanded for better conceptual abstractions for crowd tasks and processes design and (automatic) generation; better understanding of crowds properties such as (soft and hard) skills, reliability, availability, capacity, precision; and better tools for measuring and driving worker engagement. Cristina Cabanillas (Sec. 3.3) considered the human resource management aspects starting from workflows to crowdsourcing. Abraham Bernstein (Sec. 3.1) discussed human computers as part of computational processes, however, with their own strengths and issues. The three traits on human computation, that are motivational diversity, cognitive diversity, and error diversity, are embraced as strengths instead of weaknesses. While the main focus of the seminar was on technical challenges, the potential impact and long-term perspectives were discussed from an interdisciplinary point of view too, given the social and human aspects of crowdsourcing. Those issues were also raised by Phuoc Tran-Gia (Sec. 3.13) and Joseph G. Davis (Sec. 3.4).

Overall there were 22 participants from 9 countries and 16 institutions. The seminar was held over 2.5 days, and included presentations by researcher and specific hands-on discussion sessions to identify challenges, evaluate viewpoints and develop a research agenda for crowdsourcing. While the abstracts of the talks can be found in Section 3, a summary of the discussions arising from those impulse talks is given in Section 7. Additional abstracts and research statements without any presentation in the plenary are also included in the report in Section 4. The different aspects of crowdsourcing were discussed in more detail in four different working groups formed during the seminar: (W1) long-term perspectives & impact on economics in five years, (W2) theory: taxonomy and dimensions of crowdsourcing, (W3) industry use cases, (W4) crowdsourcing mechanisms and design. The summary of those working groups can be found in Section 5.

Please note that a related seminar on "Cloud-based Software Crowdsouring" (Dagstuhl Seminar 13362), organized by Michael N. Huhns, Wei Li, Martin Schader and Wei-Tek Tsai, took place in parallel to this seminar. We held a joint social event and a session on discussing research challenges and planned publications. In this late night session, on one hand ethical issues in the area of crowdsourcing were raised in a stimulus talk by Martha Larson (TU Delft). On the other hand, Munindar P. Singh (North Carolina State University) intended to provoke with his talk on the critique of current research in the area of social computing and crowdsourcing. A summary can also be found in Section 7.

A comprehensive list of open problems and challenges in the area of crowdsourcing as observed and stated by the participants is another key outcome of the seminar which is provided in Section 6.

2 Table of Contents

3 Overview of Talks

In a madness session, all participants introduced themselves, stated their areas of interest, their expectations from the seminar, and their view on crowdsourcing science within at most five minutes. Before the seminar, the participants were asked to address the following questions. *What is your specific interest in crowdsourcing? What do you expect from this seminar? What are your objectives? Which challenges and specific questions would you like to address or see addressed? What do you have to offer? What do you expect from industry/academia?* Those questions were then addressed in more detail in the presentation session. The participants were asked whether they want to give a presentation and volunteers were given a 30 minutes slot for presentation including time for intensive discussions on their research. Based on the interests and the abstracts, the program of the seminar and corresponding presentation sessions were formed. A tag cloud of the most common words in the abstracts is visualized in Figure 1, while the abstracts itself can be found below.

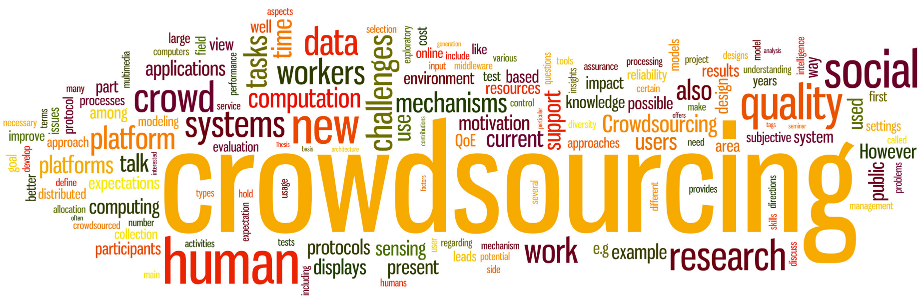


Figure 1 Tag cloud of the most common words in the abstracts.

3.1 Programming the Global Brain with CrowdLang and CrowdOS – Challenges and Ideas

Abraham Bernstein (Universität Zürich, CH)

Before the Internet most collaborators had to be sufficiently close by to work together towards a certain goal. Now, the cost of collaborating with anybody anywhere on the world has been reduced to almost zero. As a result large-scale collaboration between humans and computers has become technically feasible. In these collaborative setups humans can carry the part of the weight of processing. Hence, people and computers become a kind of “global brain” of distributed interleaved human-machine computation (often called collective intelligence, social computing, or various other terms). Human computers as part of computational processes, however, come with their own strengths and issues.

In this talks we take the underlying ideas of Bernstein et al. (2012) regarding three traits on human computation—motivational diversity, cognitive diversity, and error diversity—and discuss the challenges and possible solution approaches in order to embrace these traits as strengths instead of weaknesses.

3.2 Crowdsourcing Engineering

Alessandro Bozzon (TU Delft, NL)

While we are just starting to understand the societal and economical impact of crowdsourcing, available models and technologies for crowd organization and control are still in their infancy. The talk is intended to stimulate discussions about the need for 1) better conceptual abstractions for crowd tasks and processes design and (automatic) generation; 2) better understanding of crowds properties such as (soft and hard) skills, reliability, availability, capacity, and precision; 3) better tools for measuring and driving worker engagement.

3.3 Human Resource Management: From Workflows to Crowdsourcing

Cristina Cabanillas (Wirtschaftsuniversität Wien, AT)

The selection and allocation of human resources to activities has been increasingly researched in the field of workflow management in the last years. In particular, we have developed a language to define selection conditions on resources called RAL (Resource Assignment Language), we have presented mechanisms to automatically resolve RAL expressions and to analyze the business process resource perspective, and we are currently working on the prioritization of resources for allocation, specifically on the definition and resolution of preferences to generate a resource priority ranking.

We have evaluated a number of crowdsourcing platforms and we have found many differences among them regarding the selection and allocation of human resources to tasks. This makes us wonder whether our contributions on the field of BPM could be suitable in crowdsourcing platforms. In particular, we are interested in delving into the gap that should be bridged in order to improve, extend, or at least ease, the way in which human resources are managed in crowdsourcing systems.

References

1. C. Cabanillas, J. M. García, M. Resinas, D. Ruiz, J. Mendling, and A. Ruiz-Cortés. *Priority-Based Human Resource Allocation in Business Processes*. ICSOC 2013, pp. In press, 2013
2. C. Cabanillas, A. del Río-Ortega, M. Resinas, and A. Ruiz-Cortés. *CRISTAL: Collection of Resource-centrIc Supporting Tools And Languages*. BPM 2012 Demos, CEUR-WS, pp. 51-56, 2012
3. C. Cabanillas, M. Resinas, and A. Ruiz-Cortés. *Automated Resource Assignment in BPMN Models using RACI Matrices*. OTM 2012 (CoopIS'12), pp. 56-73, 2012
4. C. Cabanillas, M. Resinas, and A. Ruiz-Cortés. *Designing Business Processes with History-Aware Resource Assignments*. BPM 2012 Workshops (BPD'12), pp. 101-112, 2012
5. C. Cabanillas, M. Resinas, and A. Ruiz-Cortés. *Defining and Analysing Resource Assignments in Business Processes with RAL*. ICSOC 2011, pp. 477-486, 2011
6. C. Cabanillas, M. Resinas, and A. Ruiz-Cortés. *RAL: A High-Level User-Oriented Resource Assignment Language for Business Processes*. BPM 2011 Workshops (BPD'11), pp. 50–61, 2011

3.4 Human Computation and Crowdsourcing

Joseph Davis (The University of Sydney, AU)

URL http://kmrg.it.usyd.edu.au/node/587

Harnessing human computation through crowdsourcing and its integration with machine computation as a viable problem solving strategy for solving complex problems has emerged as an important research focus in recent years. I present the conceptual foundations of this approach and review some of the major contributions in this area.

3.5 Next-Generation Micro-task Crowdsourcing Platforms

Gianluca Demartini (University of Fribourg, CH)

At the eXascale Infolab in Fribourg, Switzerland we have been building hybrid human-machine information systems over the last 2+ years. Such systems involve both scalable/efficient data processing done by machines as well as effective processing of selected data done by humans. To build such systems it is important to have great control over the crowd performing micro-tasks. In my talk I will present current research directions in our group on how to improve current micro-task crowdsourcing platforms including quality assurance mechanisms by modeling workers in the crowd.

3.6 Data Analysis Tools for Crowdsourced Video Quality Tests

Klaus Diepold (TU München, DE)

Joint work of Klaus Diepold, Christian Keimel

We are active in moving Subjective Video Quality Tests from the laboratory space into the crowd. The ratings of a set of video sequences will be used to train a mathematical model in order to replace tests employing real eye balls for predicting the subjective quality of video sequences. To this end we propose to use tools from the multivariate data analysis toolbox, notably L-PLS Regression, to incorporate contextual information to further validate the mathematical model. This is a conceptual presentation void of results, featuring a list of expected outcomes which are up for discussion.

3.7 Supporting Exploration and Rapid Development by Citizens to Collect the Right Information through Crowd Sensing

Shinichi Konomi (University of Tokyo, JP)

To collect the information through crowd sensing, it is often required to define the goal and the method of data collection clearly in advance. However, it is not always possible to define them clearly in advance in many real-world situations. Therefore, I argue that there is the need to support exploratory activities in crowd sensing, in which participants can modify the relevant goal incrementally as they go about data collection activities. The Scene Memo system is an environment that supports exploratory crowd sensing based on a mechanism to share photos and tags in real time on smartphones and tablets. A small field study with the Scene Memo system suggested that shared tags can provide social cues and scaffold participants, thereby demonstrating the feasibility of a support mechanism for exploratory crowd sensing. We are also considering a rapid development environment of crowd sensing tools so that tools can be tailored easily to suit the changing goals of data collection.

3.8 Crowdsourcing Beyond the Desktop: Experiences and Challenges

Vassilis Kostakos (University of Oulu, FI)

Joint work of Vassilis Kostakos, Jorge Goncalves
Main reference J. Goncalves, D. Ferreira, S. Hosio, Y. Liu, J. Rogstadius, H. Kukka, V. Kostakos, "Crowdsourcing on the Spot: Altruistic Use of Public Displays, Feasibility, Performance, and Behaviours," in Proc. of the 2013 ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing (UbiComp'13), pp. 753–762, ACM, 2013.
URL http://dx.doi.org/10.1145/2493432.2493481

We discuss our work in moving crowdsourcing beyond the desktop. We present is the first attempt to investigate altruistic use of interactive public displays in natural usage settings as a crowdsourcing mechanism. We test a non-paid crowdsourcing service on public displays with eight different motivation settings and analyse users' behavioural patterns and crowdsourcing performance (e.g., accuracy, time spent, tasks completed). The results show that altruistic use, such as for crowdsourcing, is feasible on public displays, and through the controlled use of motivational design and validation check mechanisms, performance can be improved. The results shed insights on three research challenges in the field: i) how does crowdsourcing performance on public displays compare to that of online crowdsourcing, ii) how to improve the quality of feedback collected from public displays which tends to be noisy, and iii) identify users' behavioural patterns towards crowdsourcing on public displays in natural usage settings.

3.9 A Homo Ludens in the Loop

Markus Krause (Universität Hannover, DE)

Main reference M. Krause, "Designing Systems with Homo Ludens in the Loop," in P. Michelucci (Ed.), Handbook of Human Computation, Springer, 2014; to appear.
URL http://www.springer.com/computer/ai/book/978-1-4614-8805-7

For applications that are driven by input from a multitude of human raters, ensuring data reliability and organizing an interactive workflow constitute a complex challenge, especially when contributors are players. This talk introduces a novel approach to ensure data reliability of crowd-based and human computation systems. The proposed algorithm features the potential for direct feedback and interactivity in human computation games.

3.10 Towards Responsible Crowdsourcing

Martha A. Larson (TU Delft, NL)

Humans are the ultimate intelligent systems. Units of human work can be used to address the problems studied in the fields of pattern recognition and artificial intelligence. After years of research to crack certain tough problems, mere utterance of the phrase "human cycle" makes it seem like someone turned on a light in the room. Suddenly, we feel we are no longer feeling our way forward in darkness as we develop solutions. Instead, a bright world of new possibilities has been opened.

The excitement that crowdsourcing has generated in computer science is related to the fact that large crowdsourcing platforms make it possible to apply abstraction to human input to the system. It is not necessary to consider who exactly provides the input, or how they 'compute' it, rather the human processor can be treated as a black box. The magic comes when it is possible to make a 'call the crowd' and be sure that there will be a crowdworker there to return a value in response to that call.

However, crowdsourcing raises a whole new array of issues. At the same time that we excitedly pursue the potential of 'Artifical artificial intelligence' (as it's called by MTurk), it is necessary to also remember 'Human human computation'.

I am not an ethicist, and my first foray into crowdsourcing ethics is necessarily superficial. In fact, I started by typing the word "ethics" into my favorite mainstream search engine and picking a definition to study that seemed to me to be authoritative. However, I am convinced that the community of crowdworkers and taskaskers together form an ecosystem and that the main threat to this ecosystem is that we treat it irresponsibly.

In other words, we should not throw out everything that we have learned over centuries of human civilization about creating healthy and happy societies, stable economies and safe and fulfilled individuals in our quest to create new systems. Ultimately, these systems must serve humanity as a whole, and not disproportionately or detrimentally lean on the portion of the population that serves as crowdworkers.

The hopeful part of this undertaking is that it revealed many solutions to address ethical aspects of crowdsourcing. Some of them pose challenges that are just as exciting as the ones that motivated us to turn to crowdsourcing in the first place.

3.11 Crowdsourcing Challenges from Platform Provider's Point of View

Nhatvi Nguyen (Weblabcenter, Inc. – Texas, US)

Microworkers.com is an international Crowdsourcing platform founded in 2009, focusing on Microtasks. Until today over 450,000 users have registered at our platform from over 190 countries. This huge and diverse workforce is the key to the current success of Microworkers, but also imposed challenges both on the underlying infrastructure, as well as on support, e.g. handling service requests from or the resolving of disputes among users.

With my talk at the Dagstuhl seminar, I want to give some insights into the daily work of a Crowdsourcing platform provider. This can help to foster the discussion about new research directions. Here I would like to not only raise technical questions, but also social and legal ones, like "Can Crowdsourcing be used to support economies in developing countries?" or "Who is responsible for the healthcare of the workers?"'.

3.12 Social Protocols as a Basis for Social Computing

Munindar P. Singh (North Carolina State University, US)

Joint work of Munindar P. Singh, Amit K. Chopra

We are interested in investigating the foundations of social computing. We claim that today's social computing approaches provide a limited basis for modeling social relationships, which are the basis for the motivation and potential power of social computing. We propose an approach for understanding social computing as a means to enable the interactions among socially autonomous parties that involves a social protocol: the parties thus enact such a protocol.

A social protocol, in turn, is based on the expectations that it establishes between the participants. Examples of such expectations include domain-independent concepts such as commitments and authorizations as well as domain-specific concepts such as friendship. Associated with each expectation is a model of the trust that each participant may hold in the others, for example, whether they are competent to hold up the expectation and have the intention to hold up the expectation. We show how current social computing platforms and applications, such as Facebook and Reddit, may be understood in terms of the protocols that they embody.

The above applications incorporate the protocols into their design. However, our proposed way to thinking of protocols leads us to a simple architecture that would facilitate the development and maintenance of new applications. Specifically, the architecture would include social middleware with coverage for each family of social expectations used in a protocol. The middleware would support the interactions of the participants by carrying out the requisite bookkeeping.

This view leads to major research challenges including (1) modeling social protocols to verify their enactability and viability; (2) engineering middleware in a modular and composable manner to support new social expectations and new combinations of expectations in protocols; and (3) developing approaches for achieving agreement among stakeholders to lead to the specification of protocols that they can adhere to.

3.13 Current Research Challenges in Crowdsourcing

Phuoc Tran-Gia (Universität Würzburg, DE)

Joint work of Phuoc Tran-Gia, Tobias Hoßfeld, Matthias Hirth

The research of crowdsourcing in several aspects is part of the work of the Chair of Communication Networks – University of Würzburg. Within this area the current and previous research topics include, the design of mechanisms for crowdsourcing, analysis and modeling of platforms and their users, and the evaluation of new crowdsourcing use cases, e.g. in subjective QoE assessments. The questions of interest beyond are for example how to implement a quality control for tasks or how a platform will grow.

New research directions are for example the investigation of new types of crowdsourcing, e.g. real-time crowdsourcing or mobile crowdsourcing. In the talk during this seminar, I give an overview of the evolution of work organization, the different layers of the crowdsourcing processes and of current and previous work in this area conducted at the chair. I also present some new research challenges which could be addressed in the future.

3.14 Enterprise Crowdsourcing in IT Service Delivery

Maja Vukovic (IBM TJ Watson Research Center, US)

To increase the value of IT, enterprises rely on insights obtained by extracting large volumes of tacit knowledge about processes, products and people. This knowledge is not systematically discoverable, as it is unstructured and widely distributed among the experts in an enterprise. Typically this 'non-discoverable knowledge' is gathered in semi-automated way, which at best provides crude estimates, and doesn't scale. In this talk I will present an enterprise crowdsourcing (self-)service based on principles of 'wisdom of crowd' to enable rapid generation of a snapshot of the state of IT systems and operation, enabling for collaborative and distributed approach to knowledge gathering. I will discuss a number of use cases that engaged over 50K employees through crowdsourcing to uncover knowledge critical to IT services delivery. I hope to learn more about the current research efforts in academia with respect to incentive mechanisms and quality assurancel and how open APIs will change the nature of crowdsourcing.

3.15 Crowdsourcing and Crowdsensing Application Scenario

Florian Zeiger (AGT International – Darmstadt, DE)

This talk gives some ideas on how crowdsourcing / crowdsensing can be used in the area of safety and security. In more details the application example of environmental sensing is described from industry point of view and several challenges and open questions are discussed. In terms of scale of crowdsourcing and crowdsensing applications initially need to motivate participants and this motivation has to be kept up over time. Other challenges are present in the area of user privacy, reimbursement methods, data quality and integrity, efficient and reliable data collection, as well as architectural decisions and flexible support of various business models.

4 Overview of Additional Abstracts

4.1 Recommender Systems in Crowdsourcing Platforms

Kathrin Borchert (Universität Würzburg, DE)

I first got in contact with the topic crowdsourcing from theoretical point of view during my Master Thesis. Further, I approached this topic from a practical side during my internship at Microworkers.com which is a crowdsouring platform focused on microtasks. A typical crowdsourcing platform offers an unmanageable amount of jobs and tasks to the workers. They have to decide which kind of tasks fit their interests and skills based on a short description of the jobs. Thus, workers might accept tasks without being qualified. This could impact the quality of the results and the employers might reject their work. So the workers and employers are unsatisfied. The integration of a recommender system potentially improve the result quality and the satisfaction of the customers. There are two possible locations to integrate such a system in a crowdsourcing platform: on employer side to help them finding workers concerning their needs and on the worker side to recommend tasks with respect to their interests and skills. The main part of my Master Thesis will be the analysis of several recommender algorithms and approaches to find such tasks or workers.

4.2 Technical and Human Aspects in Crowdsourcing Research

Matthias Hirth (Universität Würzburg, DE)

Crowdsourcing offers a lot new possibilities in computer science. Suddenly not only algorithmic results are available in the computation process, but also subjective human judgments. This enables new types of services and fosters user centric evaluations like, QoE tests or network measurements on end-user devices.

However, in crowdsourcing systems human factors have a significant influence on the results. This leads to new challenges which require a holistic view on these systems including both their technical as well as their human aspects. Therefore the Crowdsourcing research at the Chair of Communication Networks aims to contribute to this holistic view by providing statistical analyses of crowdsourcing data, models of crowdsourcing platform, and, in collaboration researcher from psychology, a better understanding of the human factors, e.g. the trustworthiness of users.

4.3 Project Management Practices in Crowdsourcing Management

Deniz Iren (Middle East Technical University – Ankara, TR)

As a means to access a scalable and rather cheap workforce, Crowdsourcing has successfully been in use for quite some time, especially by risk-prone entrepreneurs. However utilization of Crowdsourcing in large organizational projects provides difficulties of management. There is no defined way of estimating the cost, time and quality of work when it is performed by a crowd. As a project manager and a researcher my goal is to develop mechanisms to make crowdsourcing measurable, estimateable and thus; more manageable.

I propose using the project management practices to solve this issue. The cost/time/quality of a crowdsourced job depends on a number of design decisions such. For starters I suggest focusing on the impact of the decision regarding which quality assurance mechanisms to use. We can model various types of quality assurance mechanisms to estimate the impact of their usage on the project cost, schedule and quality. Using these models with certain statistical methods we believe we can estimate this impact quite accurate.

4.4 Crowdsourcing of Multimedia QoE Subjective Experiments

Christian Keimel (TU München, DE)

The evaluation of multimedia Quality of Experience (QoE) in traditional laboratory-based experiments is time consuming and expensive. Using crowdsourcing to conduct these subjective tests online not only reduces time and cost, but also allows for the participation of a large and diverse panel of international, geographically distributed users in realistic user settings. But moving from the controlled laboratory environment to the uncontrolled crowdsourced online environment leads to new challenges. Some issues, as for example the assessment of the workers reliability, are similar to other crowdsourcing applications. Other issues, however, are more specific to the evaluation of multimedia QoE. Long established best practices for laboratory-based tests aiming at preventing biases or other unwanted effects in the quantification of the subjects' judgments, for example, can possibly not be adapted to the crowdsourcing test environment. Moreover, it is also unclear if all commonly used subjective test designs can be used or only a limited subset of the exiting designs. This raises the question, if it is perhaps necessary to develop completely new test designs for the crowdsourced multimedia QoE evaluation.

4.5 Motivation and Quality Assessment in Online Paid Crowdsourcing Micro-task Platforms

Babak Naderi (TU Berlin, DE)

During my PhD thesis, in Quality and Usability Lab – TU-Berlin, I am working on motivation and quality assessment in online paid crowdsourcing micro-task platforms. Although it looks that the main drive for workers is the money they earn, I am looking on different type of motivation and their effects on the quality of work. In addition I am trying to create a model for predicting worker's motivation, generated data quality based on workers' behaviors in platform.

5 Working Groups

The intention of the seminar was on the identification of a (inter-disciplinary) research agenda, common research methodology, joint (industry and academia) activities and collaboration for crowdsourcing. To this end, the participants were asked before the seminar to complete a short user survey indicating their main interests and potential discussion items. The results of this user survey can be found in Table 1. Some additional discussion topics were raised ('other') that are massive open online courses, social good applications, human computation, resource selection and quality assurance.

Table 1 Number and ratio of participants' interests in different research topics.

Discussion Item	Interest	Ratio
Practical issues and experiences	17	16%
Industry use cases for crowdsourcing	16	15%
Improvement mechanisms	13	12%
Interdisciplinary discussions, e.g. law, psychology	13	12%
Long-term perspectives	13	12%
Standardization and open APIs	9	8%
Non-for-profit and academic use cases for crowdsourcing	9	8%
Theoretical frameworks for evaluation	6	6%
Path towards deployment	5	5%
Other	6	6%

Based on the survey and the observations of interests, discussions, interactions among participants etc. during the seminar, four different working groups were formed during the seminar in order to discuss certain aspects in more detail, that are

(W1) long-term perspectives & impact on economics in five years,
(W2) theory – taxonomy and dimensions of crowdsourcing,
(W3) industry use cases,
(W4) crowdsourcing mechanisms and design.

5.1 Working Group 1: Long-Term Perspectives & Impact on Economics in 5 Years

Christian Keimel (TU München, DE)

Joint work of Gianluca Demartini; Klaus Diepold; Christian Keimel; Martha Larson; Nhatvi Nguyen; Phuoc Tran-Gia

The discussions in the focus group on long-term perspectives, ethical issues, social aspects & impact in 5 years on economics lead to the following conclusions: - Crowdsourcing is sustainable in the future but emerging public perception or legal regulation may come into play.

- Workers, requesters, lawyers and platform providers should work together to define a vision of crowdsourcing ahead of regulations.
- More crowd-building, resulting in specialized crowds.
- Crowdsourcing is opening new and thrilling aspects for empirical research, but commercial aspects need to be added.
- Microworking should strive towards a marketable skill development.
- Building economic capacity in geographical regions and social segments where needs exist.

5.2 Working Group 2: Theory – Taxonomy and Dimensions of Crowdsourcing

Tobias Hoßfeld (University of Würzburg, DE)

Joint work of Abraham Bernstein; Joseph G. Davis; Tobias Hoßfeld; Deniz Iren; Munindar P. Singh

Currently, we are in the early stamp collection phase of crowdsourcing which means that the landscape is not clearly known. Therefore, this working group aimed in analyzing the dimensions of crowdsourcing as a basic step towards a common taxonomy. The goal of the taxonomy is not to achieve automation (based on this taxonomy), but rather to have good guidelines for the design and development of crowdsourcing tasks and campaigns. According to the group discussions, the five major dimensions of crowdsourcing are for each task (1) who, (2) where (3) why, (4) what, (5) how. Their meaning is explained in the following with several examples to illustrate the dimensions properly.

Who? A more detailed question is what is the type of the crowd. Here we have to differentiate the actual performers conducting the task. A crowd required for a certain task can be described in terms of (i) anonymity, (ii) number of (reliable) performers, (iii) social structure of the crowd and prior relationships between performers, (iv) diversity of performers (with respect to cognitive, cultural, gender, or personal bias), (v) trust, (vi) and the communication structure within the crowd. But we also have to take into account additional individuals or parties involved in the crowdsourcing process, including (a) employers, task assigners, or promoters, (b) beneficiaries, or (c) platform providers. This means the entire stakeholder structure has to be taken into account. This brings us directly to the next question.

Where? Another dimension is the location or context of the task, the crowdsourcing platform and the performers. The context where the task is executed may be important, e.g. for crowdsensing. However, it has to be differentiated between a physical location and context. For example, the context of this seminar is the spirit of Dagstuhl, while it does not matter if this spirit appears in Kaiserslautern or in Saarbrücken. The crowdsourcing platform used has to be considered too, since it brings in not only its own crowd, but also certain mechanisms, which may affect the crowdsourcing task design.

Why? The incentives provided by the task and its execution are relevant towards a taxonomy. Thus, motivational diversity must be taken into account. Examples are (i) money as provided by commercial platforms like Amazon Mechanical Turk or Microworkers.com, (ii) fame and reputation e.g. for open source implementation, (iii) learning, i.e. the performer wants to benefit from executing the task by gaining new knowledge e.g. for open source implementation, (iv) altruism like social responsibility as observed for wikipedia, (v) side-effects like ReCaptcha where the actual task does not bring any incentives, but the task has to be conducted e.g. in order to get access to download software.

What? The type of goal or problem which is to be solved by the task is captured by this dimension. This includes the following aspects, (i) size of the task itself which may vary from contests to micro-tasks and therefore may require different mechanisms and design approaches, (ii) type of activity covering human computation, sensing, information foraging, evaluation, creation e.g. software or logo design, (iii) stopping condition of a tasks, i.e. whether the employer knows when the campaign is really done[1], (iv) possibility of evaluating the quality and reliability of the submitted results[2], (v) level of engagement of the performer which may be high e.g. crowdfunding, medium e.g. do something, or low e.g. sensing which is done by the user's device in the background, (vi) degree of control.

How? This dimension spans a variety of aspects on how the task is actually designed and executed. Thereby we differentiate on a higher level between (A) the method, mechanisms, or solution architecture as well as (B) the types of coordination mechanisms. Examples for coordination mechanisms are (1) aggregation of users from different platforms, of user results, and of tasks to macro-tasks, (2) selection of users or (sub-)tasks, (3) scheduling of the (sub-)tasks, or (4) resource allocation in terms of cost, time, and human efforts. To be more precise, the coordination mechanisms can be separated according to (i) the functionality, (ii) time, e.g. synchronized vs. asynchronous task execution by performers, e.g. real-time vs. serial execution, (iii) governance, (iv) coordination science framework.

Finally, the methods, mechanisms, or solution architectures address (I) degree of specificity, (II) degree of control, (III) locus of control, and (IV) the actual approach of the task. Thereby, we can differentiate between transparent tasks and obtrusiveness, direct vs. side-effect, piggyback e.g. sensing, parasitic e.g. captcha, or deliberate approaches.

Those five dimensions **Who, Where, Why, What, How** provide a basic framework for a taxonomy of crowdsourcing which may lead to useful guidelines for the design and development of crowdsourcing tasks and campaigns.

---

[1] The stopping condition is sometimes a non-trivial task, as it may require real-time evaluation of the submitted crowdsourcing results or a continuous execution of a task is necessary.

[2] For tasks like subjective testing, it is difficult to evaluate quality or reliability of results, as no correct answer and gold data exists. Thus, other quality assurance and reliability mechanisms are needed.

5.3 Working Group 3: Industry Use Cases

Maja Vukovic (IBM TJ Watson Research Center, US)

Joint work of Kathrin Borchert, Shi'nichi Konomi, Vasilis Kostakos, Maja Vukovic, Florian Zeiger

This workgroup has tackled the challenges in identifying trends and understanding what makes crowd sourcing applications successful, with focus on, but not limited to industry domain. We started this discussion session by having each participant share their favorite example of the crowd sourcing application. These ranged from Netflix, GalaxyZoo, to Amazon Mechanical Turk applications. The team concurred that often, most successful examples are the ones with most significant publicity and exposure. At the same time, the team has observed that there is still lack of convergence towards a unified crowd sourcing platform. There are numerous custom platforms, each of which is a solution for different type of crowd sourcing application. Finally, we discussed the notion of quality of crowd sourcing tasks, and let of clear metrics. Perception of quality often depends on different stakeholders in crowd sourcing process.

We concluded the discussion by identifying following five key challenges:

1. Advanced research and development of best design practices for crowd sourcing
2. A framework for evaluating crowd sourcing design considerations: task complexity, quality, price, and reaction time.
3. How can we "pre-certify" crowds, so that they can be quickly deployed to work on critical tasks, while maintaining expected quality levels.
4. How can we bring back and integrated research results from related crowd sourcing efforts?
5. There is an opportunity to engage community to reduce the isolation of crowd sourcing research.

5.4 Working Group 4: Crowdsourcing Mechanisms and Design

Matthias Hirth (University of Würzburg, DE)

Joint work of Alessandro Bozzon; Cristina Cabanillas; Matthias Hirth; Andreas Hotho; Markus Krause; Babak Naderi

This report summarizes the discussion and the outcome of the group work on "Crowdsourcing Design and Mechanisms" of Alessandro Bozzon, Cristina Cabanillas, Matthias Hirth, Andreas Hotho, Markus Krause, and Babak Naderi during the Dagstuhl seminar on "Crowdsourcing from theory to practice". The discussion mainly focused on two questions:
1) *"How can crowdsourcing experiments be designed in such a way, that they their results can be reproduced and validated by other researchers?"*
2) *"How and which steps in crowdsourcing measurements can be abstracted or generalized to speed up the task design process and to later provide guide lines for task design?"*.

Currently, experimental driven research papers in the field of crowdsourcing include detailed descriptions of the test setup, the proposed/used methodology, and the obtained results. In contrast, details about the actual participants of the test, their demographics, their hardware equipment, or their cultural background are often neglected. However even the results of simple tasks, like the transcriptions of texts on images can be influenced by these characteristics, because workers understanding the content of the texts can correct characters or passages which are unreadable based on the context. This is not possible for workers which are unfamiliar with the texts' language. Therefore, the results of subsequent test repetitions might significantly differ depending on the participants, even if the task setup remains unchanged. To address this challenge, the group discussed about *how crowdsourcing experiments can be designed in such a way that they their results can be reproduced and validated (by other researchers)*.

During the discussion, the group agreed that some preliminary questions have to be addressed in order to tackle reproducibility of task results. First, similarity metrics of tasks have to be defined in order to identify task groups, among which the result should be comparable. Second, ways have to be found to measure similarity of crowds and to make crowds comparable. This can help to consider crowd specific biases and how to remove them in order to compare the results from different crowd provider. And finally, it has to be considered which parameters of an experiment have to be reported in a publication. Is it enough to simply mention the used crowdsourcing platform, or is it required to include detailed demographics of the workers? Crowdsourcing experiments are more difficult to repeat than measurements in technical environments, because of the human factors and possible varying behavior of the users. However, similar issues also exist in the field of psychology or human computer interaction research. Methods applied in these files might be suitable for conducting reproducible crowdsourcing experiments, even if some adaptions might be necessary. But also new methods have to be developed to consider crowdsourcing specific issues, e.g. technical changes in the crowdsourcing platform which affect the users' behavior. A simple solution to overcome some of these issues is using only a single crowdsourcing platform and performing multiple iterations. However, it is doubtful if these results would be really representative for crowdsourcing in general.

Another big challenge in current crowdsourcing research is the design and implementation of the crowdsourcing tasks themselves. Due to the lack of a standardized task descriptions and design guidelines, many researchers perform redundant work. This challenge was also discussed in the group and the participants tried to answer the questions *"How and which steps in crowdsourcing measurements can be abstracted or generalized to speed up the task design process and to later provide guide lines for task design?"*

One step to reach this goal is finding general properties of tasks, which can be used to describe tasks in a standardized way. A possible solution could be a semantic description of tasks. This can enable an abstraction of crowdsourcing task creation, e.g. via crowdsourcing programming languages, which has again benefits and drawbacks. The abstraction would enable an easy setup of crowdsourcing experiments, but would not reduce the number of influence factors on the results. Instead the abstraction layer hides the influence factors and makes them harder to analyze.

To enable the development of generalize task properties and design guidelines, it is helpful to separate experiments into different layers, namely

- the design of the experiment workflow,
- the design of actual tasks, and
- the evaluation of the results using technical and algorithmic tools.

This decomposition into layers fosters the development of design patters for tasks. Here similar ides from distributed computations approaches (e.g. Hadoop) can be included in the design of crowdsourcing workflows and task design. In this area some research studies also already available. To describe tasks in general, different dimensions are required. The group identified the following preliminary set of task dimensions, which can be used as a starting point but is definitely not complete yet.

- Task requirements
  - Time to complete
  - Required skills of the workers
  - Required diversity of workers
- Task properties
  - Decomposability
  - Possibility to complete multiple instances in parallel
  - Complexity
- Task output
  - Expected confidence level
  - Data properties
  - Data complexity
- Task input
  - Diversity of the task data in terms of complexity (e.g. skewness of the data, might be analyzed using machine learning)
  - Data properties (size, ...)
  - Data complexity

Further the impact of pilot studies on the result quality was discussed. Pilot studies can either be specialized tasks to evaluate the qualification of workers for the actual research study or shortened version of tasks to evaluate the task design. This approach is assumed to reduce the costs for the experiments, because pilot tasks are usually paid less and reduce the number of required test iterations, as the task design is already optimized based on the pilot task results. Further the quality of the experiments results is also supposed to be better, because unqualified workers are filtered based on their performance in the pilot studies. Nevertheless, pilot studies could also influence the experiment results in unexpected ways or even falsify them. For example, demographical properties of the test participants might be skewed due to filtering of the workers. This in turn can lead to biased results. Further, in subjective experiments, pilot studies might change or influence expectations of participants which can lead also to based results. During the discussion, the participants agreed that the influence of pilot studies on the result quality highly depends on the specific task and that no general conclusions can be drawn.

6 Open Problems and Challenges for Crowdsourcing

During the seminar, several open problems and challenges for crowdsourcing were discussed. Although most of those aspects are covered in the summaries of the working groups and the presentation sessions, an (unsorted) list of concrete questions is given below to provide a comprehensive overview.

- What is an appropriate taxonomy for classifying and analyzing crowdsourcing systems? How can crowdsourcing tasks be grouped based on their task complexity and along key challenges in successfully harvesting expertise of large human networks?

- Which use cases and applications will exploit the potential of crowdsourcing?
- How does the research community approach improved crowdsourcing mechanisms e.g. for quality and cost control or reliability of users and devices? Which requirements and challenges occur for particular operational conditions, like ubiquitous crowdsourcing due to the user mobility in time and space?
- How to design incentive schemes for coordinated problem solving of the crowd among individual humans with their own goals and interests? How to realize gamification of work for improved user engagement? How to identify expertise of users? How to implement such incentive schemes technically?
- How can the experiment and task design be standardized? Which kinds of APIs or templates are promising and useful in practice?
- What are the objectives to be fulfilled and the necessary capabilities of platforms towards the provision of Future Internet services built on top of crowdsourcing facilities?
- How can crowdsourcing systems be evaluated? Which common research methodologies are applicable? Which theories and models from a number various fields are applicable, including artificial intelligence, multi-agent systems, game theory, operations research, or human-computer interaction? How to include human-centric measures such as costs, availability, dependability and usability, including device-specific properties in evaluation frameworks?
- How does the research agenda for crowdsourcing look like in the next years?
- How can crowdsourcing experiments be designed in such a way, that they their results can be reproduced and validated by other researchers?
- How and which steps in crowdsourcing measurements can be abstracted or generalized to speed up the task design process and to later provide guide lines for task design?
- How will open APIs change the nature of crowdsourcing?
- How to motivate participants especially in terms of scale of crowdsourcing and crowd-sensing applications and how to keep the motivation up over time?
- How can we “pre-certify” crowds, so that they can be quickly deployed to work on critical tasks, while maintaining expected quality levels.
- How can we bring back and integrated research results from related crowd sourcing efforts?
- Can Crowdsourcing be used to support economies in developing countries?
- Who is responsible for the healthcare of the workers?

The following challenges are observed for crowdsourcing by the seminar’s participants.

- Need for better conceptual abstractions for crowd tasks and processes design and (automatic) generation;
- Better understanding of crowds properties such as (soft and hard) skills, reliability, availability, capacity, and precision.
- Better tools for measuring and driving worker engagement.
- Advanced research and development of best design practices for crowd sourcing.
- A framework for evaluating crowd sourcing design considerations: task complexity, quality, price, and reaction time.
- Engaging community to reduce the isolation of crowd sourcing research.
- Commercial aspects to be added to current crowdsourcing research.
- Microworking should strive towards a marketable skill development.
- Managing human resources in crowdsourcing systems.

- Integration of crowdsourcing with machine computation.
- Improvement of current micro-task crowdsourcing platforms including quality assurance mechanisms.
- Modeling workers in the crowd for optimization purposes and improved mechanisms.
- Specialized crowds and building of specialized crowds for particular campaigns in contrast to wisdom of the crowd.
- Dynamic adaptation of crowdsourcing systems during operation, e.g. to suit the changing goals of data collection in environment sensing systems.
- Evaluation of new crowdsourcing use cases.
- Organizing and managing the different layers of the crowdsourcing processes.
- User privacy of crowdsourcing users, e.g. in crowdsensing campaigns.
- Appropriate reimbursement methods and flexible support of various business models.
- Architectural design of crowdsourcing and machine clouds.
- Better understanding of incentives for participation in crowdsourcing for improved motivational design of campaigns.
- Organization of interactive workflows and direct feedback in crowdsourcing systems for improving reliablity and data quality.
- Responsible usage of crowdsourcing and ethical aspects.
- Building economic capacity in geographical regions and social segments where needs exist.
- Modeling social protocols to verify their enactability and viability.
- Developing middleware in a modular and composable manner to support new social expectations and new combinations of expectations in protocols.
- Approaches for achieving agreement among stakeholders to lead to the specification of protocols that they can adhere to.
- Emerging public perception or legal regulation of crowdsourcing and the impact on sustainability.
- Interworking of employers, employees, lawyers, and platform providers to define a vision of crowdsourcing ahead of regulations.

7 Panel Discussions

Based on the abstracts of the talks and interests of the participants, the seminar was structured accordingly. In five presentation and discussion sessions, the diverse aspects of crowdsourcing were elaborated. A joint late night session was organized with the parallel seminar "Cloud-based Software Crowdsourcing" to discuss crowdsourcing with respect to ethics and its relation to social computation. The topics of the sessions covered

(S1) crowdsourcing in general,
(S2) industry use cases,
(S3) crowdsourcing design and engineering,
(S4) programming and implementing crowdsourcing,
(S5) applications of crowdsourcing,
(S6) social computing and ethics.

A summary of the discussions during those sessions can be found in the following.

7.1 Presentation Session 1: Crowdsourcing in General

Matthias Hirth (Universität Würzburg, DE)

The session was opened by **Phuoc Tran-Gia** giving a talk on *"Current research challenges in crowdsourcing"*. In his talk, he detailed on the changing granularity of work enabling the transition from traditional outsourcing to crowdsourcing. Further, the similarities between human and machine clouds were discussed, as well as possible interconnections between them. Finally, influencing factors on the data quality and current research challenges were presented.

In the following discussion Munindar Singh asked, whether the granularity of tasks influences the quality of the resulting data. Phuoc Tran-Gia mentioned that there are some correlations, but a suitable selection of the workers can help to minimize the impact of this factor.

Thereafter the question was raised, what defines crowdsourcing. Joseph Davis mentioned that the one key aspect is that the workers are not chosen for a specific task, but can the tasks is distributed in an open call to an anonymous crowd. However, Martha Larson replied that the current anonymity of the employers and workers it no a mandatory property of the crowdsourcing system, but results from the lack of appropriate features in the current platforms. Tobias Hoßfeld had a similar point of view as Joseph Davis, as he stated that a relevant characteristic of crowdsourcing is that the workers select their work. Maja Vukovic introduces the payment system as criteria for crowdsourcing, because in crowdsourcing payment is usually based on a competition approach and the workers are only paid based on their results. Vassilis Kostakos asked whether it is important at all to have a common definition of crowdsourcing and Deniz Iren proposed that there can be different types of crowdsourcing. However, a taxonomy of these different types is required develop different management strategies. In reply to the analogue of human and machine clouds, Martha Larson stated that it is important that people know their applications are human powered. Klaus Diepold replied that in his opinion, not all user of cloud solutions are interested about the detailed realization of the product. Abraham Bernstein agreed with Klaus Diepold and suggested to foster ethical standards for crowdsourcing providers, similar to fair trade initiatives.

Munindar Singh stated that for some tasks, customers have to be explicitly notified that the used solution is crowdsourcing based, e.g. if the tasks includes the processing of confidential data. In the end a few statements about the future drivers of crowdsourcing were made. Phuoc Tran-Gia mentioned that crowdsourcing can be used as instrument to provide work in developing countries. Martha Larson drew an attention to the ethical challenges of this approach. Alessando Bozzon stated that crowdsourcing will be successful, because it enables fast and scalable problem solving.

The second talk about *"Human Computation and Crowdsourcing"* was given by **Joseph Davis**. Joseph Davis illustrated that the idea of human computation has a long tradition. Already in 1794, Gaspard de Prony hired unemployed hairdresser who knew only basic calculations for producing logarithmic tables. Joseph Davis introduced Crowdsourcing as a way to integrate human and machine computation, which enables the combination of the power of computers and human intelligence to solve complex problems that are beyond the scope of existing AI algorithms. This especially applies to problems involving conceptual thinking or perceptual skills. However, it is still challenging to combine and aggregate the results obtained from different workers. Further, a good understanding of the applied aggregation mechanisms is required, because different mechanisms generate also different final results.

After the talk a discussion about Open Innovation arose. Tobias Hoßfeld asked, whether it is better to have larger number of homogeneous users or a smaller but more diverse group.

Joseph Davis replied that the optimal crowd composition depends on the tasks, but in most cases a bigger diversity is more important. Phuoc Tran- Gia addressed the issue of intellectual properties and rewards in crowd contests and Joseph Davis agreed that this is still an open problem. Maja Vukovic further mentioned the problem of unintended outcome of large scale crowdsourcing task. Even if user data is anonymized before publication, it might still be possible to track individuals and de- anonymization user profiles.

7.2 Presentation Session 2: Industry Use Cases

Kathrin Borchert (University of Würzburg, DE)

The first presentation was given by **Maja Vukovic** about "Enterprise Crowdsouring in IT Services Delivery". She described a use case of crowdsourcing in the background of enterprise IT transformation. In this context there are different tasks and challenges to solve and the required knowledge is often available in the enterprise crowd. An example is to find the hosting servers of specific applications or to define the production server out of an amount of registered. In their crowdsourcing model the first step is defining tasks and then finding the specific crowd with the knowledge and the skills to send them requests. The requested people can devide the tasks in subtasks and delegate them to other participants. After completing the whole task the workers get points as payment. The first question was raised by Phuoc Tran-Gia: "What was the incentive for the people to participate?". Maja Vukovic answered that the gathered data is disclosed to the participants, so they can also use it. She explained that it took 3 month to collect the information about the application owners, the last 1% took quiet long. Here Phuoc Tran-Gia asked for the quality assurance techniques for the results. The first 20% of the results have been cross validated by asking the application owners on which server the applications are hosted and later asked the server owners which applications are hosthed on their servers, was the answer of Maja Vukovic. Then Munindar P. Singh liked to know if there are any savings by using the crowd approach. Maja Vukovic replied that even with considering the overhead of the crowdsourcing efforts a huge amount of money was saved. The next question was requested by Vassilis Kostakos: "What is the impact of the approach on the business itself? People do no longer come together, just work remote?" This approach offers a possibility to find knowledge experts was the response. Furthermore, Vassilis Kostakos asked for side effects and if workers do more microtasks than their regular work. Maja Vukovic replied that nowadays, worker often distributed they work to others instead of doing it by themself. This might be a general development. Phuoc Tran-Gia liked to know if they also tested the quality of experience? But they did not analyzed that yet. The same answer was given to the question "How much workforce for the regular work did you loose by asking people to perform microtasks?" raised by Abraham Bernstein. The panel starting a discussion about the point system and what is their impact. Maja Vukovic told them that the points are only shown to participants. Then Abraham Bernstein liked to know what the usage of the points is. They are just a leaderboard was the explanation. Deniz

Iren asked the next question: "What happens to inaccurate results?" The answer was given by Maja Vukovic. She said that 80% of the collected data was correct. So the wrong part was not significant. After that, Vassilis Kostakos asked for legal and general issues. Maja Vukovic replied that there are some internal issues as privacy, spamming due to the high amount of email and employment status of participants. Nhatvi Nguyen focused back to the leaderboard by requesting a question concerning the implementation of its. He asked if they added penalties if a participant did not work accurate. But they did not. Martha Larson liked to know if the task was more about gathering data or updating the information. Maja Vukovic responded that the crowd helped gathering the data but then they were no longer interested in it. Further, she explained that the other crowdsourcing projects in IBM are crowdbased translation and crowd coder as Phuoc Tran-Gia asked for them. The last question concerning the presentation was raised by Tobias Hoßfeld: "Are there now innovate use cases research at the moment?" Maja Vukovic negated.

The topic of the second talk was "Crowdsourcing/Crowdsensing Application Scenarios" and was given by **Florian Zeiger**. He presented the usage of crowdsensing by explaining a project about environmental sensing concerning air pollution. Here, the participants get sensor devices which are configurable by a smartphone via bluetooth. The goal is to detect pollution events, hot spots and to localize the source of them. Tobias Hoßfeld asked the first question about the realization of the source localization. Florian Zeiger answered that if the model is not able to explain the pollution it asks for more data. Tobias Hoßfeld continued asking: "How can you trust the measurement results?" Florian Zeiger responded that they have good ways to verify data of fixed sensors, but it is different for crowd-measurements. It is difficult to manage that in an open system because the motivation of the users differ and it is possible for them to manipulate the results. He continued his talk with the presentation of open challenges and questions, e.g. scale of the system, privacy of the users, the support of different business models, the data quality and the reliability. The next question was raised by Joseph Davis: "40 million people downloaded an app for managing traffic status and maintaining map. Is there a community based approach, where a group maintains the data instead of the company?" Florian Zeiger answered that it depends on the ecosystem which shall be developed. There are possible scenarios in which this approach might be applicable, but there is no general answer. Joseph Davis also liked to know what the motivation of the users is and if there a community beyond the system? There is a system to register the people replied Florian Zeiger and the main motivation to participate is the interest in how the own environment is. But if it is good they lost interest in it. Then Babak Naderi asked for the experience on the usability of the system. The system is currently in a very early stage, limited to this area and there are no clear results, was the given answer. Then Klaus Diepold requested if there are a desired sensor resolution. Florian Zeiger responded that it depends on the geographical setup. In New York it would be great to have a sensor every few hundred meters which are measuring every 30 seconds. On german cities it is easier. That means it is not necessary to install sensors in the same high resolution. "Can the platform be bought?", was the next question of Andreas Hotho. Florian Zeiger explained that its only a prototype and the system is in the evaluation phase. So only user have access to the data and the measurement results are not sold. Someone asked if the platform is webbased. Florian Zeiger negated. The system is closed in the company. Abraham Bernstein brought up the question why these sensors are not deployed to public workers or soldiers to increase the sensor resolution? Florian Zeiger expounded that this is like the chicken-egg problem. Measurement probes are required to estimate the gain, but companies only buy if they know the possible gain. The last question concerning the presentation was raised by Gianluca

Demartini. He liked to know if the results are different than the results of the fixed sensors. Florian Zeiger affirmed.

Nhatvi Nguyen gave the last talk of this session. It was about crowdsourcing platforms from an operators point of view. The focused platform was Microworkers.com which mostly provids microtasks. Nhatvi Nguyen presented an overview of daily challenges which he as provider has to solve and explained the basic functionalities of the platform. One part is the competition between the workers. That means there are no guarantee to submit the completed task because the workers do not like task lock in the past. Martha Larson asked if Microworkers.com set up on the completion approach? That means workers will not get their money if they are not fast enough? Nhatvi Nguyen explained that there are two setups. The basic approach is race based and the other campaign type uses a time based approach. Cristina Cabanillas liked to know who decides if the workers get paid. Nhatvi Nguyen responded that in general the employer has the decision. But there is also a review system to prevent the employers from mis-rating tasks. That means Microworkers.com might interfere if the ratings are unfair. Babak Naderi revisited the topic of task completion by asking if the quality is better using the race approach or the time based approach. Race based campaigns are only used for simple tasks. To enable the time based setup the employers add a second layer. So the time based approach offers the request of more difficult tasks was the response of Nhatvi Nguyen. Babak Naderi continued asking: "Do the workers argue that they get no money?". The speakers affirmed: Microworkers.com will change the system to time based. He continued with the presentation and explained the payment system. The payment varies between the countries. So it is important to verify the location of the workers. He also told us that to improve the result quality Microworkers.com has a reviewing system for the campaigns. Here Joseph Davis liked to know if Microworkers.com is doing better than Mturk because of the manual reviews. Nhatvi Nguyen answered: "Yes, we put effort in this". He explained that Microworkers.com also runs cheat detection to improve the quality of the results. Then Martha Larson raised the question: "Does the black and white judgment, cheater/ not-cheater, of worker help or stand in your way?" Nhativ Nguyen gave the answer that the review feature helps to overcome this. Martha Larson also liked to know: "Using crowdsourcing for the first time is like playing a video game. Workers are offended by the term cheater, employers just want to get rid of cheater?". Microworkers.com gives multiple changes to the workers before classify them as cheaters and kick them out was the reply of Nhatvi Nguyen. The next asker was Florian Zeiger. He requested if there is an influence of the price on the quality in respect to the explained price model. Joseph Davis referenced to a paper and said that there is an optimal value of wages. Then Alessandro Bozzon changed the topic to the ethical aspects of crowdsourcing, e.g. the fair treatment of the workers including fair prices. He asked: "Should social aspects be addressed by Microworkers.com or by the employers?" Nhatvi Nguyen responded that Microworkers.com should foster it. Abraham Bernstein raised up a discussion with his question if the listed problems or issues are really crowdsourcing issues or real life issues? The discussion is focused on the topic which party is the employer and who is responsible for the workers. There is no clear answer to this. After that discussion Andreas Hotho liked to know how Microworkers.com reacts on jobs concerning grey level areas, for example downloading iPhone apps and requesting positive reviews. This kind of campaigns are removed on request told us Nhatvi Nguyen. Then Joseph Davis requested how Microworkers.com calculates the minimum wages, because it is difficult to set the price and to define what is a fair price. Nhatvi Nguyen explained that by manually reviewing the campaigns the price is set to a meaningful way. So if the price is too low, they ask the employer to raise the price.

7.3 Presentation Session 3: Crowdsourcing Design and Engineering

Christian Keimel (TU München, DE)

Markus Krause: *Designing Systems with Homo Ludens in the Loop*

The classification in the proposed gaming applications is only possible if the ground truth is available and the conclusion by the proponent was that games are useful human computation tools, but do not elicit intrinsic motivation, as the payment is still important. With respect to "magic recipes", the contributor remarked that "gamification sucks" and that the player should be put first i.e. the desires of player are paramount. In particular, task and game ideas need to be able to be linked and also the supporting infrastructure needs to be sufficient for the number of gamers. The complexity of the games did not change over time and the workers were provided with a button in the game, leading to a description of the scientific purpose of the game, but were not told directly the game's purpose in order to avoid the puppet master effect, as fun should be the main motivator for the game. Moreover the data was anonymized. The contributor recommends gamification, but suggests that one should be careful, and there should be nothing to lose for players, especially that there should be nothing dangerous for the players. Another question raised was if there are only certain human intelligence tasks that are suitable for a game or if there are limitations on possible human intelligence tasks. In particular, how much of a game the game should be for crowdsourcing tasks with gamification. A-priori tests in a laboratory can be used to identify overall issues with the gamified task design before a larger deployment in the crowd. In the provided example game for image annotation, between 240-1000 users participated.

Alessandro Bozzon: *Crowdsourcing Engineering*

The design of the underlying data is important for the task design and the data needs to be understood, possibly with crowdsourcing, before the design can be made accordingly. Also it was mentioned that current crowdsourcing platforms mostly focused on questionnaires. The experiments with the contributor's framework suggest that in crowdsourcing with social networks it is important how much the task requester's connections "care" about the requester and the more people "care" about the requester, the faster the task is finished. For traditional crowdsourcing it takes usually longer and corresponding data is already available in literature. Also it may be sensible to retain qualified workers that performed well in past tests and build task specific communities. Such task specific communities are different to groups established by mediator platforms, as these platforms do not know the details and purpose of a task as well as the requester. It was suggested that it may be necessary in future to build a real relationship between workers and requesters in order to address potential ethical and social issues.

Shi'ichi Konomi: *Citizen sensing, applications, transportations, rapid prototyping, collaboration, exploration*

One of the main conclusions by the contributor is that one size may not fit all and that crowd sensing can be improved by designing the right support tools e.g. explorations and rapid development tools for crowd sensing applications, allowing for a rapid adaptation to novel crowd sensing tasks. In the study of the contributor, about 30 users were used for trialling the wearable prototype in the initial crowd sensing deployment.

7.4 Presentation Session 4: Programming and Implementing Crowdsourcing

Deniz Iren (Middle East Technical University – Ankara, TR)

Abraham Bernstein: *Programming the Global Brain with CrowdLang and CrowdOS – Challenges and Ideas*

"Before the Internet most collaborators had to be sufficiently close by to work together towards a certain goal. Now, the cost of collaborating with anybody anywhere on the world has been reduced to almost zero. As a result large-scale collaboration between humans and computers has become technically feasible. In these collaborative setups humans can carry the part of the weight of processing. Hence, people and computers become a kind of 'global brain' of distributed interleaved human-machine computation (often called collective intelligence, social computing, or various other terms). Human computers as part of computational processes, however, come with their own strengths and issues.

In this talk we take the underlying ideas of Bernstein et al. (2012) regarding three traits on human computation—motivational diversity, cognitive diversity, and error diversity—and discuss the challenges and possible solution approaches in order to embrace these traits as strengths instead of weaknesses."

Following issues were raised in the discussion part:

Martha Larson: How do you define success in so called global brain?

– Both Linux and Wikipedia are examples of a successful global brain. The successful examples exists however we do not yet know how to develop them systematically.

Vassilis Kostakos: How would you deal with the fact that people contribute from different countries with different laws and culture?

– As researchers (of computer science) we are not the first people in the history to encounter the globalization problem. Humanity faced this problem before and overcome it. We just have to find the ways to deal with it according to the labor laws.

Alessandro Bozzon: Does global brain have to be designed or can it develop by itself organically like cities do?

– There are cities built with detailed planning. Depending on what you are trying to accomplish, hierarchical design may not be a bad thing.

Joseph Davis: Can you specify abstract model of the global brain?

– I would like to do that but I don't know how. Not yet. I would like to use the diversities between human and machines to produce a better solution. But in the end we may take an approach similar to software engineering.

Gianluca Demartini: *Next-Generation Micro-task Crowdsourcing Platforms*

"At the eXascale Infolab we have been building hybrid human-machine information systems over the last 2+ years. Such systems involve both scalable/efficient data processing done by machines as well as effective processing of selected data done by humans. To build such systems it is important to have great control over the crowd performing micro-tasks. In my talk I will present current research directions in our group on how to improve current micro-task crowdsourcing platforms including quality assurance mechanisms by modeling workers in the crowd."

Following issues were raised in the discussion part:

Phuoc Tran-Gia: How do you know who is online to make real-time contributions? How can you count on their real-time contributions?

– It depends on the budget and the task completion time. One increases when the other one decreases.
Cristina Cabanillas: Some quality assurance mechanisms are needed. Selecting the most trusted individual based on Facebook data can be misleading.
– I agree. It is still a work in progress.
Phuoc Tran-Gia: Crowd selection has been done by other crowdsourcing platforms before. What is new here?
– We aim at crowd selection in an algorithmic manner in order to optimize it.

Cristina Cabanillas: *Human Resource Management from Workflows to Crowdsourcing*
"The selection and allocation of human resources to activities has been increasingly researched in the field of workflow management in the last years. In particular, we have developed a language to define selection conditions on resources called RAL (Resource Assignment Language), we have presented mechanisms to automatically resolve RAL expressions and to analyze the business process resource perspective, and we are currently working on the prioritization of resources for allocation, specifically on the definition and resolution of preferences to generate a resource priority ranking.

We have evaluated a number of crowdsourcing platforms and we have found many differences among them regarding the selection and allocation of human resources to tasks. This makes us wonder whether our contributions on the field of BPM could be suitable in crowdsourcing platforms. In particular, we are interested in delving into the gap that should be bridged in order to improve, extend, or at least ease, the way in which resources are managed in crowdsourcing systems."

Following issues were raised in the discussion part:
Gianluca Demartini: Can standard resource modelling techniques be used for crowdsourcing?
– Modeling the resources within an organization is straightforward however doing it in crowdsourcing is not.

7.5 Presentation Session 5: Applications of Crowdsourcing

Babak Naderi (TU Berlin, DE)

Vassilis Kostakos: *Social Media, Public Displays, Crowdsourcing*
Most of discussions were around the study: Rogstadius et al 2011 "An Assessment of Intrinsic and Extrinsic Motivation on Task Performance in Crowdsourcing Markets". To evaluate the intrinsic motivation authors published two equal task with different description regarding to organization whom is going to use the result, one profit and the other non-profit company. They conclude extrinsic motivators increase workers' willingness to accept a task & the speed at which a task is completed. However intrinsic motivators improve output quality. The discussion was focused on morality & ethical issues in designing the experiment. Questions were if the participants have been informed afterwards that the images and tasks were fake or how do you expect that workers do not cheat when you are manipulating task description. Another interesting part was deploying tasks on touch interaction screens in the campus which leads to engage more workers than similar task running in MTurk meanwhile. However due to the time limit this part was not discussed.

Klaus Diepold: *Data Analysis Tools for Crowdsourced Video Quality Tests*
Klaus Diepold has presented application oriented perspective to crowdsourcing with the goal of optimizing video services for visual quality. Their effort for replacing subjective visual quality testing in laboratory with crowdsourcing approach and their mathematical model for predicting quality have been shown. In this context, using crowd sourcing test leads to reduce cost, speedup testing, possibility of larger tests (more reliable statistics), and improved diversity. In this session different points have been discussed: First of all considering outlier detection it has been discussed if consistency of responses should be taken in account rather than accuracy for detecting outlier. When a worker is jugging all the time same thing it shows his/her different point of view. Second discussion point was what type of features should be provided by crowdsourcing platforms for facilitating conducting such a studies? Different points have been suggested like delivery mechanisms should not affect the quality of content (high bandwidth for serving videos), and while sessions of quality judgments are long (i.e. about 20 minutes), therefore platform should take care about consequences (e.g. worker becomes tired). Another comment regarding to this perspective was exploiting the crowdsourcing idea for doing science (here improve encoder using mathematical analysis) is a potential use case for crowdsourcing. Other suggesting like using real-time feedback of crowd regarding to encoding mechanism, effect of intrinsic motivation in subjective quality tests, and current state of research in this area has been discussed.

7.6 Late Night Session: Social Computing and Ethics of Crowdsourcing

Tobias Hoßfeld (Universität Würzburg, DE)

For the late night session, the participants from the parallel seminar on "Cloud-based Software Crowdsouring" (Dagstuhl Seminar 13362) organized by Michael N. Huhns, Wei Li, Martin Schader and Wei-Tek Tsal were invited to join the discussions. On one hand ethical issues in the area of crowdsourcing were raised in a stimulus talk by Martha Larson (TU Delft). On the other hand, Munindar P. Singh (North Carolina State University) intended to provoke with his talk on the critique of current research in the area of social computing and crowdsourcing.

Munindar P. Singh: *Honest-to-Goodness Social Computing: Critique of Current Research and a Call to Arms* In social computing, the word "social" is introduced. Thereby, the social relationships are the basis for the motivation and potential power of social computing. The social state is a snapshot of a set of social relationships, while social computing means to compute the social state, that is, operate a social machine. Doing so involves conceptually modeling and specifying a social machine. The principals are autonomous parties i.e. people or organizations who are accountable. There may be a hidden principal behind the machine. On top of communication, there is (a) communication transport i.e. the protocol, (b) machine which means API and communication, and (c) interactions combining protocol and social middleware. The social computing architectural schematic is composed of several layers that are (i) social application: employing social relationships, (ii) social middleware: maintaining social relationships, (iii) distributed substrate: computing, communications, information. This architecture is grounded in social abstractions and a decentralized conception to support autonomous operation. Agents enact a specified social protocol on behalf of their principals, while the social middleware maintains the social state.

The major research challenges are the following. (1) Modeling protocols and ensuring their enactability and viability. (2) Developing middleware to compute and progress the social state. (3) Specifying a social protocol (collectively by principals). Generalizing social computing means to have models of problems vis a vis architectures of solutions. Concrete questions are the following. *Who may initiate a computation? Who selects the participants? Are the parties interested in the outcome? Do the parties interact repeatedly? Do the parties learn and might useful outcomes emerge? Is it a majority or a minority game? How do participants interact?*

The conclusions on achieving the promise of social computing are the following. Computer science is a game of abstractions and for social computing we need new abstractions. In particular, human-level abstractions are to be incorporated, problems are to be elicited more precisely, more flexibility to participants is to be granted, a clearer accountability of actions is to be obtained, and work is to be held to higher standards of norms and ethics. The discussion then was about the following questions. *What is the relation to software development? May the agent be abstracted by API? What is the social middleware in case of crowdsourcing? Is it Enterprise Crowdsourcing?*

Martha A. Larson: *Crowdsourcing Ethics: Towards operationalizing ethics*

Ethical issues in the area of crowdsourcing were then raised in the second talk of the late night discussion session. First the concept of ethics was elaborated. It involves systematizing, defending, recommending concepts of right and wrong behavior. The normative principles in applied ethics considers (i) personal benefit, i.e. acknowledge the extent to which an action produces beneficial consequences for the invidiual in question, (ii) social benefit, (iii) principle of honesty, lawfulness, autonomy (acknowledge a person's freedom over his/her actions or physical body), justice, (iv) rights. The question arises why we should care about ethics in the context of crowdsourcing. We are actors in crowdsourcing systems to directly benefit from an environment that respects ethics. In general, long run altruism beats greed. An ethical system is more sustainable (everyone is empowered to reach full capacity).

The next question is when we should care about ethics. The answer is simple – always. In cases where crowdwork income is not essential: *What are people not doing when they are crowdsourcing? Does crowdsourcing have an addictive side?* In cases where people are subsisting on crowdwork income: *Is the crowdwork paying only money, without building a longer term future? Is the money enough? Is the timing of the payment ok?*

There are easy ways for taskmakers to take ethics into consideration. (1) Try some crowdwork. (2) Do not reveal identity of the crowdworkers. (3) Address crowdworkers politely (and try to apologize if you goof up). (4) Don't engage in the "race of bottom". (5) Review work promptly. (6) Realize there is an investment of time and not only money. (7) Respond to inquiries (even the ones with horrible grammar). (8) Use contracts to clearly define your relationships. (9) Be open and honest.

Nevertheless, there are challenging ways towards ethics. There is more than one system involved in determining what is ethical and non-ethical behavior on the platform. Further it should be ensured to give people the information that they need to make informed decisions about the use of their work and their information. Understanding the past will also be an important issue. Privacy is crucial and health state and personality information need to be protected like it was priceless art. More challenging ways towards ethics address the following. (A) Think beyond instant gratification. (B) Spend lots of time following guidelines for human subject research. (C) Spend lots of time doing non-science activities like joint task forces to improve and innovate crowdsourcing practices. (D) Be close to workers. (E) Mirror the ethics mechanisms of the real world on the crowdsourcing platform. (F) Engage in ethics by design. A crowdwork unit should be associated with certification information (e.g. fair trade). Crowdworkers can feedback information and self organize if they can communicate among themselves. Every party on a crowdsourcing platform should reach all other parties.

Some interesting notes taken from the discussion: "CS workers are not anonymous, they have skills and motivations." "Crowdsourcing reduces humans to mechanical devices." "In cases where crowdwork income is not essential: What are people not doing when they are crowdsourcing? Does crowdsourcing have an addictive side? In cases where people are subsisting on crowdwork income: Is the crowdwork paying only money, without building a longer term future? Is the money enough?" "There are easy ways for task makers to take ethics into consideration: do not reveal identity of the crowdworkers; address crowdworkers politely; review work promptly; realize there is an investment of time and not only money; respond the inquiries; use contracts to clearly define your relationships; mirror the ethics mechanisms of the real world on the crowdsourcing platform."

Participants of the Seminar

In total, 22 researchers (18 male, 4 female) from 9 different countries and 16 different institutions or companies participated in the seminar. The majority of participants is working in academia (86 %) in Europe (77 %). But it has to be highlighted that especially the participants from industry were enriching the seminar and brought in important and complementary viewpoints. Similarly, the young researchers (36 %) actively brought in their opinions and enriched the discussions especially in the working groups. The complete list of participants can be found below.

Participants

- Abraham Bernstein
Universität Zürich, CH
- Kathrin Borchert
Universität Würzburg, DE
- Alessandro Bozzon
TU Delft, NL
- Cristina Cabanillas
Wirtschaftsuniversität Wien, AT
- Joseph Davis
The University of Sydney, AU
- Gianluca Demartini
University of Fribourg, CH
- Klaus Diepold
TU München, DE
- Matthias Hirth
Universität Würzburg, DE
- Tobias Hoßfeld
Universität Würzburg, DE
- Andreas Hotho
Universität Würzburg, DE
- Deniz Iren
Middle East Technical University – Ankara, TR
- Christian Keimel
TU München, DE
- Shinichi Konomi
University of Tokyo, JP
- Vassilis Kostakos
University of Oulu, FI
- Markus Krause
Universität Hannover, DE
- Martha A. Larson
TU Delft, NL
- Babak Naderi
TU Berlin, DE
- Nhatvi Nguyen
Weblabcenter, Inc. – Texas, US
- Munindar P. Singh
North Carolina State Univ., US
- Phuoc Tran-Gia
Universität Würzburg, DE
- Maja Vukovic
IBM TJ Watson Res. Center, US
- Florian Zeiger
AGT International – Darmstadt, DE

Report from Dagstuhl Seminar 13362

Cloud-based Software Crowdsourcing

Edited by

Michael N. Huhns[1], Wei Li[2], and Wei-Tek Tsai[3]

1 **University of South Carolina, US,** huhns@sc.edu
2 **Beihang University – Beijing, CN,** liwei@nlsde.buaa.edu.cn
3 **ASU – Tempe, US,** wtsai@asu.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13362 "Cloud-based Software Crowdsourcing".

In addition to providing enormous resources and utility-based computing, clouds also enable a new software development methodology by crowdsourcing, where participants either collaborate or compete with each other to develop software. Seminar topics included crowd platforms, modeling, social issues, development processes, and verification.

Seminar 01.–04. September, 2013 – www.dagstuhl.de/13362
1998 ACM Subject Classification K.6.3 Software Management
Keywords and phrases Crowdsourcing, Software Development, Cloud Computing
Digital Object Identifier 10.4230/DagRep.3.9.34
Edited in cooperation with Wenjun Wu

1 Executive Summary

Michael N. Huhns
Wei-Tek Tsai
Wenjun Wu

Crowdsourcing software development or software crowdsourcing is an emerging software engineering approach. Software development has been outsourced for a long time, but the use of Internet with a cloud to outsource software development to the crowd is new. Most if not all software development tasks can be crowdsourced including requirements, design, coding, testing, evolution, and documentation. Software crowdsourcing practices blur the distinction between end users and developers, and allow the co-creation principle, i.e., a regular end-user becomes a co-designer, co-developer, and co-maintainer. This is a paradigm shift from conventional industrial software development to a crowdsourcing-based peer-production software development. This seminar focused on the notion of cloud-based software crowdsourcing, with the following goals:

1. to establish a theoretical framework for applying software crowdsourcing, and identify the important design patterns and highly interactive and iterative processes in a cloud-based infrastructure.
2. to propose and design a reference architecture for software crowdsourcing
3. to develop and finalize the research roadmap for software crowdsourcing for the next five years

The grand research challenge in cloud-based software crowdsourcing is how to embrace elements from the two aspects: cloud infrastructure and software crowsourcing. Metaphorically, it can be regarded as synergy between two clouds – machine cloud and human cloud, towards the ultimate goal of developing high-quality and low cost software products. This seminar intended to bring together scientists from both fields to tackle the major research problems in this emerging research area.

More than twenty researchers, who work on different domains such as crowsourcing, human-computer interaction, cloud computing, service oriented computing, software engineering and business management attended the seminar. In addition to regular 5-minute talks from every participant in the seminar, the organizer arranged a keynote speech delivered by Prof Schahram Dustdar, which summarizes large-scale collective problems solving research enabling software crowsourcing. The topics covered by their presentations can be roughly categorized into three groups: software crowdsourcing process and models, crowdsourcing cloud infrastructure and human crowd management. To promote in-depth discussion among these topics, we also divided people into five discussion groups including:

Crowd Source Software Engineering Design-Group: This group identified the three main areas in the design of software crowdsourcing: processes, models, and techniques. It highlighted the importance of standardized generic models of software crowdsourcing study, and explored multiple crowdsurcing techniques, especially virtual team formation and quality assessment.

Worker-centric design for software crowdsourcing: This group focused on the crowd management in software crowdsourcing and aimed to answering the question about how to make a sustainable software crowdsourcing industry. Discussion in the group covered the major issues such as careers and reputation development of workers, trust among workers and "employers" (task solicitors) on crowdsourcing markets, virtual team selection and team building.

Cloud-based Software Crowdsourcig Architecture: This group discussed the possible common architectures of crowd-sourcing applications and explored two complementary architectural approaches.

Experimentation Design for Software Crowdsourcing: The central topic of this group is about how to design a valid and reproducible experiment for software crowdsourcing research. The group had extensive discussion on software crowdsourcing experiment approaches and the major crowdsourcing infrastructures.

Infrastructure and Platform: This group reviewed the motivations to construct the crowdsourcing platform, analyzed architecture design issues, and proposed a educational platform for software crowdsourcing.

During the session of our seminar, Dagstuhl also set up a parallel seminar named "Crowdsourcing: From Theory to Practice and Long-Term Perspectives", which mostly focused on general crowdsourcing research and service platforms. Software crowdsourcing can be regarded as one of the most complex crowdsourcing activities that often need intense dedication from workers with high-level skills of software engineering. Thus, there are some interesting overlapping areas such as worker incentive and quality assurance, between our seminar and the parallel seminar. To foster collaboration among the two groups, we hold a joint discussion session for introducing and sharing findings from each group, followed by an evening session with two presentations from the general crowdsourcing group.

We believe this seminar is a good start for software crowdsourcing research. Finding and consensus generated from the seminar have been formalized in the wiki page of software crowdsourcing (http://en.wikipedia.org/wiki/Crowdsourcing_software_development) to give a clear definition and initial reference architecture of cloud-based crowdsourcing software development. More efforts will be put into the growth of the research community and production of joint research publications.

2 Table of Contents

3 Overview of Talks

3.1 Large-Scale Performance Testing by Cloud and Crowd

Xiaoying Bai (Tsinghua University – Beijing, CN)

Joint work of Bai, Xiaoying; Tsai, Wei-Tek; Wu, Wenjun

Software scale and complexity increase tremendously in recent years. Performance testing of Internet scale software systems is usually expensive and difficult. The challenges include: (1) to simulate diversified usage scenarios; (2) to generate various workload distributions; and (3) to measure and evaluate performance from different aspects. Cloud computing and Crowdsourcing are two emerging techniques in recent years. They promote new testing architectures that are promising to address the challenges. Cloud-based testing, such as TaaS (Testing-as-a- Service), aims to support on-demand testing resources and services in/on/over clouds for testers at any time and all time. Testing by crowdsourcing, such as uTest and Mob4hire, follows the Web 2.0 principle of harnessing collective intelligence and various testing tasks can be crowdsourced including test case design, script development, script debugging, test execution, and test result evaluation.

This paper first investigates the framework to facilitate large-scale performance testing by integrating cloud infrastructure and crowd wisdom. A Cloud-based testing platform, Vee@Cloud was built to provide a cross- cloud virtual test lab to support on-demand test scripts provisioning and resource allocation, scalable workload simulation, and continuous performance monitoring. Following the crowdsourcing approach, participants can join the platform for different purposes, such as biding testing tasks, contributing test cases and scripts, renting test resources, and carrying on test executions. The process of establishing test collaboration can be formulated as a multi-criteria decision making problem, using qualitative evaluation models for different factors like task allocation and scheduling, quality control, and cost control.

3.2 Finding Experts

Xavier Blanc (Univ. Bordeaux, LaBRI – Talence, FR)

Heavy usage of third-party libraries is almost mandatory in modern software systems. The knowledge of these libraries is generally scattered across the development team. When a development or a maintenance task involving specific libraries arises, finding the relevant experts would simplify its completion. However there is no automatic approach to identify these experts. In this talk we propose LIBTIC, a search engine of library experts automatically populated by mining software repositories. We show that LIBTIC finds relevant experts of common Java libraries among the GitHub developers. We also illustrate its usefulness through a case study on the Apache HBase project where several maintenance and development use-cases are carried out.

3.3 Crowdsourcing Cloud Infrastructure using Social Networks

Kyle Chard (University of Chicago, US))

The increasing pervasiveness of online social networks is not only changing the way that people communicate and interact but it is also allowing us to represent, document and explore inter-personal relationships digitally. Social networking platforms have gone beyond a platform for communication and are now a viable platform in their own right on which to implement unique socially oriented services with access to an increasingly complex social graph modeling every aspect of an individual life. Building upon this social fabric services can leverage real world relationships, inferring a level of trust between users, exploiting intrinsic social motivations, and developing socially aware algorithms. This talk describes the experiences and lessons learned developing a Social Cloud, a platform that enables the construction of a crowdsourced cloud infrastructure to facilitate resource and capability sharing within a social network. Social Clouds are motivated by the need of individuals or groups for specific resources or capabilities that can be made available by connected peers. Such resources are not necessarily only computational or software resources, but can be any electronically consumable service, including human skills and capabilities. Social Clouds leverage lessons learned through volunteer computing and crowdsourcing such as the willingness of individuals to make their resources available and offer their expertise altruistically for good causes. This talk explores aspects such as inter-personal trust, platform implementation, social incentives and use cases, by leveraging methodologies from computer science, economics and sociology. It looks specifically at these aspects in the context of social network based crowdsourcing and attempts to generalize the approaches used in Social Clouds to other forms of software crowdsourcing.

3.4 Cloud based Crowdsourcing Software Development – Keynote

Schahram Dustdar (TU Wien, AT)

In this talk I begin with analyzing the historical roots of large-scale collective problems solving research in Computer Science. this is followed by a detailed analysis of the mechanisms, algorithms, models, and software architectures and deployment models of software solutions helping to enable crowdsourcing software development on a large scale. This is discussed in relationship with an analysis of currently missing aspects of solutions aiming at supporting cloud based crowdsourcing for software development.

3.5 Hyperscale Development of Software

Michael N. Huhns (University of South Carolina, USA)

The benefits of the open-source (Bazaar? approach to software development have not been fully realized, because the number of software developers is still relatively small and orders of magnitude smaller than the number of users. Developers typically are experts in computing, whereas users typically have domain expertise: this produces a disparity in viewpoints, causing a mismatch between the developed software and its desired use. Moreover, the proposition, given enough eyeballs, all bugs are shallow,would take on much greater significance, if a larger fraction of the users could also be developers. A solution to this problem has so far been impractical: end-users often do not have sufficient expertise to contribute software, nor the time to learn how to do so, and there was no way to meld it with existing software until it was proven correct or from a trusted developer.

In this paper I describe an approach for broadening dramatically the number of people who contribute to the development of software. If successful, the approach will result in a more effective software-development process, greatly improved software reliability, and increased end-user satisfaction. The approach is ambitious – in that it affects all stages of software development and several levels of the software execution process and transformative – in that the software will be developed and executed in new ways.

The approach makes use of take advantage of-all contributed code, components, and designs until they have proven to be of no value. The talk will describe how agent-based wrappers can manage the necessary collaboration and competition, allowing the contributions to be used alongside their existing counterparts until their behavior and features can be assessed. The solution exploits concepts from N-version programming, multicore processors, model- driven architectures, test-driven development, autonomic computing, negotiation in multiagent systems, group decision-making, and consensus.

The two main threads of the proposed research are broader participation in developing software and multiagent systems for the use and execution of the software. For each we are developing prototypes and realistic evaluations demonstrating greater robustness and usefulness. The research can help solve the problem of incorrect software by improving robustness, while enabling a wider cross-section of society to develop and personalize software. This can lead to greater understanding, satisfaction, and utility of software that behaves as people generally want it to.

3.6 "Microtask" vs. freelancer platforms – how crowdsourcing can complement software development

Robert Kern (IBM Deutschland – Böblingen, DE)

In the era of cloud computing, mobile computing, collaboration and big data, software development requirements are significantly changing. Users and organizations are asking for shorter development cycles, improved ease of use, better integration and lower administration and operation overhead. This results in a need for flexible software development and operation processes combined with a new assignment of roles in order to hide manual efforts from the user.

Crowdsourcing has the potential to address these challenges in several ways. On the one hand, platforms like TopCoder or oDesk enable for flexible outsourcing of design and implementation efforts to freelancers. On the other hand, "microtask" platforms like Crowdflower or Amazon MTurk provide human workforce as a scalable service in order to fulfill formalized tasks which are difficult to automate. Such services can either be integrated into SW services to deliver complex services like search engine optimization, data cleansing and social media analysis, or they can complement SW operation and development, e.g. by providing testing services.

After contrasting the two types of platforms, this presentation focuses on the latter type. By elaborating on the analogy of cloud computing and crowdsourcing, a "cloud labor" stack is introduced that seamlessly integrates with the concepts of cloud computing. The capabilities provided by the different layers of the stack are identified and implementation options are outlined. The considerations are then validated by discussing a crowd-based medical coding service offered by IBM. Finally, a bow is drawn back to the freelancer platforms and initial ideas are provided on how the two complementary concepts could be merged in the future.

3.7 Software Development Crowdsourcing Issues: Organization Design and Incentive Design

Donghui Lin (Kyoto University, JP)

In the talk, we discuss two important issues in Software Development Crowdsourcing: Organization Design and Incentive Design. We share the experience of problem-based learning course of crowdsourcing in university.

3.8 Crowdsourcing with Expertise

Greg Little (ODesk Corp. – Redwood City, US)

Crowdsourcing often focusses on small tasks that require no special skills other than being human. I'm interested in tasks and workflows that do require special skills, like programming or art skills. For instance, one could imagine a crowdsourced logo that involves several phases. The first phase might pay 20 sketch artists to sketch logo ideas, and then pay a traditional crowd to vote for the best 5. The next phase might then pay 5 designers to flesh out the best 5 logos from the previous phase, followed by another round of voting. The final phase might involve hiring an expert to finalize the top design.

Crowdsourcing with expertise is difficult today because it is difficult to programmatically hire many experts – hiring experts typically involves looking at resumes and portfolios. My research has focussed on finding scalable ways to reliably identify experts. Two promising ideas include: (1) having people play simple subjective games based on their expertise, and rating each others results; (2) having public work histories which include both the task and the results, as opposed to portfolios today which typically only display the results, but it is hard to know what this worker contributed, or whether their result met their client's requirements.

3.9 Multi-Agent System Models and Approach of Crowdsourcing Software Development

Xinjun Mao (National University of Defense Technology – Changsha, China)

The advent and successful practices of software crowdsourcing needs to investigate its in-depth essence and seek effective technologies to support its activities and satisfy increasing requirements. We highlight crowdsourcing participants consist of a multi-agent system and software crowdsourcing is a multi-agent problem-solving process. This paper discusses the characteristics and potential challenges of software crowdsourcing in contrast to traditional software development, and present a general analysis framework based on multi-agent system to examine the organization and behaviors of software crowdsourcing. Several software crowdsourcing models performed on typical platforms like Topcode, uTest are established and their organization and coordination are discussed. We have developed a service-based multi-agent system platform called AutoService that provides some fundamental capabilities like autonomy, monitoring, flexible interaction and organization, and can serve as an infrastructure to support software crowdsourcing models and tackle its challenges. A software crowdsourcing prototype is developed and some scenarios are exemplified to illustrate our approach.

3.10 Crowds, Clouds, Agents and Coordination

Dave Murray-Rust (University of Edinburgh, GB)

Joint work of Murray-Rust, Dave; Robertson, Dave

The term "social machines" describes a class of systems where humans and machines interact, and the mechanical infrastructure supports human creativity. As well as software crowdsourcing projects such as TopCoder and oDesk, this includes distributed development platforms such at GitHub and Bitbucket. In this paper, we describe a formalism for social machines, consisting of i) a community of humans and their "social software" interacting with ii) a collection of computational resources and their associated state, protocols and ability to analyse data and make inferences. These social machines are increasingly the target of software development, and as such, they represent an interesting problem, as the community must be "programmed" as well as the machines. This leads to evolving and unknown requirements, and having to deal with much softer concepts than formal systems designers usually work with. Our model hence uses two coupled social machines. There is the target machine, and the machine which is used to create it—much as GitHub and associated resources might be used to form a social machine to create "the next Facebook". We draw on the ideas of 'desire lines' and 'play-in' to argue that top down design of social machines is impossible, that we hence need to leverage computational support in creating complex systems in an iterative, dynamic and emergent manner, and that our formalism provides a possible blueprint for how to do this.

3.11 Collaborative Majority Vote: Improving Result Quality in Crowdsourcing Marketplaces

Khrystyna Nordheimer (Universität Mannheim, DE)

Joint work of Nordheimer, Dennis; Nordheimer, Khrystyna; Schader, Martin; Korthaus, Axel

Crowdsourcing markets, such as Amazon Mechanical Turk, are designed for easy distribution of micro-tasks to an on-demand scalable workforce. Improving the quality of the submitted results is still one of the main challenges for quality control management in these markets. Although beneficial effects of synchronous collaboration on the quality of work are well-established in other domains, interaction and collaboration mechanisms are not yet supported by most crowdsourcing platforms, and thus, not considered as a means of ensuring high-quality processing of tasks. In this paper, we address this challenge and present a new method that extends majority vote, one of the most widely used quality assurance mechanisms, enabling workers to interact and communicate during task execution. We illustrate how to apply this method to the basic scenarios of task execution and present the enabling technology for the proposed real-time collaborative extension. We summarize its positive impacts on the quality of results and discuss its limitations.

3.12 The Open Source Volunteering Process

Dirk Riehle (Universität Erlangen-Nürnberg, DE)

Today's software systems build on open source software. Thus, we need to understand how to successfully create, nurture, and mature the software development communities of these open source projects. In this article, we review and discuss best practices of the open source volunteering and recruitment process that successful project leaders are using to lead their projects to success. We combine the perspective of the volunteer, looking at a project, with the perspective of a project leader, looking to find additional volunteers for the project. We identify a five-stage process consisting of a connecting, understanding, engaging, performing, and leading stage. The underlying best practices, when applied, significantly increase the chances for a successful open source project.

3.13 Artifact-centric Incentive Mechanisms for Socio-technical Systems

Ognjen Scekic (TU Wien, AT)

Joint work of Scekic, Ognjen; Truong, Hong-Linh; Dustdar, Schahram
Main reference O. Scekic, H.-L. Truong, S. Dustdar, "Incentives and rewarding in social computing," Communications of the ACM, 56(6):72–82, 2013.
URL http://dx.doi.org/10.1145/2461256.2461275

Crowdsourcing systems of the future (e.g., social compute Units – SCUs, collective adaptive systems) promise to support processing of richer and more complex tasks, such as software development. This presupposes deploying ad-hoc assembled teams of human and machine services that actively collaborate and communicate among each other, exchanging different artifacts and jointly processing them. Major challenges in such environments include team formation and adaptation, task splitting and aggregation, and runtime management of data flow and dependencies, collaboration patterns and coordination mechanisms. These challenges can be somewhat alleviated by delegating the responsibility and the know-how needed for these duties to the participating crowd members, while indirectly controlling and stimulating them through appropriate incentive mechanisms.

In this paper we present a novel, artifact-centric approach for modeling and deploying incentives in rich crowdsourcing environments. Artifact's lifecycle model is augmented with incentive mechanisms to create encapsulated units that can be offered to the crowd for processing. The incentive mechanisms are adaptive and constantly advertised to the crowd to drive the processing in the envisioned direction and tackle the aforementioned challenges. They are designed to promote teamwork, and to support time and data dependencies.

3.14 Engineering Multi-Cloud Service-Oriented Applications

Lionel Seinturier (Lille I University, FR)

Joint work of Paraiso, Fawaz; Merle, Philippe; Seinturier, Lionel

Cloud platforms are increasingly being used for hosting a broad diversity of services from traditional e-commerce applications to interactive web-based IDEs and crowdsourcing systems. However, the proliferation of offers by cloud providers raises several challenges. Developers will not only have to deploy applications for a specific cloud, but will also have to consider migrating services from one cloud to another, and to manage distributed applications spanning multiple clouds. In order to address these challenges, we present a federated multi-cloud PaaS infrastructure that is based on three foundations: i) an open service model used to design and implement both our multi-cloud PaaS and the SaaS applications running on top of it, ii) a configurable architecture of the federated PaaS, and iii) some infrastructure services for managing both our multi-cloud PaaS and the SaaS applications. We report on the deployment of this cloud-based infrastructure on top of 10 existing IaaS/PaaS.

3.15 On Assuring Quality of Results in Hybrid Compute Units in the Cloud

Hong-Linh Truong (TU Wien, AT)

Joint work of Truong, Hong-Linh; Dustdar, Schahram

Recently there is an increasing trend of utilizing hybrid compute units consisting of software-based and human-based services to solve complex problems. When both software-based and human-based services are provisioned based on pay-per-use, it is a great challenge to assure the quality of results, which center on the cost, response time and the quality of data. We discuss a range of techniques, from composition to monitoring phases, to support the assurance of quality of results in cloud-based crowdsourcing software development.

3.16 Software Crowdsourcing Maturity Models

Wei-Tek Tsai (ASU, Tempe, AZ , US)

Some speculated that crowdsourcing can be used for ultra large systems. But examinations of existing crowdsourcing websites such as TopCoders, uTest, and AppStori indicates that it may take a long time to reach that levels of maturity. What will be the roadmap for crowdsourcing growth? What will be the platform architecture to support future crowdsourcing? How about conventional issues such as testing (including regression), configuration, scalability, software architecture, fault-tolerant issues, concurrent software development processes?

Here we define a Crowdsourcing Maturity Model:

Level 1: Single persons, well-defined modules, small size, limited time span (less than few months), quality products, current development processes such as the one by AppStori, TopCoder, and uTest. At this level, coders are ranked, websites contains online repository crowdsourcing materials, software can be ranked by participants, crowdsourcing platforms have communication tools such as wiki, blogs, comments as well as software development tools such as IDE, testing, compilers, simulation, modeling, and program analysis.

Level 2: Teams of people (< 10), well-defined systems, medium size, medium time span (multiple months to less than one year), and adaptive development processes with intelligent feedback in a common cloud platform where people can freely share thoughts. At this level, a crowdsourcing platform supports adaptive development process that allow concurrent development processes with feedback from fellow participants; intelligent analysis of coders, software products, and comments; multi-phase software testing and evaluation; Big Data analytics, automated wrapping software services into SaaS (Software-as-a-Service), annotate with ontology, cross reference to DBpedia, and Wikipedia; automated analysis and classification of software services; ontology annotation and reasoning such as linking those service with compatible input/output.

Level 3: Teams of people (< 100 and > 10), well-defined system, large systems, long time span (< 2 years), automated cross verification and cross comparison among contributions. A crowdsourcing platform at this level contains automated matching of requirements to existing components including matching of specification, services, and tests; automated regression testing.

Level 4: Multinational collaboration of large and adaptive systems. A crowdsourcing platform at this level may contain domain-oriented crowdsourcing with ontology, reasoning, and annotation; automated cross verification and test generation processes; automated configuration of crowdsourcing platform; and may restructure the platform as SaaS with tenant customization.

3.17 Trustie: a Platform for Software Development Ecosystem incorporating Engineering Methods and Crowd Wisdom

Huaimin Wang (NUDT – Hunan, CN)

Software production activity includes the process of software creation, which relies on developers' inspiration and talent, and the process of software manufacture, which is executed under strict engineering code. Traditional Software Engineering stresses software manufacture, while the Open Source Software development focuses on software creation. These two software production models each has its strengths and weaknesses, but are complementary to each other. Literature research and engineering practices have shown that traditional software engineering approaches have encountered obstacles in several creation activities such as requirement elicitation. By exploiting the crowd wisdom, open source development provides a better environment for software creation. However, it can neither make promise on specific software requirements, nor guarantee the progress and quality of production. In this paper, we suggest to introduce a software development ecosystem which combines the strengths of these two models. First, we propose the general service model of a platform which can support such a dual function development ecosystem. Architecturally, the core of this service model contains a novel software process model and software evidence model. Besides, it integrates collaborative development, resource sharing and analysis into a unified framework. Based on this service model, we designed Trustie: a software development platform, under the support of a Chinese national 863 grand project. Trustie is equipped with built-in collaboration tools and an open and evolving software repository with deep analysis utilities. Trustie bridges software manufacture activities and software creation activities in three featured aspects:
1. It enables crowd-oriented collaboration among developers and stakeholders.
2. It supports massive software resource and knowledge sharing.
3. It provides historical software data analysis and software quality evaluation.

Up till now, Trustie has already been successfully adopted by a series of on-line projects and large-scale industry applications.

3.18 Crowdsourcing for Software Ecosystem

Wenjun Wu (Beihang University – Beijing, CN)

Software development is complex and creative as it involves requirement analysis, design, architecture, coding, testing and evaluation. Recently, software crowdsourcing, or outsource software development to the crowd, has been popular with numerous software coders participated in various software competitions. We first analyze the data collected on popular software crowdsourcing and summarizes major lessons learned. We then examine two popular software crowdsourcing processes including TopCoder and AppStori processes. Specifically, this paper evaluates competition rules used in these processes, and compare with a traditional software development process IBM Cleanroom methodology as it claims of delivering zero-defect software. We identify an important design element in software crowdsourcing for software quality and creativity is the min-max nature among participants. The min-max nature comes from game theory where participants compete to win the game with one party tires to minimize an objective or the other party tries to maximize the same object. However, in software crowdsourcing, the min-max is done by one party tries to maximize the finding of bugs in a set of artifacts, and the other parties try to minimize the potential bugs in the same artifact. The min-max can be practiced without being a zero-sum competition where a gain from one party will be matched by another party. In other words, software crowdsourcing can be a win-win for all parties and they can collaborate while practicing a min-max game. By using this approach, lots of aspects of software development can be crowdsourced from initial project concepts, to specification, design, algorithms, coding, and testing, with the crowd can contribute their creativity to each aspect.

3.19 An Evolutionary and Automated Virtual Team Making Approach for Crowdsourcing Platforms

Tao Yue (Simula Research Laboratory – Lysaker, NO)

Crowdsourcing has demonstrated its capability of supporting various software development activities including development and testing as it can be seen by several successful crowdsourcing platforms such as TopCoder and uTest. However, to crowd source large-scale and complex software development and testing tasks, there are several optimization challenges to be addressed such as division of tasks, searching a set of registrants, and assignment of tasks to registrants. Since in crowdsourcing a task can be assigned to registrants geographically distributed with various background, the quality of final task deliverables is a key issue. As the first step to improve the quality, we propose a systematic and automated approach to optimize the assignment of registrants in a crowdsourcing platform to a crowdsourcing task. The objective is to find the best fit of a group of registrants to the defined task. A few examples of factors forming the optimization problem include budget defined by the task submitter and pay expectation from a registrant, skills required by a task, skills of a registrant, task delivering deadline, and availability of a registrant.

4 Working Groups

4.1 Cloud Infrastructure for Software Crowdsourcing

Wei-Tek Tsai

The group discussed about this topic from two point of views:
1. Understand the current cloud infrastructures such as IaaS (infrastructure-as-a-service), PaaS (platform-as-a-service), and SaaS (software-as-a-service), and how these infrastructures can support software crowdsourcing.
2. Understanding the needs of software crowdsourcing including its processes, organization structure, collaboration patterns, user support, user interface, and payment features. Cloud Features to Support Software Crowdsourcing

In general, the group felt that most of cloud infrastructure including all the related features such as metadata-based design, scalability architecture, multi-tenancy architecture, automated migration, automated redundancy management, automated recovery, new database design, runtime system composition, execution, and monitoring are useful to support software crowdsourcing. These features may be made as menu selection for software crowdsourcing organizers. For example, an organizer may like triple redundancy for collaboration-based crowdsourcing, but additional redundancy for competition-based crowdsourcing. The additional redundancy is needed to ensure that competition data will not be lost.

Software Crowdsourcing Process and Organization Needs

In general, the group felt that while different software crowdsourcing processes may have different needs, but among the diversity of software crowdsourcing processes, they actually share much commonality. Common themes for software crowdsourcing processes include:

1. Collaboration and communication tools such as a distributed blackboard system where each party can participate in discussions,
2. Participant ranking and recommendation tools;
3. Software development tools such as modeling tools, simulation tools, programming language tools such as compilers and intelligent editors, design notations, and testing tools;
4. Cloud payment and credit management tools;
5. Cloud service management dashboard for system administrators and software crowdsourcing organizers;
6. Board where user can register and upload their profiles information; and
7. Repository of software development assets such as modules, specifications, architecture and design patterns.

Major variations will come from different software crowdsourcing processes, for example, competition-based processes such as those done by TopCoder will be different from collaboration-based processes, such as by AppStori. Competitions require enforcement of game rules including time management and rigorous evaluation, collaborations require communication, publishing, sharing, alerts, automated text and index processing.

The group came out with a draft reference architecture as in Fig. 1.

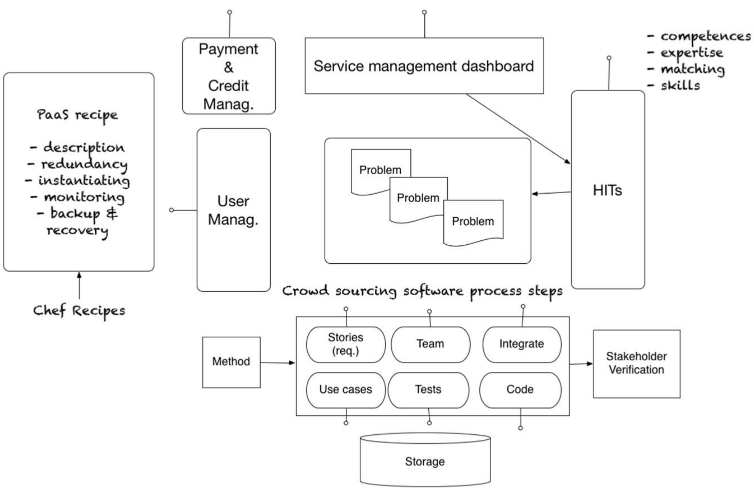


Figure 1 Reference architecture of cloud-based software crowdsourcing.

4.2 Crowd Source Software Engineering Design

Shaukat Ali

Joint work of Shaukat Ali, Michael Huhns, Khrystyna Nordheimer, Yue Tao, Hong-Linh Truong, Wenjun Wu

The discussion in this group was focused on Crowd Source Software Engineering design in the three main areas, processes, models, and techniques.

4.2.1 Processes

Software engineering development projects based on crowd sourcing can be developed using any software development process models such as Water Fall and Agile development. Such information may be requested by a project submitted or alternatively can be suggested by a cloud platform. Which approach is better is still an open issue and requires further investigation.

4.2.2 Models

There is a need for standardized generic models (ontologies/domain specific language) for capturing information about tasks (such as task description, micro-tasks and atomic tasks), individual in crowds (such as expertise, payment history and successful project history) and project information (such as cost, time, and required expertise). There are several benefits to create such models:

1. A standardized way of capturing various concepts in crowd sourcing and their relationships to promote a common and unified understanding in the community. These can further be pushed for standardization in the future.
2. A systematic way of specializing the models for specific applications of software development such as implementation and testing.
3. Promote automation of various activities in crowd sourcing such as team formation and ranking.

An important issue is how to divide a project into a set of atomic tasks that can be performed by different individuals in the crowd independently. The following potential solutions were discussed:

1. A project submitted by a project owner may be divided into atomic tasks by the project owner. However, this may not always be optimal if not impossible. Such process may be improved by restricting the project owner to submit the project in a specific format that can facilitate division of the project into atomic tasks. However, this raises several potential open issues such as which kind of restricted format can be used and if there are several alternatives, which one is the best one for what kind of tasks.
2. An semi/automated way of introducing a platform independent broker, which could somehow divides the project into independent tasks, which can then be sent to crowd sourcing platforms. Such broker may not be visible to the project submitted.
3. A common practice in the current crowd sourcing platforms is the use of man-in-the-loop approach, where a project moderator coordinates with the bidders and help dividing the project into different atomic tasks.

4.2.3 Techniques

Crowd sourcing activities may be implemented by various different techniques. In the discussion, first we talked about various techniques for ranking individuals in a crowd. Several machine learning techniques have been applied by various researchers to rank individuals. Secondly, we talked about team formation techniques. Similar to ranking techniques, different approaches such as based on search algorithms and machine learning have been applied for team formation. A fundamental question to answer for both of the above activities is which approaches are cost/effective in which situations. Doing so requires thorough empirical evaluations based on focused case studies to study strengths and weakness of the proposed approaches.

4.2.4 Other Open Issues

In addition, we also discussed some of the open issues related to the quality of crowd sourcing from various perspectives:
- How to assess the quality of a crowd?
- How to assess the quality of an individual in a crowd?
- How to assess the quality of crowd software?
- How to assess the quality of crowd sourcing infrastructure?

4.3 Worker-centric design for software crowdsourcing

Dave Murray-Rust (University of Edinburgh, GB)

Joint work of Murray-Rust, Dave; Yin, Gang; Scekic, Ognjen; Wang, Huaimin; Lin, Donghui; Mao, Xinjun; Kern, Robert; Little, Greg; Riehle, Dirk

Crowdsourcing is emerging as a compelling technique for the cost-effective creation of software, with tools such as ODesk and TopCoder supporting large scale distributed development. From the point of view of the commissioners of software, there are many advantages to crowdsourcing work – as well as cost, it can be a more scalable process, as there is the possibility of selecting from a large pool of expertise. From the point of view of workers, there is a different set of benefits, including choice of when and how to work, providing a means to build a portfolio, and a lower level of commitment to any particular employer.

Most analyses of crowdsourcing take the point of view of the commissioners of work: how is it possible to get work done better, more cheaply, more robustly etc. When considered as an "outsider" technology, this need to prove value to commissioners of work is completely understandable. Crowdsourcing is no longer a niche activity, however. From 2000–2009, cloudworkers had been paid up to \$2Bn; the number of participants has grown by over 100% per year, and there are now over 6 million cloudworkers worldwide.

In this group, we engaged with the question of what it would take to make software crowdsourcing a sustainable industry. This means being able to attract intelligent, motivated individuals, who can make enough money to satisfy themselves. Essentially, we asked the question "What would we want from a crowdsourcing marketplace if we were going to work in it".

Software crowdsourcing is markedly different to Turking and other crowdlabour projects, for a number of reasons. If crowdlabour is divided into micro tasks, macro tasks, small projects and complex projects, then while much of the crowdsourcing industry focusses on

Mechanical Turk style microtasks, software creation tends to fall into the small- or complex-project brackets, requiring workers to bring in existing skills, and some degree of coordination or direct worker contact. However, if crowdsourcing were to replace traditional employment for a significant proportion of software developers, the reduced levels of commitment between workers and commissioners could prove problematic for workers over time. In this paper, we identify several areas of interest, and discuss what the issues are, and how current and future solutions could address them: careers and reputation development; trust; team selection and team building; and contextualisation of the work carried out.

4.3.1 Trust and Reputation

Arguably, co-workers are one of the most important factors contributing to a pleasant and productive working environment . In traditional companies workers usually cannot directly select their co-workers. However, since the nature of the employment relationship is a long-lasting one, it gives them time to get to know their colleagues and forge working relationships. The management will actively monitor these relationships in order to achieve a more harmonic, and thus more productive or creative environment.

In crowdsourcing environments, the relationship of workers with the platform and co-workers are irregular and short-lived. This leaves no time to get to know and other workers. Crowdsourced teams are often unique, both time- and composition-wise. Co-workers are often hidden behind digital profiles, creating an atmosphere of distrust and uncomfort. Furthermore, such settings make for an attractive environment for attempting fraudulent activities, such as multitasking, rent-seeking or tragedy of the commons

The proliferation of different metrics and trust models indicate that agreeing on a uniform, context-independent trust and reputation model is practically unfeasible. A new approach and some out-of-the-box thinking will be needed take to address this problem.

4.3.2 Team Selection

In the crowdsourcing environment, large complex projects always need to be conducted by cooperation of a number of crowd workers. As well as the issues of trust and reputation discussed previously, the composition of the team can have an effect on performance, and also the satisfaction of the workers who constitute the teams.

Existing crowdsourcing platforms do not have much support for coordination and interaction among crowd workers. In some platforms like Amazon Mechanical Turk, tasks are separated in an atomic manner so that crowd workers do not need to collaborate with each other. Other platforms support offline collaboration among crowd workers with the guidance of the work requester for complex projects. However, creative work like software development requires a large degree of knowledge integration, coordinated effort and interaction among workers.

The task allocation problem has been discussed for decades in artificial intelligence and distributed computing circles. In crowdsoucing environments, recent researches focus on task decomposition for modeling appropriate workflow with iterative tasks for the purpose of quality assurance. However, task matching for crowd workers is also important for creative complex task in crowdsourcing, where two main factors should be considered: the skills posessed by crowd-workers, and the incentives needed to motivate them.

It will become increasingly necessary to provide mechanisms by which software crowd-workers can collaborate with people they know and trust; where they can organise themselves effectively as situtations and contexts evolve; and where they are able to utilise—and improve—their skils on a variety of non-monotonous tasks.

4.3.3 Contextualisation

The context and purpose of software development can be a large motivating factor for workers; there is a need to reduce the anonymity on both working and comissioning sides, to provide task context; decontextualized tasks remove the ability of workers to understand the moral valence of their labour, and decide whether the task they are carrying out is morally acceptable to them. Examples range from spammers attempting to break Captchas to governments outsourcing recognition of persons of interest in photographs.

When discussing general collective intelligence situations, Malone describes the reward for taking part as being based on "Money, Love or Glory". The complement of this (leaving aside the pecuniary aspects) is that one should be engaged in a task that one does not hate, and is not ashamed of. Additionally, Malone suggests that commissioners of collective intelligence should engage with the design questions: What is being done? Who is doing it? Why are they doing it? How is it being done? These questions can be reversed to create a list of questions which crowdworkers should be able to ask, both for their own peace of mind and as a way for commissioning entities to engage with the Love and Glory motivations:

- "What is the overall project?"
- "Who is comissioning the work?"
- "Why are they comissioning it?"
- "How is it being carried out?"

We have discussed the need for trust and repuation between crowdworkers; there is also the need for accountability for comissioners of work. Requesters on Mechanical Turk currently are not bound by reputation systems, allowing them to act with impunity when designing tasks.

Traditional workers have the benefit of many organisational structures that support them. Labour laws ensure a safe and healthy environment; employers are tasked with managing the physical space that they inhabit in working hours; they will meet other people in they workplace; advocacy groups and unions may exist to represent the needs of workers. For a crowd working career, something providing some of the properties of these structures would need to be created.

4.4 Architecture for Cloud-based Software CrowdSourcing

Michael Maximilien (IBM Almaden Center – San José, USA)

Joint work of Michael Maximilien, Xiaoying Bai, Kyle Chard, Schahram Dustar, Wei-Tei Tsai, Lionel Seinturier

Crowd-sourcing platforms have been mostly proprietary for the most parts. Unlike trendy cloud platforms, there are no dominant platforms for building crowd-sourcing applications. This could be due to the fact that crowd-sourcing applications are varied and have similar needs as those for cloud applications. However, in our discussions and analysis, it seems that common architectures may actually exist for crowd-sourcing applications, and creating a reference OSS version of such platform might be of interest.

With this aim, we have explored two complementary approaches which we discuss here:

4.4.1 Cloud Applications – Specifically Crowd-Sourcing for PaaS

In this approach we are assuming that a Platform-as-a-Service (PaaS) exists, such as the ones provided by Heroku, Microsoft Azure, or the OSS CloudFoundry as well as others. The crowd-sourcing application is simply installed onto this PaaS and tuned to address its needs. We are assuming that the PaaS platform provides primitives to support the needs of the crowd-sourcing application, such as, a flexible database, e.g., MongoDB, an application server, e.g., Ruby on Rails or Node.js, as well as various connections with services that might also be needed, e.g., LDAP, OAuth, Payment, and others. Since the crowd-sourcing application integrates into the PaaS it can be scaled up and down using the PaaS primitives. This usually involves adding more containers for the crowd-sourcing application deployment, e.g., replicating the application server layer and adding a high-availability (HA) proxy.

4.4.2 Distributed Loosely Coupled Crowd-Sourcing Platform

In this approach, the focus is on the components making up the crowd-sourcing platform and pulling them apart and distributing them. That way each component exposes an API(s) and can be combined to create the crowd-sourcing application. Where and how these components are hosted is not taken into consideration. A PaaS could be used or the component could be hosted directly onto an IaaS or bare metal environment. To illustrate the point, think of a simplified crowd-sourcing application needing to host tasks and have a pool of workers. One API would be to retrieve, update, and create tasks. One API would be to retrieve, update, and create new workers. And one more API would be to match workers to tasks and get the workers paid when tasks are completed. In such a distributed service-oriented approach the crowd-sourcing application becomes essentially like a mashup of these primitive crowd-sourcing APIs. Of course, a challenge there is security and scaling since you have no control into the storage, security measures, and scaling capability of the APIs you are using.

4.5 Infrastructure and Platform

Xiaoying Bai (Tsinghua University, CN)

Joint work of Michael N. Huhns, Michael Maximilien, Xavier Blanc, Greg Little, Huaimin Wang, Gang Yin, Kyle Chard, Robert Kern, Lionel Seinturier

Crowdsourcing promotes software development by open collaborations. Cloud is expected to provide necessary services to support development and collaboration activities. To address the needs, the group discussion reviewed the motivations to construct the platform, identified the key components of Cloud infrastructure and platform, analyzed architecture design issues, and proposed an experiment to build a crowdsourcing platform for educating software development.

4.5.1 Motivation Revisited

Crowdsourcing is a promising development method to reduce cost and enhance quality by contracting experts in open communities. However, it is not easy to organize crowdsourced software development. It requires a lot of effort and experiences to decompose software into well-organized modules and development tasks, to find good candidates for each task, to communicate requirements clearly and precisely, and to integrate submitted code into project framework. Hence, crowdsourcing may be not suitable for software development in general, but for specific modules, whose requirements and outcome can be explicitly defined, even quantitatively measured. For example, companies may outsource algorithm design by sponsoring a TopCoder algorithm completion. From another perspective, instead of an engineering platform, crowdsourcing could be an incubator for innovations. It seeks ideas and solution options from crowd wisdom. Problems like GUI design can be open to communities to foster innovative ideas.

4.5.2 Architecture Design

The Cloud for crowdsourcing need to provide services for two categories of activities: software development and open collaboration. To support software development, it needs software lifecycle process tracking platform from requirements distribution to software testing and integration, code repositories with version control system and online IDE environment, document management system, and so on. To support community-based collaboration, it needs to support user account management, expert recommendation, group coordination, and so on. Following the open architecture principle, some of the services can be built by reusing with existing open services, for examples, GitHub as implementation repository, Amazon for computing and storage resources, Cloud9 as online IDE, Google App Engine for software hosting and Google Doc for group discussion. It can also integrate with other system such as sharing use accounts from different social network systems. In addition, it has unique requirements to support this new development method, especially for community management and collaborations. For example, reputation and profile management, expert searching, evaluation and recommendation, and so on.

4.5.3 Experiment: Educating Software Engineering by a Crowdsourcing Platform

A platform can be built for educating purpose so that SE courses from different Universities can share their resources and collaborate in an experiment process. Students can join teams and projects across the world to learn by experience global software development methods, techniques, and challenges.

4.6 Experimentation Design for Software Crowdsourcing

Wenjun Wu (Beihang University, CN)

Joint work of Yue Tao, Shuat Ali, Wei-Tek Tsai; Hong-Linh Truong; Ognjen Scekic; Donghu Lin; Xinjun Mao

In this group, we discussed how to design and conduct software crowdsourcing experiments to validate theoretical models and refine crowdsourcing algorithms. We also listed the major crowdsourcing infrastructures that can be used to support experiments. Given the dynamic nature of crowdsourcing activity, it is a challenge to develop a repeatable experimental plan. Researchers must consider both social and technical factors in a software crowdsourcing process. Social factors such as constitutes of the community, skill and knowledge levels of participants, the incentive of participation often have significant impact on possibility of replicating the same experiment. Technical factor such as software development process and cloud software products are also vital to enable in-depth investigation among the synergy between people and software system. A successful software experiment should carefully integrate these major elements in the settings and describe them in a clearly defined model.

With such a model, the research community of crowdsourcing can share their datasets and derive knowledge from them through cast studies and statistics-driven data analytics.

4.6.1 Experimental Approaches

There are three approaches to perform crowdsourcing experiments and analysis: data analysis based on the process of real projects on crowdsourcing platforms, controlled experiments and simulation.

First, one can always gather data from existing projects that are hosted on the major crowdsourcing platforms such Topcoder [1], Amazon Mechnical Turk [2], Github [3] and Sourceforge [4]. Such a community platform keeps track of software development activities of tens of thousands completed projects. Via web crawlers or access APIs of the platforms, we can download all the activity data and make further analysis to to verify software crowdsourcing models. The advantage of this analytic approach is to access real crowdsourcing data with a low monetary cost and time effort. It is especially useful for statistical analysis and case studies.

Second, when a researcher needs to run controlled experiments to compare different strategies and mechanisms with software crowdsourcing design, he can launch real projects on the platforms mentioned above. The positive side of this approach is that researchers can devise their own experimental scenarios and explore completely new schemes that may never be tried by others. But since the experiment is run on commercial platforms, the researcher has to pay for recruiting community workers as his study subjects.

Last, one can always rely on simulation tools to run software crowsourcing experiments. Actually, one of the talks presented in this seminar discussed how to design simulation experiments following the models of search-based software engineering to study software crowdsourcing. Also, agent-based simulation can be adopted to study incentive models and production quality assurance on large-scale crowds. The major benefit of simulation method is flexibility and low-cost. And it empowers researchers to check their models in a much larger setting than what is available in real environments.

Perhaps a hybrid method combining real projects, data analysis and simulations will be the best way to practice software crowdsourcing studies. Imaging we can model a software crowdsourcing process through data analysis of public crowdsourcing platforms. More case studies based on real projects can be done to improve the model. And the model can be further used on a simulation scenario to demonstrate emergent behaviors in a large-scale crowdsourcing system.

4.6.2 Experiment Design

A researcher needs to define his experimental objective, primary factors and study objects before starting his experiments on software crowdsourcing:

Whether does he plan to verify his model or algorithm? How can he model crowdsourcing tasks, workers, and costs in term of time and budget? How to design the mechanism for ranking and expertise matching to facilitate virtual team formation?

In the design of a controlled experiment, a researcher should isolate or control multiple design factors in order to identify primary factors that are most influential. Typical factors in a crowdsourcing experiment include participants, prize setting, and task design and so on. If he wants to test a crowdsourcing task among the same group of people, he may have to design a two-around crowdsourcing mechanism to retain the participants who have involved in the previous around.

Choice of study subjects in an experiment can also significantly affect the outcome. Since the community is often the study subject in a crowdsourcing project, a researcher must decide whether to run the project with IT professionals or college students. Our group believes that experiments with student groups can produce valid results as ones with professional developers. They are suitable in different kinds of experiments. IT professionals, especially software engineers are always good candidates for software engineering experiments that demands more advanced programming skills and expertise for system design. In other kinds of experiments with less restricted requirements for software design and programming, researcher can choose computer science students as his study subject.

The last but not the least concern for researchers is to design an appropriate ontology model to describe software crowdsourcing processes. If everyone organizes his datasets in a standard process model like SPEM [5] in UML specification, this data interoperability enables very convenient data exchange within the community and facilitates peer researchers to reuse the dataset in their experiments.

4.6.3 Experiment infrastructure

Our discussion group listed the major commercial platforms available for researchers. Well-known examples of crowdsourcing platforms are Topcoder, Amazon Mechanical Turk and Odesk [6]. The distinguishing features among these platforms include community and crowdsourcing style. Most players in the Topcoder community are young people with strong enthusiasm for programming. So when a researcher needs his coding project accomplished within a week, Topcoder seems a good place to go. Because most crowdsourcing tasks in Amazon Mechanical Turk are very cheap laboring jobs that often require only a few minutes of effort. So its community is not very suitable for programming tasks demanding dedication and high skills.

In addition to these above commercial platforms, group members highlighted open source crowdsourcing platforms under development. Vienna's team mentioned about a plan to release a crowdsourcing service platform. Beihang team is developing an educational platform to support Massive Open Online Courses (MOOCs), which could be extended to support software crowdsourcing with MOOC learners. National University of Defense Technology introduces two open source platforms namely Trustie (http://www.trustie.net/) and OW2 Open Source community (http://ow2.org/), which combine the traditional software engineering methods with the crowd wisdom, and provide a promising infrastructure to enable software crowdsourcing projects. Further work is needed to explore all the platforms and build up good cases so that we could summarize the best practices and define guidelines to perform software crowdsourcing experiments.

References

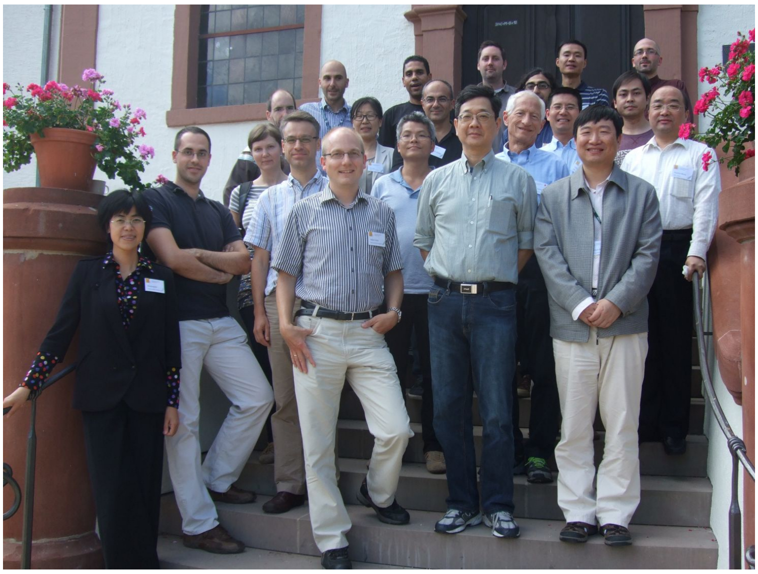
1 Topcoder, http://www.topcoder.com
2 Amazon Mechnical Turk, http://www.mturk.com
3 Github, http://www.github.org
4 Sourceforge, http://www.sourceforge.net
5 SPEM, http://www.omg.org/spec/SPEM/2.0/?
6 Odesk, http://www.odesk.com

5 Open Problems

Open problems were described throughout the previous sections, in particular, in the working group summaries.

Participants

- Shaukat Ali
Simula Research Laboratory – Lysaker, NO
- Xiaoying Bai
Tsinghua Univ. – Beijing, CN
- Xavier Blanc
University of Bordeaux, FR
- Kyle Chard
University of Chicago, US
- Schahram Dustdar
TU Wien, AT
- Michael N. Huhns
University of South Carolina – Columbia, US
- Robert Kern
IBM Deutschland – Böblingen, DE
- Donghui Lin
Kyoto University, JP
- Greg Little
ODesk Corp. – Redwood City, US
- Xinjun Mao
National University of Defense Technology – Hunan, CN
- Michael Maximilien
IBM Almaden Center – San José, US
- Dave Murray-Rust
University of Edinburgh, GB
- Khrystyna Nordheimer
Universität Mannheim, DE
- Dirk Riehle
Univ. Erlangen-Nürnberg, DE
- Ognjen Scekic
TU Wien, AT
- Lionel Seinturier
Lille I University, FR
- Hong-Linh Truong
TU Wien, AT
- Wei-Tek Tsai
ASU – Tempe, US
- Huaimin Wang
NUDT – Hunan, CN
- Wenjun Wu
Beihang University – Beijing, CN
- Gang Yin
NUDT – Hunan, CN
- Tao Yue
Simula Research Laboratory – Lysaker, NO



Report from Dagstuhl Seminar 13371

Quantum Cryptanalysis

Edited by

Serge Fehr[1], Michele Mosca[2], Martin Rötteler[3], and Rainer Steinwandt[4]

1 **CWI – Amsterdam, NL,** serge.fehr@cwi.nl
2 **University of Waterloo, CA and Perimeter Institute for Theoretical Physics, Waterloo, CA,** mmosca@iqc.ca
3 **Microsoft Research – Redmond, US,** martinro@microsoft.com
4 **Florida Atlantic University – Boca Raton, US,** rsteinwa@fau.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13371 "Quantum Cryptanalysis". In the first part, the motivation and organizational aspects of this meeting are outlined. Thereafter, abstracts for the presentations are provided (sorted alphabetically by last name of the presenter).

Seminar 08.–13. September, 2013 – www.dagstuhl.de/13371
1998 ACM Subject Classification E.3 Code Breaking, F.2 Analysis of Algorithms and Problem Complexity, G.2 Discrete Mathematics, G.3 Probability and Statistics
Keywords and phrases security of cryptographic schemes, quantum algorithms, computational hardness assumptions
Digital Object Identifier 10.4230/DagRep.3.9.59
Edited in cooperation with John M. Schanck

1 Executive Summary

Serge Fehr
Michele Mosca
Martin Rötteler
Rainer Steinwandt

Motivation and Background

This (second) quantum cryptanalysis seminar aimed at improving our understanding of quantum attacks against modern cryptographic schemes, a task that is closely related to the question of plausible quantum computational hardness assumptions. By bringing together researchers who work in the field of quantum computing with those who work in the field of classical cryptography, the seminar aimed at identifying practical approaches to achieve cryptographic security in the presence of quantum computers. A lesson learned from an earlier edition of this seminar (Dagstuhl Seminar 11381) was that statements about the security of cryptographic schemes in the presence of a quantum attacker require the study and characterization of quantum security parameters. Those parameters measure the amount of resources that have to be spent in order to "break" a system. In this spirit, the following three topics turned out to be particularly relevant for the seminar:

- *Quantum attacks on currently deployed schemes*: Derive quantitative estimates for the resources (like no. of qubits and quantum gates) that are needed to carry out quantum attacks with cryptographically relevant parameter choices.
- *New quantum algorithms to attack potential new hardness assumptions*: For instance, can quantum algorithms be used to improve on classical solutions for computational problems in lattices or for the decoding of error-correcting codes?
- *Quantum computational assumptions*: Which problems are currently considered as intractable, even for a quantum computer, and possibly might have the potential to be of cryptographic interest? Examples are certain hidden shift and hidden subgroup problems.

One indicator for the importance of these topics for the seminar was that most talks addressed (at least) one of them. The invited group of researchers as well as the organizing team was chosen to offer a balance of expertise from the different relevant disciplines, but also to have a substantial common ground for making progress towards the seminar goal.

Seminar Organization

The seminar involved 37 participants from around the globe, ranging from young researchers to colleagues with many years of interdisciplinary research experience. For young researchers the interdisciplinary set-up of the seminar offered an excellent opportunity to make new connections beyond the familiar research communities. Based on the experience from the predecessor (Dagstuhl Seminar 11381), we decided for a schedule which has enough flexibility to add presentations that grow out of discussions during the week, and indeed these additional slots could be brought to good use. We made an effort to keep the number of presentations limited to have ample time for open discussions between presentations. Having two research communities present at the meeting, it also seemed realistic to assume that not all participants are familiar with the latest developments in the complementing discipline. Placing survey presentations on critical topics early in the schedule was well received by the participants.

To ensure an adequate connection with the technological state-of-the-art of implementing quantum computers, one of the survey presentations was specifically devoted to this subject, and the seminar included discussions on implementation aspects of quantum computing. Keeping with the Dagstuhl tradition and the tradition of the predecessor, for Wednesday afternoon we did not schedule any presentations, allowing seminar participants to enjoy a hike in the woods, a visit to Trier, or to use the time for longer technical discussions.

Achievements and Next Steps

As in the first edition of this seminar, there were many fruitful discussions across discipline boundaries. At the time of writing this report, two seminar participants had already published a preprint with a generalization of a previously known quantum attack to a more general class of algebraic structures. We expect further publications to come forward in the coming months. While we are still far from a thorough understanding of the cryptanalytic potential of quantum computing, synergetic collaborations of seminar participants have helped greatly to advance the state-of-the-art in quantum cryptanalysis.

The seminar also successfully facilitated the exchange among colleagues from academia, government, and industry. We believe that in regard to a standardization of post-quantum cryptographic solutions, this type of exchange across community boundaries is valuable and deserves to be intensified further in future meetings.

2 Table of Contents

3 Overview of Talks

3.1 Quantum algorithms for the subset-sum problem

Daniel J. Bernstein (University of Illinois – Chicago, US)

Joint work of Bernstein, Daniel J.; Jeffery, Stacey; Lange, Tanja; Meurer, Alexander

At Eurocrypt 2010, Howgrave-Graham and Joux introduced a subset-sum algorithm with heuristic asymptotic cost exponent 0.337. The idea, suitably adapted, can be combined with quantum walks, achieving exponent 0.241; this is joint work with Stacey Jeffery, Tanja Lange, and Alexander Meurer. I'll also include some experimental material on simulating quantum algorithms.

3.2 Hash-based signatures

Johannes A. Buchmann (TU Darmstadt, DE)

Joint work of Buchmann, Johannes A.; Coronado, Corlaos: Dahmen, Eric; Hülsing, Andreas

Digital signatures a very important tools for the protection of IT systems and, in particular, the Internet. They protect the authenticity of software updates. This is crucial since operating systems and application software are never error-free and must be fixed on a regular basis.

The digital signature schemes that are currently being used in practice are RSA or methods whose security is based on the hardness of computing discrete logarithms over finite fields or in the group of points of elliptic curves over finite fields. As Peter Shor showed in the late 90s, all these schemes can be broken by quantum computers. It is therefore essential to come up with alternative that resists quantum computer attacks.

In my talk I report on the extended Merkle signature scheme (XMSS) that is based on the Merkle signature scheme invented in the late 70s. I show that this scheme is very promising both from the theoretical and practical side. On the theoretical side, I prove that XMSS has minimal security requirements. In fact, it can be shown, that a secure instance of XMSS can be constructed as long as there is a one-way function. As one-way functions are known to be the minimal security requirement for digital signatures, this shows that XMSS has in fact minimal security requirements. I also report on the algorithmic improvements that have been found in my research group in the last years. Implementations and experiments show that the latest version of XMSS can compete with established signature schemes.

3.3 A new definition for the quantum conditional Rényi entropy

Frederic Dupont-Dupuis (Aarhus University, DK)

Joint work of Müller-Lennert, Martin; Dupuis, Frédéric; Szehr, Oleg; Fehr, Serge; Tomamichel, Marco
Main reference M. Müller-Lennert, F. Dupuis, O. Szehr, S. Fehr, M. Tomamichel, "On quantum Renyi entropies: a new generalization and some properties," arXiv:1306.3142v3 [quant-ph], 2013.
URL http://arxiv.org/abs/1306.3142v3

Very recently, a new definition for the quantum conditional Rényi entropy has been proposed. Unlike previous definitions, it satisfies most of the basic properties one would expect from a conditional entropy, and it coincides with other widely used entropies (min-, max- and collision entropy) for appropriate choices of parameters. It has also found an application to prove a strong converse for coding over entanglement breaking channels. I will present these recent developments and try to show why this might be the "right" definition.

The results presented can be found in the following papers:

References

1 Martin Müller-Lennert, Frédéric Dupuis, Oleg Szehr, Serge Fehr, Marco Tomamichel. *On quantum Rényi entropies: a new definition and some properties*. arXiv:1306.3142.
2 Mark M. Wilde, Andreas Winter, Dong Yang. *Strong converse for the classical capacity of entanglement-breaking channels*. arXiv:1306.1586.
3 Rupert L. Frank, Elliott H. Lieb. *Monotonicity of a relative Rényi entropy*. arXiv:1306.5358.
4 Salman Beigi. *Quantum Rényi divergence satisfies data processing inequality*. arXiv:1306.5920.

3.4 On quantum versions of the McEliece cryptosystem

Markus Grassl (National University of Singapore, SG)

Joint work of Grassl, Markus; Ezerman, Fred

Recently, quantum versions of the McEliece cryptosystem have been proposed. We will take a closer look at them and discuss consequences of particular choices made in the design, in particular the use of so-called CSS codes, a subclass of stabilizer quantum codes. This yields to some interesting questions which are open to us.

3.5 Lattice-based cryptography

Nadia Heninger (University of Pennsylvania, US)

We give a short survey of hard lattice problems and lattice-based cryptography. We discuss the shortest vector problem and closest vector problem, show how they are related to the short integer solution and learning with errors problem, and show how to construct collision-resistant hash functions and public-key encryption systems based off of the average case hardness of these problems. We finish with a discussion of ring learning with errors and hard problems on ideal lattices.

3.6 Hidden subgroup problems in quantum-resistant cryptography?

Gabor Ivanyos (Hungarian Academy of Sciences, HU)

This talk addressed the potential hardness of some simple Hidden-Subgroup-like problems for quantum computers. One of these is the discrete logarithm problem over a black box group with non- unique encoding. Another is the shift problem over cyclic groups (equivalently, the dihedral HSP). Regev's reduction to a version of the dihedral HSP suggests that it might be hard. Some easy and potentially difficult natural non-oracle instances of these problems were discussed: shift problems over cyclic permutations groups (e.g., shifts of graphs and hypergraphs), or shift problems over cyclic linear groups (e.g., shifts of linear subspaces).

3.7 Quantum walks for cryptanalysis

Stacey Jeffery (University of Waterloo, CA)

The aim of this talk will be to present the quantum walk search algorithm framework, and several recent extensions, so that a crypto audience will be able to apply the framework to construct quantum attacks on cryptosystems. I will present several examples of quantum walk algorithms that can solve very general types of problems, and explain how these may be adapted to other settings.

3.8 Quantum lattice cryptanalysis – Part 1

Thijs Laarhoven (TU Eindhoven, NL)

Main reference T. Laarhoven, M. Mosca, J. van de Pol , "Solving the Shortest Vector Problem in Lattices Faster Using Quantum Search," in Proc. of the 5th Int'l Conf. on Post-Quantum Cryptography (PQCrypto'13), LNCS, Vol. 7932, pp. 83–101, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-38616-9_6

Recently lattices have found many applications in cryptography, both in constructive "post-quantum" cryptography (FHE) and in cryptanalysis. Estimating the (quantum) complexity of lattice algorithms is crucial in understanding the security of lattice-based cryptography, and for choosing parameters in these cryptosystems. In Part 1, I will discuss how quantum algorithms can be used to significantly speed up two algorithms for finding shortest vectors in lattices, namely sieving and saturation. Other lattice algorithms, for which we did not yet obtain quantum speed-ups, are discussed in Part 2.

3.9 Comments on computing using a D-Wave chip

Bradley Lackey (National Security Agency – Fort Meade, US)

It is not uncommon in the literature to find the "D-Wave problem" referring to the problem of finding a ground state of a (typically random) Ising model on a particular graph. What a D-Wave chip does is produce a large number of samples from the lowest energy configurations of the programmed Ising model Hamiltonian. In this open discussion session, I commented that there are computational problems (e.g. decoding an LDPC code or, in more generality, Bayesian re-estimation) that can be formulated as finding the Helmholtz distribution of an Ising model Hamiltonian at some positive temperature. Since this is similar to what a D-Wave chip does empirically, it may be possible to solve such problems on a D-Wave machine better than known techniques. As an example I indicated that small cycles in the Tanner graph of an LDPC code's parity check matrix are known to cause belief propagation difficulties, however such structures seem irrelevant if one were to sample from the Helmholtz distribution directly.

3.10 Recent advances in decoding random binary linear codes

Alexander May (Ruhr-Universität Bochum, DE)

URL http://www.cits.ruhr-uni-bochum.de/personen/may.html

We review some recent algorithmic progress that led to faster algorithms for decoding random linear codes over $\mathbb{F}_2^n$. The worst case complexity for decoding these codes dropped from roughly $2^{n/8}$ to $2^{n/10}$. We also discuss some open problems whose (quantum) solution would provide further progress.

References

1 Anja Becker, Antoine Joux, Alexander May, Alexander Meurer *Decoding Random Binary Linear Codes in $2^{n/20}$: How $1+1=0$ Improves Information Set Decoding* , In Advances in Cryptology (Eurocrypt 2012), Lecture Notes in Computer Science, Springer-Verlag, 2012

2 Alexander May, Alexander Meurer, Enrico Thomae *Decoding Random Linear Codes in $O(2^{0.054n})$* , In Advances in Cryptology (Asiacrypt 2011), Lecture Notes in Computer Science, Springer-Verlag, 2011

3.11 Code-based verifiable encryption

Kirill Morozov (Kyushu University, JP)

Joint work of Hu, Rong; Morozov, Kirill; Takagi, Tsuyoshi
Main reference R. Hu, K. Morozov, T. Takagi, "Proof of plaintext knowledge for code-based public-key encryption revisited," in Proc. of the 8th ACM SIGSAC Symp. on Information, Computer and Communications Security (ASIA CCS'13), pp. 535–540, ACM, 2013.
URL http://dx.doi.org/10.1145/2484313.2484385

In this talk, we present a verifiable encryption with equality relation for the IND-CPA McEliece public key encryption. More specifically, this is a zero-knowledge (ZK) proof that a given (IND-CPA McEliece) ciphertext contains a given plaintext without revealing any information about the randomness used for encryption. Instrumental in our construction is code-based ZK identification scheme by Stern. Potential applications include designated confirmer signatures and escrow schemes.

3.12 Easy and hard functions for the Boolean hidden shift problem

Maris Ozols (IBM TJ Watson Research Center – Yorktown Heights, US)

Joint work of Childs, Andrew M.; Kothari, Robin; Ozols, Maris; Roetteler, Martin
Main reference A. M. Childs, R. Kothari, M. Ozols, M. Roetteler, "Easy and hard functions for the Boolean hidden shift problem," arXiv:1304.4642v1 [quant-ph], 2013.
URL http://arxiv.org/abs/1304.4642v1

We study the quantum query complexity of the Boolean hidden shift problem. Given oracle access to $f(x + s)$ for a known Boolean function f, the task is to determine the n-bit string s. The quantum query complexity of this problem depends strongly on f. We demonstrate that the easiest instances of this problem correspond to bent functions, in the sense that an exact one-query algorithm exists if and only if the function is bent. We partially characterize the hardest instances, which include delta functions. Moreover, we show that the problem is easy for random functions, since two queries suffice. Our algorithm for random functions is based on performing the pretty good measurement on several copies of a certain state; its analysis relies on the Fourier transform. We also use this approach to improve the quantum rejection sampling approach to the Boolean hidden shift problem.

3.13 On group isomorphism problem when Cayley tables are given

Youming Qiao (National University of Singapore, SG)

Joint work of Grochow, Joshua A.; Qiao, Youming
Main reference J. A. Grochow, Y. Qiao, "Algorithms for group isomorphism via group extensions and cohomology," arXiv:1309.1776v1 [cs.DS], 2013; and ECCC TR13–123.
URL http://arxiv.org/abs/1309.1776v1
URL http://eccc.hpi-web.de/report/2013/123/

Given the Cayley tables of two finite groups of order n, it is easy to test their isomorphism in time $n^{\log n+O(1)}$. The main question is to bring to polynomial time. Recently polynomial-time algorithms for several group classes have been devised, e.g. [1], [2], and [3]. In this work, we show that there is a common underlying scheme supporting these previous works: naively speaking, the key is to transform testing isomorphism of abstract groups, to testing whether certain functions arising from these abstract groups are in the same orbit under some concrete group action. These functions are derived from the extension theory of groups. Based on this underlying scheme, we devise a polynomial-time algorithm to test isomorphism of groups as central extensions of elementary abelian groups by a direct product of nonabelian simple groups. We then briefly describe the ingredients and ideas used to settle a minimal case in the work [2], as well as the above mentioned group class.

References

1 László Babai, Paolo Codenotti, and Youming Qiao. Polynomial-time isomorphism test for groups with no abelian normal subgroups – (extended abstract). In *ICALP*, pages 51–62, 2012.
2 László Babai and Youming Qiao. Polynomial-time isomorphism test for groups with Abelian Sylow towers. In *29th STACS*, LIPIcs, Vol. 14, pp. 453–464, Schloss Dagstuhl, doi: 10.4230/LIPIcs.STACS.2012.453.
3 Mark L. Lewis and James B. Wilson. Isomorphism in expanding families of indistinguishable groups. *Groups – Complexity - Cryptology*, 4(1):73–110, 2012.

3.14 Complete insecurity of quantum protocols for classical two-party computation

Christian Schaffner (University of Amsterdam, NL)

Main reference H. Buhrman, M. Christandl, C. Schaffner, "Complete Insecurity of Quantum Protocols for Classical Two-Party Computation," Physical Review Letters, Volume 109, Issue 16, 160501 (5pp.), 2012.
URL http://dx.doi.org/10.1103/PhysRevLett.109.160501

A fundamental task in modern cryptography is the joint computation of a function which has two inputs, one from Alice and one from Bob, such that neither of the two can learn more about the other's input than what is implied by the value of the function. In this work, we show that any quantum protocol for the computation of a classical deterministic function that outputs the result to both parties (two-sided computation) and that is secure against a cheating Bob can be completely broken by a cheating Alice. Whereas it is known that quantum protocols for this task cannot be completely secure, our result implies that security for one party implies complete insecurity for the other. Our findings stand in stark contrast to recent protocols for weak coin tossing and highlight the limits of cryptography within quantum mechanics. We remark that our conclusions remain valid, even if security is only required to be approximate and if the function that is computed for Bob is different from that of Alice.

3.15 Practical signatures from the partial Fourier recovery problem

John M. Schanck (University of Waterloo, CA)

Joint work of Hoffstein, Jeff; Pipher, Jill; Schanck, John M.; Silverman, Joseph H.; Whyte, William
Main reference J. Hoffstein, J. Pipher, J. Schanck, J. H. Silverman, W. Whyte, "Practical Signatures from the Partial Fourier Recovery Problem," Cryptology ePrint Archive: Report 2013/757, 2013.
URL http://eprint.iacr.org/2013/757

PASS, an authentication and signature scheme based on the problem of finding bounded norm preimages of a partial discrete Fourier transform, was first presented by Hoffstein, Lieman, and Silverman in 1999. The system had several attractive features, but suffered from transcript analysis attacks which rendered it completely insecure. Perhaps as a consequence, this interesting trapdoor function has received little attention since its introduction. In my presentation I demonstrated how recent techniques developed for lattice cryptography can be used to prevent the transcript attacks while salvaging most of the efficiency of the scheme, and discussed several open questions related to the scheme's security. Concrete parameter sets and benchmarks were also be presented.

A paper describing the system is in preparation, and software implementing it is available at https://github.com/NTRUOpenSourceProject/ntru-crypto.

3.16 Generic decoding of linear codes

Nicolas Sendrier (INRIA – Siège, FR)

The security of code-based cryptosystems relies heavily on the hardness of decoding in an arbitrary linear code. The problem only has exponential solutions in the classical computing model, and it seems that it remains the case with quantum computing, placing code-based crypto among the so-called post-quantum cryptographic techniques.

Even though quantum computer are not likely to allow generic decoding techniques with non-exponential complexity, it still allows some significant improvements. A few works have already explore this path.

The purpose of this survey is not to propose new decoding techniques taking advantage a quantum computing. Instead we intend to give a comprehensive description of the main decoding techniques. The best techniques are recent (2011 and 2012) and are in fact finely tuned trade-offs between several very basic techniques. We will come back to those basics, which are the birthday decoding (based on the birthday paradox), and the Prange or Lee & Brickell algorithms (which are essentially enumerations).

We feel that understanding each of those techniques in a quantum setting will simplify the search of an optimal quantum algorithm, and, hopefully, provide a simpler approach than directly "quantumize" the best known decoding techniques.

3.17 Quantum-resistant multivariate public key cryptography

Daniel Smith (NIST – Gaithersburg, US)

We discussed some of the developments which have arisen in post- quantum multivariate schemes. In particular, we focused on some of the fundamental complexity theoretic problems on which the security of a variety of multivariate schemes depend. We reviewed the classical complexity of the Morphism of Polynomials (MP) Problem as well as the closely related problems Isomorphism of Polynomials (IP) and IP with one secret (IP1S).

Some interesting complexity theoretic results are known in the classical model; however, IP1S lies in a particularly interesting location in the complexity hierarchy. The IP1S problem is graph isomorphism-complete, which places it in a small class of problems which are believed to lie strictly between P and NP. Since the factoring problem thus far seems to exist in this nether region in the classical model, there is some justification in addressing the complexity of IP1S in the quantum model.

For completeness, the discussion reviewed several relevant computational techniques of use in the cryptanalysis of modern multivariate schemes, and presented some recent theoretical results establishing some benchmarks for security, including classification techniques and theorems proving immunity from structural attacks as well as the calculation of degree of regularity indicating the complexity of algebraic analysis. There seems to be a dichotomy in multivariate cryptanalysis, with many schemes falling into potential wells into a realm of structural weakness, or of susceptibility to algebraic attacks. It is precisely the collection of schemes which avoid the known types of structural weakness and which also have a high enough degree of regularity which have survived. While classification techniques are algebraic arguments independent of the model of computing, it is possible that there may be polynomial speed-ups for generic algebraic polynomial system solvers in the quantum model.

The discussion concluded with reference to some of the recent developments in these areas, including some recent advances in the complexity theory of the above mentioned fundamental problems, as well as some new security calculations for specific schemes in terms of structural classification theorems and degree of regularity calculations.

3.18 Exponential improvement in precision for Hamiltonian-evolution simulation

Rolando Somma (Los Alamos National Lab., US)

Joint work of Somma, Rolando; Berry, Dominic; Cleve, Richard
Main reference Dominic W. Berry, Richard Cleve, Rolando D. Somma, "Exponential improvement in precision for Hamiltonian-evolution simulation," arXiv:1308.5424v3 [quant-ph], 2013.
URL http://arxiv.org/abs/1308.5424v3

I will present a quantum method for simulating Hamiltonian evolution with complexity polynomial in the logarithm of the inverse error. This is an exponential improvement over existing methods for Hamiltonian simulation. In addition, its scaling with respect to time is close to linear, and its scaling with respect to the time derivative of the Hamiltonian is logarithmic. These scalings improve upon most existing methods. Our method is to use a compressed Lie-Trotter formula, based on recent ideas for efficient discrete-time simulations of continuous-time quantum query algorithms.

3.19 Single-qubit quantum circuit decomposition

Krysta Svore (Microsoft Research – Redmond, US)

Joint work of Bocharov, Alex; Gurevvich, Yuri; Svore, Krysta
Main reference A. Bocharov, Y. Gurevich, K. M. Svore, "Efficient Decomposition of Single-Qubit Gates into V Basis Circuits," Physical Review A, Vol. 88, Issue 1, No. 012313 (13pp.); and available as pre-print arXiv:1303.1411v1 [quant-ph] , 2013.
URL http://dx.doi.org/10.1103/PhysRevA.88.012313
URL http://arxiv.org/abs/1303.1411v1

In this talk, I will present recent developments in single-qubit quantum circuit decomposition, a necessary component to implementation of quantum algorithms. In the past year, constructive algorithms for decomposing single- qubit rotations efficiently into $O(\log(1/\epsilon))$-depth circuits have been developed for several universal bases. I will present two algorithms for compiling single-qubit unitary gates into circuits over the universal V basis. The V basis is an alternative universal basis to the more commonly studied $\{H, T\}$ basis. The first algorithm has expected polynomial time (in precision $\log(1/\epsilon)$) and offers a depth/precision guarantee that improves upon state-of-the-art methods for compiling into the $\{H, T\}$ basis by factors ranging from 1.86 to $\log_2(5)$. The second algorithm is analogous to direct search and yields circuits a factor of 3 to 4 times shorter than our first algorithm, and requires time exponential in $\log(1/\epsilon)$; however, in practice the runtime is reasonable for an important range of target precisions.

3.20 Quantum lattice cryptanalysis – Part 2

Joop van de Pol (University of Bristol, GB)

Joint work of Laarhoven, Thijs; Mosca, Michele; van de Pol, Joop
Main reference T. Laarhoven, M. Mosca, J. van de Pol, "Solving the Shortest Vector Problem in Lattices Faster Using Quantum Search," in Proc. of the 5th Int'l Conf. on Post-Quantum Cryptography (PQCrypto'13), LNCS, Vol. 7932, pp. 83–101, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-38616-9_6

In Part 1 we have seen that we can use quantum algorithms to speed up lattice cryptanalytic algorithms such as the sieving/saturation algorithms. But what about the other lattice algorithms that are used in cryptanalysis? I will briefly describe our attempts in applying similar methods to these algorithms (basis reduction, enumeration and Voronoi cell) and the obstacles we encountered.

3.21 Simulating quantum circuits with sparse output distributions

Maarten van den Nest (MPI für Quantenoptik, DE)

Joint work of Schwarz, Martin;

We show that several quantum circuit families can be simulated efficiently classically if it is promised that their output distribution is approximately sparse i.e. the distribution is close to one where only a polynomially small – though a priori unknown – subset of the measurement probabilities are nonzero. Classical simulations are thereby obtained for quantum circuits which, without the sparsity promise, are considered hard to simulate. Our results apply in particular to a family of Fourier sampling circuits which contain Shor's factoring algorithm as a special instance, but also to other circuit families, such as IQP circuits. To achieve an efficient classical simulation, we employ and extend the Goldreich-Levin algorithm in combination with probabilistic simulation methods for quantum circuits.

3.22 Implementations of quantum computers

Frank Wilhelm-Mauch (Universität des Saarlandes, DE)

Implementing a scalable and useful quantum computer is a daunting task. I turns out that this goal has come a lot closer due to two reasons: i) the discovery of surface codes for error correction, which promise high error thresholds based on nearest-neighbour interaction only and ii) the progress in quantum computer implementations. I have mostly presented point ii) by giving an introduction to superconducting qubits and by reviewing milestones such as the great improvement of control and coherence over the last decade.

This presentation was mostly giving context to the main theme of the workshop.

Participants

- Aleksandrs Belovs
University of Latvia, LV
- Daniel J. Bernstein
Univ. of Illinois – Chicago, US
- Johannes A. Buchmann
TU Darmstadt, DE
- Andrew Childs
University of Waterloo, CA
- Frédéric Dupont-Dupuis
Aarhus University, DK
- Serge Fehr
CWI – Amsterdam, NL
- Katalin Friedl
Budapest University of Technology & Economics, HU
- Markus Grassl
National Univ. of Singapore, SG
- Nadia Heninger
University of Pennsylvania, US
- Peter Høyer
University of Calgary, CA
- Gabor Ivanyos
Hungarian Acad. of Sciences, HU
- Stacey Jeffery
University of Waterloo, CA
- Stephen P. Jordan
NIST – Gaithersburg, US
- Thijs Laarhoven
TU Eindhoven, NL
- Bradley Lackey
National Security Agency – Fort Meade, US
- Tanja Lange
TU Eindhoven, NL
- Yi-Kai Liu
NIST – Gaithersburg, US
- Alexander May
Ruhr-Universität Bochum, DE
- Kirill Morozov
Kyushu University, JP
- Michele Mosca
University of Waterloo and Perimeter Institute for Theoretical Physics – Waterloo, CA
- Maris Ozols
IBM TJ Watson Res. Center – Yorktown Heights, US
- Youming Qiao
National Univ. of Singapore, SG
- Martin Rötteler
Microsoft Res. – Redmond, US
- Louis Salvail
University of Montreal, CA
- Miklos Santha
University Paris-Diderot, FR
- Christian Schaffner
University of Amsterdam, NL
- John M. Schanck
University of Waterloo, CA
- Nicolas Sendrier
INRIA – Siège, FR
- Daniel Smith
NIST – Gaithersburg, US
- Rolando Somma
Los Alamos National Lab., US
- Fang Song
Penn State University, US
- Rainer Steinwandt
Florida Atlantic University – Boca Raton, US
- Krysta Svore
Microsoft Res. – Redmond, US
- Wim van Dam
University of California – Santa Barbara, US
- Joop van de Pol
University of Bristol, GB
- Maarten van den Nest
MPI für Quantenoptik, DE
- Frank Wilhelm-Mauch
Universität des Saarlandes, DE

Report from Dagstuhl Seminar 13372

Integration of Tools for Rigorous Software Construction and Analysis

Edited by

Uwe Glässer[1], Stefan Hallerstede[2], Michael Leuschel[3], and Elvinia Riccobene[4]

1 **Simon Fraser University, B.C., CA,** glaesser@cs.sfu.ca
2 **Aarhus University, DK,** stefan.hallerstede@wanadoo.fr
3 **Heinrich-Heine-Universität Düsseldorf, DE,** leuschel@cs.uni-duesseldorf.de
4 **University of Milan, IT,** elvinia.riccobene@unimi.it

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13372 "Integration of Tools for Rigorous Software Construction and Analysis". The 32 participants came from 10 countries: Australia, Austria, Brazil, Canada, Denmark, France, Germany, Great Britain, Italy, Norway. The aim of the seminar was to bring together researchers and tool developers from different state- and machine-based formal methods communities in order to share expertise and promote the joint use of modelling tool technologies. Indeed, each of these communities – from Abstract State Machines, to B, TLA, VDM, Z – has valuable tools and technologies which would be beneficial also for the other formal approaches. Understanding and clarifying their commonalities and differences is a key factor to achieve a possible integration or integrated use of these related approaches for accomplishing, in a rigorous way, the various modelling and analysis tasks to construct reliable high quality software systems.

The working group formula offered by the Dagstuhl seminar was a fruitful way to share knowledge of the various techniques and tools – such as simulators, animators, model checkers, theorem proves – developed for the individual methods, and to constructively experiment the combined use of different approaches by means of a series of well known case studies. Participants did not arrive with well-prepared solutions, but all the modelling and integration work was directly done in Dagstuhl in a very exciting and competitive atmosphere. Some related presentations were also given on recent advances on methodologies and tools.

The seminar posed the basis for a series of future research collaborations between different, and until now closed, formal method communities. An LNCS volume will be prepared from the contributions of the participants to give the *common vision* of future methodology and tool integration.

Seminar 08.–13. September, 2013 – www.dagstuhl.de/13372

1998 ACM Subject Classification D.2.4 Software/Program Verification, F.3.1 Specifying and Verifying and Reasoning about Programs, F.4.1 Mathematical Logic, I.6.4 Model Validation and Analysis, I.6.5 Model Development, I.6.8 Types of Simulation

Keywords and phrases Applied Formal Methods, Modelling Formalisms, Modelling Tools, Abstract State Machines, B Method, Event-B, TLA$^+$, VDM, Z, Verification, Validation, Proof, Simulation, Animation, Visualisation, Model Checking, Tool Integration

Digital Object Identifier 10.4230/DagRep.3.9.74

Edited in cooperation with Hamed Yaghoubi Shahir

1 Executive Summary

Uwe Glässer
Stefan Hallerstede
Michael Leuschel
Elvinia Riccobene

Motivation

Dagstuhl Seminar 06191 had been a success in establishing the "ABZ" joint conference for the different state-based modelling communities (e.g., ASM, B, VDM, Z, TLA$^+$) with venues in London (2008), Orford, CA (2010), Pisa (2012) and Toulouse (2014). It was a first step toward bringing these communities closer together. However, the conference, although being a place where the researchers meet, does not produce in itself a significant number of collaborations across the communities. The organisers of this seminar consider such collaborations vital in order to achieve a larger impact academically and industrially.

Aims of the seminar

The seminar aims to

1. Inspire exchange and joint use of formal modelling tool technologies
2. Establish long-term cross-community collaboration
3. Work towards a common vision on formal modelling

Points 2 and 3 are particularly important for future tool developments, a common methodical foundation, and more economic use of the necessary and available resources.

Preparation

At first the organisers intended to give the participants of the seminar case studies in advance that the participants could work on prior to the seminar to showcase their methods and tools. However, a downside of this common organisational practice is that most attendees arrive at the seminar with well-prepared, polished formal models and presentations. This would have resulted in conference-style presentations, not leaving much room for cross-community group work on problems with mixed-method approaches. Thus, no substantial gain above and beyond what the "ABZ" conferences already accomplish would have been achieved. Hence, the organisers decided to take the risk not to ask for advance preparation but have all the work done collaboratively during the seminar. This was thought to create a more open atmosphere and leave room for discussion. The organisers chose candidates for case studies to be carried out during the seminar and asked the participants to explore solutions with diverse methods in small, often mixed, groups formed dynamically based on interests. A tentative schedule for the week was published prior to the seminar. It was adapted by the organisers every night, taking into account the actual progress by the work groups and feedback received in plenum discussions held every day in the late afternoon or evening.

Execution

On Sunday evening the organisers held a three hour meeting to prepare day 1 of the seminar. It was decided that, in general, evenings should be left for the participants to socialise. The case study for day 1 needed to be well-chosen to engage the participants in the seminar. It was required that

- a single problem should be treated to minimise presentation overhead necessary for explaining the model
- the problem to be solved should not be trivial but solvable within 3 hours
- the problem should not leave too much room for interpretation so that the models, methods and tools used are more readily comparable
- the problem to be solved should come with a sketch of a solution so that focus would be on the modelling activity itself and not on finding the smartest solution.

The decision was made to use the problem of "Derivation of a termination detection algorithm for distributed computations" (EWD840) by E. W. Dijkstra.

On day 1, all but one group succeeded in producing a formal model. (That group produced their model on day 2 in the after-lunch session.) At the end of day 1, the organisers felt that not enough discussion across community boundaries was taking place. This would have to be addressed in the following days. The planning for day 2, payed specific attention to this aspect. In the evening of day 1, a two-hour planning meeting among the organisers was held. It was decided that a good way of getting the different communities involved in discussions would be to reshuffle the groups of day 1 somewhat. To carry out a comparison between the methods and tools, each group would have some members that produced the original model and some "envoys" of a group that had modelled the problem in a different notation. (This turned out to work well. By lunchtime on day 2, live discussions across the community boundaries had effectively started.) On suggestion of the participants, an originally planned plenum discussion on tool integration was carried out in three groups dealing with methodology, abstract syntax and low-level integration. (The actual number of groups was decided together with all participants in the beginning of the corresponding session. Even though it may have appeared frustrating at times for some participants, the organisers thought involving all participants in some of the decision making would also improve everyone's commitment.) The organisers also started incorporating talks. This was also considered useful for breaking the routine of the seminar.

In the evening of day 2, a one hour meeting among the organisers was held to plan day 3. It was thought that the participants could be involved closer by forming new groups that should each address a problem using two different approaches and tools. The comparison would then be possible while modelling. The modelling problem chosen was the FM'99 ATM modelling challenge. Two more talks followed on day 3 and some planning for integration meetings that should be held in smaller groups on day 4. The latter were considered to be fruitful by many with a lot of common interests being announced. In the afternoon of day 3, a shorter hike provided a welcome break, as the weather did not invite for larger excursions.

On day 3 in the evening, a 30 minute meeting of the organisers was held. The plan for day 4 was mostly to tie up the open threads from the preceding days. A short wrap up of the case studies followed by presentations giving a comparison between two methods. In the evening, a first discussion of post-seminar work was held, discussing the Dagstuhl report and a joint book on "comprehensive modelling and modelling tools". In the evening of day 4, a 30 minute meeting by the organisers was held. It was decided that the morning of day 5 should be spent discussing the joint book. Uwe Glässer presented an alternative case for use in the book to start a discussion about the writing approach that should be taken. (On day 5, it was decided to keep the ATM study but improve its description.)

Outcomes and Outlook

The main outcomes of the seminar are (a) various new collaborations across community boundaries to achieve a possible integrated use of different methods and tools, and (b) concrete plans for a book on "comprehensive modelling", a step towards a common vision of the research field. An agreement has been reached with Springer Verlag to publish the post proceedings of the seminar in the State-Of-The-Art series of Lecture Notes in Computer Science. This should improve visibility of the effort started at the seminar.

The organisers seek to get funding, e.g., by way of a network of excellence, to continue the integration work and keep up the momentum achieved during the seminar. The aim will be to develop a common vision and a more coordinated research agenda where, in particular, resources for tool development could be used more efficiently in future.

2 Table of Contents

3 Overview of Talks

3.1 Defining Communication for Abstract State Machines

Egon Börger (University of Pisa, IT)

Traditional ASMs come with the following classification of locations resp. functions (for executing agents a) [1, Ch.2.2.3]:

- *controlled*: readable and writable by a,
- *monitored*: only readable (not writable) by a,
- *output*: only writable (not readable) by a,
- *shared*: readable and writable by a and some other agent(s) subject to some conflict preventing protocol.

This classification turned out to be rather useful for numerous modelling endeavours, but it comes at the price of a rather abstract (*synchronous global*) way to deal with *communication between agents*: either via output ('send') and monitored ('receive') locations or via shared ('send/receive') locations. In particular the environment steps are supposed to happen only between two (discrete global) steps of an agent.

Our goal is to define a conservative extension of ASMs (to be compatible with previous ASM modelling work) by a model of interaction between agents which a) is helpful for practical modelling where communication between agents is an issue, b) is truly concurrent (not global synchronous) and c) offers an abstract *explicit Send/Receive* mechanism between ASMs with only private local locations. The definition should not be bound (but adaptable) to any concrete message passing system or transmission protocol, reflect in a uniform way synchronous and asynchronous communication and be implementable by an appropriate plugin for CoreAsm.

We propose for further experimentation concrete definitions for abstract synchronous and asynchronous versions of *Send/Receive* communication actions as update instructions, i.e. instructions generating updates à la CoreAsm [2]. We base these definitions on an abstraction of the input pool concept in S-BPM [3]. The details can be retrieved from the slides of the talk (and hopefully later from a paper we intend to write).

References

1 E. Börger and R. F. Stärk. *Abstract State Machines. A Method for High-Level System Design and Analysis.* Springer, 2003.
2 R. Farahbod et al. *The CoreASM Project.* http://www.coreasm.org.
3 A. Fleischmann, W. Schmidt, C. Stary, S. Obermeier, and E. Börger. *Subject-Oriented Business Process Management.* Springer Open Access Book, Heidelberg, 2012. www.springer.com/978-3-642-32391-1.

3.2 Describing the ATM in a Domain Specific Language

Albert Fleischmann (Metasonic AG – Pfaffenhofen, DE)

ASM and Event B are general methods for creating models and from practitioner's point of view not easy to understand. If requirements are defined by people belonging to the application domain then it is very helpful to use an appropriate specification language. Therefore it seems useful to create domain specific languages based on formal languages like event B or ASM. An example of such a language is S-BPM. S-BPM allows describing business processes. The semantics of S-BPM is defined as an ASM. The worked out example shows a S-BPM description of the automatic teller machine which is used as a use case. Behind that model there is a corresponding ASM and therefore the model is executable which means domain people can test a specification whether it meets their intentions.

3.3 Formal Modelling Problem: Distributed Termination Detection

Uwe Glässer (Simon Fraser University – B.C., CA)

Aiming at a simple requirements description that is concise and yet meaningful as an introductory illustrative example for comparing different state-oriented formalisation methods along with methodical aspects, we have chosen the termination detection algorithm for distributed computations proposed by Dijkstra et al. in 1983 [1]. This termination detection algorithm runs on a computer network with N machines, assuming a common network topology and reliable communication mechanisms. The original description derives the algorithm in several steps, together with an invariant that demonstrates that the algorithm works as expected. We presented the problem to the seminar participants in terms of an informal description of the basic requirements for the algorithm. A question and answer session provided further clarification. Additionally, a copy of the original paper by Dijkstra et al. was made available.

Interesting challenges in formal modelling dynamic properties of this algorithm are the inherently distributed nature of the problem and the quest for finding concise and yet appropriate abstract representations of interfaces between the algorithm and its operational environment. This environment refers to the distributed computations and the communication network. After working for several hours in small groups, a number of different solutions using diverse formalisation approaches emerged and were presented and discussed in a plenary session at the end of Day 1. Despite the relative simplicity of the problem, the comparison of the various solutions revealed interesting insights into strengths and weaknesses of the underlying formal methods.

This exercise provided a good starting point for a broader discussion about comparing and combining different methods and supporting tools for software and system analysis, validation and verification in order to complement their strengths. A central question is whether one should generally aim at tight integration methods based on common semantic frameworks or rather limit to loose integration based on transformations between formal models. While tight integration may be considered ideal, greatly simplifying the combined use of several methods, the cost of integrating continuously evolving tools appears to be prohibitive, severely limiting the practicability of any such approach. These open research questions need more scientific discussion to be answered comprehensively.

References

1 E.W. Dijkstra, W.H.J. Feijen, and A.J.M. van Gasteren. Information Processing Letters 16: 217-219, 1983.

3.4 System Modelling and Analysis using ASMETA

Paolo Arcaini (University of Bergamo, IT), Angelo Gargantini (University of Bergamo, IT), and Elvinia Riccobene (University of Milan, IT)

The ASMETA (ASM mETAmodeling) framework [1] is a set of tools around the Abstract State Machines (ASMs). It supports different activities of the system development process, from specification to analysis. ASMETA has been developed [2, 3] by exploiting concepts and technologies of Model-Driven Engineering (MDE). The starting point of the development has been the *Abstract State Machine Metamodel* (AsmM) [4], an abstract syntax description of a language for ASMs. *AsmetaL* is a platform-independent concrete syntax, and *AsmetaLc* a text-to-model compiler that parses AsmetaL models.

Simple model validation can be performed by *simulating* ASM models with the simulator *AsmetaS* [5] that supports *invariant checking*, *consistent updates checking* for revealing inconsistent updates, and *random* and *interactive* simulation. A more powerful validation approach is *scenario-based validation* by the ASM validator *AsmetaV* [6] that permits to express (through the *Avalla* modeling language) execution scenarios in an algorithmic way as interaction sequences of *actions* committed by the *user*.

Model review is a validation technique aimed at determining if a model is of sufficient quality; it allows to identify defects early in the system development, reducing the cost of fixing them. The *AsmetaMA* tool [7] permits to perform *automatic* review of ASMs; it looks for typical vulnerabilities and defects a developer can introduce during the modelling activity using the ASMs.

Formal verification of ASM models is supported by the *AsmetaSMV* tool [8] that maps AsmetaL models into specifications for the model checker NuSMV. It supports the declaration of both *Computation Tree Logic* (CTL) and *Linear Temporal Logic* (LTL) formulas.

Runtime verification is a technique that allows checking whether a run of a system under scrutiny satisfies or violates a given correctness property. *CoMA* (Conformance Monitoring by Abstract State Machines) [9] is a specification-based approach (and a supporting tool) for runtime monitoring of Java software: the conformance of a Java program is checked at runtime with respect to its formal specification given in terms of ASMs.

Model-based testing aims to use models for software testing. One of its main applications consists in test case generation where test suites are automatically generated from abstract models of the system under test. The ATGT tool [10] is available for testing of ASM models; it implements a set of adequacy criteria defined for the ASMs [11] to measure the coverage achieved by a test set and determine whether sufficient testing has been performed.

References

1 The ASMETA website. http://asmeta.sourceforge.net/, 2013.
2 P. Arcaini, A. Gargantini, E. Riccobene, and P. Scandurra. A model-driven process for engineering a toolset for a formal method. *Software: Practice and Experience*, 41:155–166, 2011.
3 A. Gargantini, E. Riccobene, and P. Scandurra. Combining formal methods and mde techniques for model-driven system design and analysis. *International Journal on Advances in Software*, 3(1,2):1–18, 2010.
4 A. Gargantini, E. Riccobene, and P. Scandurra. Ten Reasons to Metamodel ASMs. In J.-R. Abrial and U. Glässer, editors, *Rigorous Methods for Software Construction and Analysis*, volume LNCS 5115, pages 33–49. Springer Verlag, 2009.

5 A. Gargantini, E. Riccobene, and P. Scandurra. A Metamodel-based Language and a Simulation Engine for Abstract State Machines. *J. Universal Computer Science*, 14(12):1949–1983, 2008.

6 A. Carioni, A. Gargantini, E. Riccobene, and P. Scandurra. A Scenario-Based Validation Language for ASMs. In *Proceedings of the 1st international conference on Abstract State Machines, B and Z (ABZ '08)*, volume LNCS 5238, pages 71–84. Springer-Verlag, 2008.

7 P. Arcaini, A. Gargantini, and E. Riccobene. Automatic Review of Abstract State Machines by Meta Property Verification. In C. Muñoz, editor, *Proceedings of the Second NASA Formal Methods Symposium (NFM 2010)*, pages 4–13. NASA, 2010.

8 P. Arcaini, A. Gargantini, and E. Riccobene. AsmetaSMV: a way to link high-level ASM models to low-level NuSMV specifications. In *Abstract State Machines, Alloy, B and Z, 2nd Int. Conference (ABZ 2010)*, volume LNCS 5977, pages 61–74. Springer, 2010.

9 P. Arcaini, A. Gargantini, and E. Riccobene. CoMA: Conformance monitoring of java programs by abstract state machines. In *2nd International Conference on Runtime Verification, San Francisco, USA, September 27–30 2011*, 2011.

10 A. Gargantini, E. Riccobene, and S. Rinzivillo. Using Spin to Generate Tests from ASM Specifications. In *Proceedings of ASM 2003*, volume LNCS 2589. Springer Verlag, 2003.

11 A. Gargantini and E. Riccobene. ASM-Based Testing: Coverage Criteria and Automatic Test Sequence Generation. *J. Universal Computer Science*, 7:262–265, 2001.

3.5 Next-preserving Branching Bisimulation

Kirsten Winter (The University of Queensland, AU)

Bisimulations are in general equivalence relations between transition systems which assure that certain aspects of the behaviour of the systems are the same in a related pair. For many applications it is not possible to maintain such an equivalence unless non-observable (stuttering) behaviour is ignored. However, existing bisimulation relations which permit the removal of non-observable behaviour are unable to preserve temporal logic formulas referring to the next step operator. In this paper we propose a novel bisimulation relation, called next-preserving branching bisimulation, which accomplishes this, maintaining the validity of formulas with the next step, while still allowing non-observable behaviour to be reduced. Based on van Glabbeek and Weijland's notion of branching bisimulation with explicit divergence, we define the novel relation for which we prove the preservation of full CTL*.

As an example for its application we show how this definition gives rise to an advanced slicing procedure for temporal logics, a technique in which a system model is reduced to a slice which can be used as a substitute in verification and debugging. The result is a novel procedure for generating a slice that is next-preserving branching bisimilar to the original model. Hence, we can assure that all temporal logic properties are preserved in the slice and consequently the verification on the slice is sound.

3.6 Tooling Support for Formal Specification Languages

Alexander Raschke (Universität Ulm, DE) and Marcel Dausend (Universität Ulm, DE)

In our talk, we emphasised the creation of specifications. There are plenty of tools processing formal specifications, e.g. theorem provers, model checkers, simulators, but most specification languages are poorly supported during the modelling process itself. As opposed to this, programming languages provide modern IDEs which significantly improve the performance of creating, changing and understanding code. Especially features like e.g. autocompletion, debugging facilities, and refactoring support can help inexperienced users to write their requirements in a formal way. Also, the performance of expert users writing complex specifications can be improved. The purpose of the specification determines the set of possible supporting functions offered to the user. We identified at least three (possibly overlapping) purposes: execution, safety and security analysis, and rapid prototyping. For each purpose, different utilities that probably depend on each other can be offered. For any specification language, typical functions for writing assistance (e.g. auto- completion, quick-fixes), comfortable navigation (e.g. cross-reference, hypertext comments), static analysis (e.g. dead code and unused parameter detection), and graphical representation of the overall text structure are helpful. Users who execute specifications, can be supported by stepwise execution with logging, saving and resuming of system states, and interactive visualisation of executions. For safety and security analysis, existing tools for model checking and theorem proving can be integrated into the development environment. Rapid prototyping based on specifications can be supported by simple yet powerful GUI integration and execution statistics for profiling. The talk concluded with a presentation of the CoreASM tool that focuses on the execution of abstract state machines and already supports some of the presented features for enhancing the creation of specifications. Finally, we suggest to step up the common effort towards integrated tools for specification languages, especially taking into account abstractions, refinement, and integration of verification tools, like model checkers or theorem provers. The main goal of this effort should be to ease the overall process of formal methods application.

4 Working Groups: Distributed Termination Detection (DTD)

4.1 An Executable CoreASM Model of the Termination Detection Protocol

Marcel Dausend, Vincenzo Gervasi, Alexander Raschke, and Hamed Yaghoubi Shahir

Based on a literal translation of the English text into an ASM specification [1], we developed an executable specification in CoreASM. In order to achieve this goal, we have had to tackle the problems mentioned in [1]. We do not repeat the complete ASM model in this abstract. Instead, we focus on the details which have to be changed to make the ASM model executable and invite the reader to read [1] in advance.

The execution of the original translated specification resulted after a few steps in an inconsistent update set (as predicted in [1], Sect. 3), because of a simultaneous update of the current state of a machine to *active* and *passive*.

This happens, when a machine decides spontaneously to become passive although it received a message resulting in an activation. This issue was fixed determining that a machine is only allowed to become (spontaneously) passive if it does not receive a message in the same step. The first and the fourth mentioned problems in [1] can be solved by a special treatment of the master machine m_0. The colour of the master machine does not influence the colour of the token, but the master machine always sends a white token at the beginning of a new probe. For this different behaviour it is necessary to adapt the rule `PassToken` such that the `colorT` is possibly set to black, only if the current owner of the token is not the master machine itself. Additionally, the changed translation of $Rule_0$ considers that `PassToken` is only called if the current machine is not the master. With these changes, it is no more possible, that an inconsistent update set occurs for `colorT`.

The problem with the location `colorM` can be solved by introducing a new constraint in $Rule_{3+4}$ that the master can start a new probe by passing a token only if it is passive.

The inconsistent update set mentioned in problem two can be solved adding one new condition that prohibits sending messages to oneself. This solution also exploits a special property of asynchronous multi-agent ASMs, mentioned in [2, page 209]: "Let X be a finite initial segment of a run of an async ASM. All linearizations of X yield runs with the same final state." This means, all linearizations of runs that would result in an inconsistent update set are discarded. Thus, the execution of an async ASM will automatically avoid all cases, where `hasMessage(..)` is set and cleared simultaneously from different agents. Therefore, it is sufficient to avoid the case, when an inconsistent update set is generated within one agent.

In summary, four of the six mentioned problems could be solved by introducing some conditions and a special treatment of the master machine. Problem 5 could be solved by using queues, a background data structure which is provided by CoreASM, but this approach was – as well as problem 6 – not further examined. The (stepwise) execution of the model helped us identifying the problems and finding solutions for them.

References

1 Vincenzo Gervasi, Elvinia Riccobene, *From English to ASM: On the process of deriving a formal specification from a natural language one.* Dagstuhl Report of Seminar 13372, 2013.
2 Egon Börger, Robert Stärk, *Abstract State Machines.* Springer Verlag, 2003.

4.2 From English to ASM: On the process of deriving a formal specification from a natural language one

Vincenzo Gervasi and Elvinia Riccobene

In this presentation, we traced the model synthesis of an ASM model, starting *literally* from Dijkstra et al.'s text describing an algorithm for distributed protocol termination [2].

In particular, we showed how the description from Dijkstra's paper, if taken at face value, would lead to an inconsistent implementation. We reflect on how a person producing a formal model might be tempted to "correct" the requirements, at times without realising it, in the process of building the model. The net result in such cases is a model which passes verification, but is not validated: thus proving that the wrong system is correct.

4.2.1 Introduction

The first task set by the organisers of Dagstuhl Seminar 13372 on the participants was to build a formal model of a distributed termination algorithm proposed in [2], where it is described in rigorous, but plain English, and try to prove certain desirable properties, such as deadlock-freeness, convergence, etc.

As a first contribution, we decided to focus on the *translation process* (from natural language to a formal model, in this case using the Abstract State Machines formalism), rather than on the *resulting model*. Our aim was to surface instances of *reader fill-in*, i.e. when the specifier, rather than faithfully translating the author's description to a more precise language, tends to fill in missing details, or make assumptions that are not explicitly stated in the source text, or force his or her particular interpretation on the author's words.

Often, this phenomenon is desirable: in that a competent specifier, knowing that the author's intention *must have been* to specify a correct algorithm, is justified in establishing any missing assumption or in choosing that particular interpretation which makes the formal model of the system correct. However, in many other cases there is a huge risk associated to this behaviour, in particular where the subject matter is not "pure" computer science (about which a specifier will probably know much), but some other domain (about which a specifier might well know little).

Our method has thus been to translate as faithfully as possible Dijkstra's English to Abstract State Machines rules, and then reflect on the shortcomings and problems with such a model.

4.2.2 Building a model

In our presentation, selected excerpts from [2] are presented side-by-side their ASM counterpart. We will consider the relevant bits of information in textual order, based on their occurrence in the paper, and later use the composition features of ASMs to collate all the rules in a single coherent model.

We start with the definition of machines and their state

We consider N machines, each of which is either active or passive.	$Machines \equiv \{m_0, ..., m_{N-1}\}$ $state : Machines \to \{active, passive\}$

The next fragment tells us that message transmission is instantaneous: that is, as soon as a machine sends a message (which we define as a macro), the message is available for the recipient to process. Notice we need not formalise the restriction that only active machines send messages (we could, for example, use an **if** clause) as we interpret this as a property that has to be verified (namely, that a non-active machine will not attempt to SendMessage).

Only active machines send messages to other machines; message transmission is considered instantaneous.	$\textsc{SendMessage}(dest) \equiv$ $\quad hasMessage(dest) := true$

Machines switch from active to passive state and vice versa according to the following two rules:

After having received a message, a machine is active. The receipt of a message is the only mechanism that triggers for a passive machine its transition to activity.	MACHINE $\equiv$ ... **if** $hasMessage(self)$ $state(self) := active$ ...
For each machine, the transition from the active to the passive state may occur "spontaneously".	MACHINE $\equiv$ ... **choose** $spontaneous$ **in** $\{true, false\}$ **do** **if** $state(self) = active$ **and** $spontaneous$ $state(self) := passive$ ...

Again we have above a negative clause, that we can verify by inspecting our machine to check that no other assignment $state(.) := active$ occurs beyond the one above. Notice that it is not clear from the text whether being passive is a prerequisite for reception of a message (probably not); this might potentially be a property for verification. On the contrary, the spontaneous transition from active to passive can only happen if a machine is in the active state.

The text continues by making it explicit what a stable state is in such a system; we model the same concept with a derived function *stable*.

From the above it follows that the state in which all machines are passive is stable.	$stable \equiv$ **forall** m **in** $machines, state(m) = passive$

The main property for the correctness of the algorithm (namely: that it does indeed detect termination) is stated as follows, and does not correspond to explicit ASM rules (but rather, to a condition that has to be proved):

The purpose of the algorithm to be designed is to enable one of the machines, machine nr. 0 say, to detect that this stable state has been reached.	How a machine can "detect" termination will be defined later.

The paper proceeds by stating initial conditions, both in terms of initial states of the various machines, and of topology (of which, we consider only the *token ring* scheme that is used afterwards in the paper).

It is furthermore required that the detection algorithm can cope with any distribution of the activity at the moment machine nr. 0 initiates the detection algorithm.	INIT $\equiv$ $master = m_0$ **forall** m **in** $Machines$ **do** $state(m) :=$**choose in** $\{active, passive\}$
Two orderly configurations present themselves [...] the N machines arranged in a ring.	$pred, succ : Machines \to Machines$ $pred(m_i) \equiv m_{(i-1) \bmod N}$ $succ(m_i) \equiv m_{(i+1) \bmod N}$

Finally, the paper introduces the *token* that is passed, according to the ring topology, from machine to machine – and which will be a determining factor in detecting termination.

We assume the availability of communication facilities such that [...]	PassToken $\equiv$
	  $hasToken(self) := false$
	  $hasToken(pred(self)) := true$

The token being returned to machine nr. 0 will be an essential component of the justification that all machines are passive.	Machine $\equiv$
	  ...
	  if $(self = master)$ **and** $tokenIndicatesFinished$
	    $done := true$

We must thus prove, $done \implies stable$. Notice that *stable* could hold even if the *master* (that is, machine nr. 0) hasn't noticed it yet, so $stable \implies done$ is not needed, nor requested.

Finally, the paper describes various "rules", that are in effect the algorithm itself. Rule 0 describes the circumstances under which the token is passed around.

Rule 0. When active, machine nr. i+1 keeps the token; when passive, it hands over the token to machine nr. i	Rule$_0$ $\equiv$
	  if $hasToken(self)$
	    if $state(self) = active$
	      skip
	    else
	      PassToken

Next to be taken into account is the possibility of messages being sent: [...]	Machine $\equiv$
	  ...
	  choose *spontaneous* **in** $\{true, false\}$ **do**
	    if $state(self) = active$ **and** *spontaneous*
	      choose *dest* **in** *machines* **do**
	        SendMessage$(dest)$

To this end each machine is postulated to be either black or white.	$color_M : Machines \rightarrow \{black, white\}$

Notice that in the first fragment, the condition **if** $hasToken(self)$ is implied by the use of the verb "keeps" in the original (but it would be wise, in a real case, to inquire with the author of the original text).

The next rule details how machines change their colour, and introduce the token's colour:

Rule 1. A machine sending a message to a recipient with a number higher than its own makes itself black.	SendMessage$(dest)$ $\equiv$
	  ...
	  if $(dest >_M self)$ /* relaxed in Rule 1' */
	    $color_M(self) := black$

To this end the token is postulated to be either black or white.	$color_T :\rightarrow \{black, white\}$

Coloring of the token is the subject of rule 3, where the concept of a *probe* is also introduced.

Rule 2. When machine nr. i+1 propagates the probe, it hands over a black token to machine nr. i if it is black itself, whereas while being white it leaves the colour of the token unchanged.	PASSTOKEN $\equiv$ $\ldots$ **if** $color_M(self) = black$ $color_T := black$
[...] unsuccessful probe: when a black token is returned to machine nr. 0 or the token is returned to a black machine nr. 0, the conclusion of termination cannot be drawn.	$unsuccessful \equiv$ $self = master \wedge hasToken(self) \wedge (color_T = black \vee color_M(self) = black)$

We will model rules 3 and 4 together, as they jointly describe how a probe is initiated.

Rule 3. After the completion of an unsuccessful probe, machine nr. 0 initiates the next probe. **Rule 4.** Machine nr. 0 initiates the probe by making itself white and sending a white token to machine nr. N-1	$\text{RULE}_{3+4} \equiv$ **if** $(self = master)$ **and** $unsuccessful$ $color_M(self) := white$ $color_T := white$ PASSTOKEN

The last rule details how passing the token changes the colour of a machine:

Rule 5. Upon transmission of the token to machine nr. i, machine nr. i+1 becomes white. (Note that its original colour may have influenced the colour of the token).	PASSTOKEN $\equiv$ $\ldots$ $color_M(self) := white$

The final specification was obtained by collating all the fragments we have described above, and adding some additional initialisation to assign initial colours to the various machines and the token, to create ASM agents and assign their programs so that each would execute (in a truly distributed, concurrent fashion) the behaviour specified for the algorithm's machines. Space considerations prevent us from listing here the final specification, and its CoreASM [4] executable equivalent (which, anyway, differs only in minor syntactic ways from what we have shown above).

4.2.3 Problems with the model

We purposely ignored many problems in the direct translation, adhering thus to the position advocated by Berry in [**?**] than an "ignorant" mindset facilitates exposing errors in translating from requirements to formal models.

In fact, our directly translated model has exposed several problems and assumptions[1], that we can only briefly list here:

1. RULE_2 and RULE_{3+4} might be executed in parallel, causing a conflicting update to $color_T$.
2. The paper does not explicitly specify when or if a message is consumed. We can infer that it will be consumed upon the machine transitioning to *active*, by adding

[1] Some of these have been revealed by direct inspection of the model, others have been exposed in simulating the execution in CoreASM.

$hasMessage(self) = false$ after $state(self) := active$, but that might cause additional problems (namely, if in the same step a machine m transitions to active and another machine m' sends a message to m, we would have an inconsistent update to $hasMessage(m)$).

3. The "spontaneous" transition from *active* to *passive* needs to be restricted by additional conditions, not mentioned in the paper. Namely, if in a step a machine that is already *active* receives a new message and spontaneously transitions to *passive*, we would have again an inconsistent update.
4. Our macros SENDMESSAGE and PASSTOKEN may cause an inconsistent update to $color_M(m)$ if m "spontaneously" decides to send a message to any other machine m' in the same step in which it is passing the token to the next machine in the loop.
5. Multiple agents could send messages to the same machine in the same step (true concurrency). We do not know if multiple messages gets collapsed into a single activation, or enqueued and then slowly consumed in subsequent steps, etc.
6. Messages in transit: while message passing is instantaneous, activation of machines might be not. In other words, it may happen that a message is delivered to a machine immediately (i.e.: in the next immediate step) after being sent, but the agent executing that machine will react to the message with a certain delay. The algorithm is correct if activation is also instantaneous, but while the text explicitly states that message delivery is instantaneous, it does not specify that machines are infinitely fast[2].

It will be apparent that many of these problematic issues are exposed by the fully parallel and concurrent nature of distributed ASMs. Any purely sequential or interleaving model would not reveal many of these problems, and would not be faithful to the kind of distributed system imagined by the authors of the original text.

4.2.4 Conclusions

Our intention for the first session of the Seminar was to focus on using ASMs for validation purposes, rather than for verification. Indeed, our attempt at direct translation of the English text from [2] has exposed a number of issues which, if only it were still possible, we would have liked to discuss with the author of our requirements, in a typical instance of the virtuous circle where difficulties with the formal specification prompt further elicitation of the requirements.

After the Seminar we became aware that an ASM model for the same algorithm, completed with a formal verification, had already been published well over a decade ago [3]. In that work, the authors stress mathematical verification, relying on an informal justification for the validation part. We intend to compare and contrast our approach to validation with the models built for verification in [3] as part of future work.

References

1 D. Berry. The importance of ignorance in requirements engineering. *Journal of Systems and Software*, 28(1):179–184, 1995.

2 E. W. Dijkstra, W. Feijen, and A. van Gasteren. Derivation of a termination detection algorithm for distributed computations. *Information Processing Letters*, 16(5):217–219, 1983.

[2] We believe that any concurrent model where the *sender* of a message alters the internal state of the *receiver* in the same atomic action is not representing faithfully the distributed nature of the system.

3 R. Eschbach. A termination detection algorithm: specification and verification. In J. Wing, J. Woodcock, and J. Davies, editors, *FM'99 – Formal Methods*, volume 1709 of *Lecture Notes in Computer Science*, pages 1720–1737. Springer Berlin Heidelberg, 1999.
4 R. Farahbod, V. Gervasi, and U. Glässer. CoreASM: An extensible ASM execution engine. *Fundamenta Informaticae*, 77:71–103, Mar./Apr. 2007.

4.3 Modelling and Validation of the Dijkstra's Case Study through the ASMETA Framework

Paolo Arcaini and Angelo Gargantini

We present the modelling and validation of the case study *Derivation of a termination detection algorithm for distributed computations* by Dijkstra [1], through the ASMETA framework [2]. We have implemented in AsmetaL [3], a concrete syntax for ASMs, the high-level model of the case study presented in the abstract of Gervasi-Riccobene and reflecting "literally" the requirements of the algorithm. By simulation and model review [7] of the AsmetaL model, we have highlighted some ambiguities of the requirements. Model review, in particular, has proved to be an affective *push-button* validation technique. The aim of model review is to determine if a model is of sufficient quality; it allows to identify defects early in the system development, reducing the cost of fixing them. The *AsmetaMA* tool [4] permits to perform *automatic* review of ASMs; it looks for typical vulnerabilities and defects a developer can introduce during the modelling activity using the ASMs.

The application of model review to the AsmetaL model has permitted to identify several inconsistent updates that could rise during simulation. Such inconsistent updates derived from simultaneous behaviours that were not properly guarded in the model.

References

1 E. W. Dijkstra, W. Feijen, and A. van Gasteren. Derivation of a termination detection algorithm for distributed computations. *Information Processing Letters*, 16(5):217–219, 1983.
2 The ASMETA website. http://asmeta.sourceforge.net/, 2013.
3 A. Gargantini, E. Riccobene, and P. Scandurra. A Metamodel-based Language and a Simulation Engine for Abstract State Machines. *J. Universal Computer Science*, 14(12):1949–1983, 2008.
4 P. Arcaini, A. Gargantini, and E. Riccobene. Automatic Review of Abstract State Machines by Meta Property Verification. In C. Muñoz, editor, *Proceedings of the Second NASA Formal Methods Symposium (NFM 2010)*, pages 4–13. NASA, 2010.

4.4 A VDM model of the Termination Detection Protocol

Jens Bendisposto, Egon Börger, Ian J. Hayes, Peter Gorm Larsen, and Andreas Prinz

It was tried to model the system in two ways. In a first attempt a functional specification style was attempted. This did not produce a satisfactory result due to the non-determinism in the protocol that lead to a too complex model. In a second attempt a state-based approach was followed which produced a satisfactory model, in particular, dealing well with the non-determinism of the protocol.

4.5 B and Event-B Models for Distributed Termination

David Deharbe, Marc Frappier, Thierry Lecomte, Michael Leuschel, and Laurent Voisin

In the talk we presented the results of our working group on developing B and Event-B versions of Dijkstra's distributed termination algorithm. In the working group, we first decided upon the global system view and the events of the systems, and then gradually defined the events and introduced variables as needed. Refinement and invariants were added later. Then an Event-B model in Rodin and a classical B model were developed in parallel. In the end, we managed to prove the invariants of the Event-B model using Rodin and the invariants of the classical B model using Atelier-B. We used ProB to validate the absence of deadlocks and various LTL properties. The proper functioning of the models was checked by graphical animation using BMotionStudio. Various errors were uncovered by both model checking and animation.

4.6 Modeling Dijkstra's Termination Detection Algorithm in TLA$^+$

Leo Freitas, Stefan Hallerstede, Dominik Hansen, Markus A. Kuppe, Fernando Mejia, Stephan Merz, Hernán Vanzetto, and Kirsten Winter

TLA$^+$ [4] is a formal specification language that is mainly intended for distributed algorithms. It is based on untyped Zermelo-Frankel set theory for modelling the data structures and on the Temporal Logic of Actions for specifying the possible executions. Typically, TLA$^+$ specifications define transition systems, represented by a state predicate defining the initial states, a transition predicate delimiting the next-state relation, and fairness conditions.

On the first day of the seminar, we modelled Dijkstra's termination detection algorithm [2] in TLA$^+$. We started by determining the representation of the system state by two variables indicating the state of every node (its status can be active or passive, its colour can be white or black) and of the token (its current position and its colour). The definitions of the initial condition and of the next-state relation follow quite directly Dijkstra's description. Having specified the state we formalised the behaviour of the system in terms of a predicative transition relation. We found that a discussion of the formal specification of the algorithm helped us clarify our understanding of it and solve problems in a first version without even resorting to formal verification. Interesting properties of the protocol were then specified in temporal logic.

We also noted a distinction in modelling style: some group members prefer directly giving definitions of algorithm operations (represented as TLA$^+$ actions), whereas others first think about state invariants. Arguably, the former is more dynamic and enables for testing the model using automated tools, whereas the latter builds a solid understanding of the reasons for the algorithm being correct. In practice, it is hard to clearly differentiate this question of style from a (perhaps unconscious) bias towards the functioning of the tools associated with the method.

Tool support for TLA$^+$ comes through the TLA$^+$ Toolbox, an IDE for developing and analysing TLA$^+$ specifications. In particular, the model checker TLC confirmed that the previously defined correctness properties hold for fixed finite instances of the algorithm. In particular, all nodes are indeed inactive when the algorithm detects termination, and the algorithm will eventually detect termination when all nodes are inactive.

The TLA$^+$ proof system TLAPS [?] provides assistance for carrying out interactive proofs and verifying properties of TLA$^+$ specifications. TLA$^+$ contains a hierarchical language for writing proofs that TLAPS checks with the help of different back-end provers, including the tableau prover Zenon, an encoding of TLA$^+$ in the logical framework Isabelle, and a generic interface to SMT solvers. We first used TLAPS to prove a simple invariant that expresses type correctness. Dijkstra's paper also states an invariant that should be satisfied by the algorithm. To our surprise we were at first unable to prove that invariant, and in fact it did not hold: we had developed a variant of Dijkstra's algorithm. We were able to restate the invariant and achieve a correctness proof of the algorithm with the help of the modified invariant. Liveness of the algorithm cannot currently be proved with the help of TLAPS, as it does not yet support proofs in temporal logic.

The TLA$^+$ model of the algorithm can be translated to B [?], and this enables the analysis and visualization of the model using ProB [5]. It would be beneficial to integrate this translation and ProB in the TLA$^+$ Toolbox.

References

1 D. Cousineau, D. Doligez, L. Lamport, S. Merz, D. Ricketts, H. Vanzetto. TLA$^+$ Proofs. 18th Int'l Symp. Formal Methods (FM'12), LNCS, Vol. 7436, pp. 147-154, Springer, 2012.

2 E. W. Dijkstra, W. H. J. Feijen, A. J. M. van Gasteren. Derivation of a termination detection algorithm for distributed computations. Inf. Proc. Letters 16:217-219 (1983).

3 D. Hansen, M. Leuschel. Translating TLA$^+$ to B for Validation with ProB. J. Derrick, S. Gnesi, D. Latella, H. Treharne (eds.): 9th Intl. Conf. Integrated Formal Methods (iFM 2012). Springer LNCS 7321, pp. 24-38. Pisa, Italy, 2012.

4 L. Lamport. Specifying Systems. Addison Wesley. Boston, Mass., 2012. See also http://research.microsoft.com/en-us/um/people/lamport/tla/tla.html.

5 M. Leuschel, M. J. Butler. ProB: an automated analysis toolset for the B method. STTT 10(2):185-203 (2008).

5 Working Groups: Automated Teller Machine (ATM)

5.1 The Integrated Use of Event-B and ASM via UML-B

Paolo Arcaini, Elvinia Riccobene, and Colin F. Snook

In this talk, we discussed the possibility to integrate the use of Event-B and ASMs (Abstract State Machines) for model development and analysis. The two methods provide different models of computation and methods of analysis, so they can be applied for different verification and validation purposes. The "Cash-point Service" case study was used to show the feasibility of the integrated use of the two formal methods. We discussed the validity of the UML-B representation as a common basis for both target notations, and an UML-B model of the case study was developed in terms of class diagrams and hierarchical state machines. Then parallel Event-B and ASM models were derived, even if not in a complete way, from the UML-B model for different analysis purposes: correct refinement proofs could be achieved by exploiting the Rodin tool support [1] for proving Event-B models; model simulation, model review and model checking could be performed by using the ASMETA toolset [2] for ASM models. The future intention is to automatise the derivation of ASM and Event-B models from UML-B, and to ensure that the Event-B and ASM models are equivalent despite differences in the semantics of the notations. We also want to experiment further advantages coming from the integrated use of the analysis techniques of the two methods.

References

1 The RODIN website. http://www.event-b.org/, 2013.
2 The ASMETA website. http://asmeta.sourceforge.net/, 2013.

5.2 Specification of the Cash Dispenser in Parallel using ASM and VDM

Cliff B. Jones, Peter Gorm Larsen, Andreas Prinz, Alexander Raschke, and Colin F. Snook

5.2.1 On the VDM model

The purpose of our first model is to make a quick analysis of the main functional requirements, essentially ignoring concurrency issues. Thus, this model describes aspects of the central resource and only one till. Once the basic functionality is better understood we start to consider multiple tills. The model is described in VDM-SL as a short, flat specification. This enables abstraction from design considerations and ensures maximum focus on high-level, precise and systematic analysis.

We develop an abstract model in VDM-SL in order to clarify and analyse requirements. This model is validated using a testing approach. The purpose of our models is to analyse the main functional aspects of the system, including security aspects related to misuse of cards. For example, the models treat the daily limit policy, the validation of PIN codes, limits on PIN code attacks, and reports of illegal (e.g. stolen) cards. However, we abstract away from, for example, the choice of concrete databases, communication protocols and encoding of PIN codes.

5.2.2 Comparison

Some "correspondences" are easy to spot:

VDM	ASM
state definition explicit	implied/extracted
state components	"controlled functions"
invariants	tool dependant
pre	written as if statements
post condition (not used in this case)	only explicit
records	via (selector) functions
not applicable; records are values	record creation is bound to be a step
pattern match on records	not available
other "sophisticated" data types	sets/sequences
doesn't have a global let	"derived" functions

ASM and VDM have very different concurrency principles

- In VDM++
  - explicit synchronization
  - Using mutex and permission predicates
  - Closer to implementation language
  - Interleaving concurrency
- In ASM
  - More high-level "magic"
  - Potential problem that implementer "forgets" synchronisation
  - Convenient abstraction for design
  - True concurrency

5.3 Specification of the Cash Dispenser in Parallel using Z and TLA$^+$

Leo Freitas

TLA$^+$focuses on operations, Z on state. Concerning the use of TLA$^+$actions and Z schemas modelling styles are similar.

5.3.1 Differences

- Outputs are handled differently. TLA$^+$uses "state" variables. Z uses schema variables marked with a "!".
- Z preconditions aren't guards.
- Z has complex types yet has untyped logic. TLA$^+$is untyped.
- Updates in Z via promotion, that is local and global. Updates in TLA$^+$are global.
- Z schema calculus
- Verification methods: TLA$^+$uses model checking and proof whereas Z uses proof only (ZEves)

5.4 Specification of the Cash Dispenser in Parallel using ASM and TLA$^+$

Egon Börger, Ian J. Hayes, Fernando Mejia, Stephan Merz, Klaus-Dieter Schewe, and Kirsten Winter

5.4.1 Desiderata

The produced models had to satisfy the following desiderata:

- component-based modelling
- replicate independent ATM components

- focus on observable ATM behaviour
- obtain as similar models as possible
- don't aim for complete specification

5.4.2 Observations

We have made the following observations concerning the two modelling approaches:

- non-interleaving, compositional specifications not supported by TLA$^+$tools
- behaviour specifications are very similar (control ASMs, state machines in TLA$^+$)
- properties stated outside the ASM language (but supported by ASM tools)
- translation from ASM to TLA$^+$should be straightforward

5.5 Specification of the Cash Dispenser in Parallel using ASM and B

Marcel Dausend, Marc Frappier, and Angelo Gargantini

We have modelled the "Cash-point" Service proposed by the organisers in collaboration and in parallel using both methods, Abstract State Machines (the tool Asmeta) and classical B with Atelier B and ProB.

The goal has been twofold. One objective has been to create two models of the case study as a basis for comparison. The subsequent objective was to compare the models' notations and the process of developing those models. For this reason, we have decided to proceed incrementally, keeping the names of variables, events and rules as close as possible in the two models. In order to guarantee that both models capture the same behaviour, we systematically performed simulation and comparison of the two models. In this way the ASM people could learn some details of the syntax and semantics of B and vice-versa.

We were able to identify some shared characteristics between ASM and B constructs. B sets and their properties are mapped to ASM (abstract) domains and their initial elements. B states correspond to (dynamic controlled) functions and variables of ASMs. Constants can be easily represented in both notations. Refinement has been used in the initialisation in B to guide ProB for the simulation of the nondeterministic abstract initialisation. In ASMs it has been done by function definition and state initialisation. ASM rules and B operations specify how the state evolves. We could map every B operation to one ASM rule. Preconditions of B operations are mapped in ASMs to guards of conditional rules, while B substitutions are mapped to ASM function updates. Both formalisms allow explicit use of variables set by the environment. These variables are normally user inputs during simulation. In B user inputs are parameters in operations, while in ASM they are modelled as monitored functions. In B all the operations are enabled (if their preconditions are true), while in ASM (at least in the used tool) the user has to write a main rule that explicitly runs all the rules in parallel.

6 Working Groups: Comparison of Methods and Tools

6.1 Summary of Meeting of Workgroup on Methodology

Albert Fleischmann, Uwe Glässer, Ian J. Hayes, Stefan Hallerstede, Dominik Hansen, Laurent Voisin, and Kirsten Winter

The discussion in the meeting proceeded in two stages. In the first stage we tried to delineate our understanding of modelling and its purpose as it would be understood by the different communities present at the seminar.

6.1.1 Stage 1

The purpose of modelling is System Engineering rather than (pure) Software Engineering. This position has consequences on the scope of problems to which modelling is applied ranging from (1) system requirements analysis to (2) system specification and verification to (3) concrete systems and software exploitation.

The 3 problem domains have different techniques associate with them. Usually tools do not focus on more than one domain. The focus on different domains may explain some differences in notation. (But often notations seem to evolve haphazard driven by the problems they were first applied to.)

Concerning (1) common techniques are
- (partial) model building
- traceability to requirements, needs and objectives
- animation
- simulation
- integrity analysis (details vary)

Concerning (2) common techniques are
- safety/liveness analysis
- traceability to requirements and statements in the formal model
- model checking, theorem proving, refinement, abstraction
- simulation

Concerning (3) common techniques are
- (program) execution
- model-based texting
- runtime verification
- traceability to requirements and statements in the formal model

Given the variation in the application domains of the different methods and tools it does not appear reasonable to attempt to unify them. But it would be useful to find ways to achieve tight collaboration in order to address problems that span several of the domains.

6.1.2 Stage 2

In the second stage we formulated some recommendations that could help achieving such collaborations.

6.1.2.1 Recommendation A

"Compatible" specification notation dialects shall be used that permit switching between methods and tools during development. The meaning of "compatible" is not well-understood. A possible meaning could be "only requires simple translation". This would coincide with the finding of the working group on abstract syntax. The different methods should co-operate in particular with respect to important modelling concepts such as concurrency and distribution.

6.1.2.2 Recommendation B

Transfer of concepts between methods should be encouraged, e.g.,
- modules from ASM to Event-B
- refinement from Event-B to ASM

6.1.2.3 Recommendation C

Challenges could be put forward resulting from the transfer, e.g.,
- use ASM for composing Event-B modules
- use Event-B to prove certain ASM refinements

6.2 Comparison of Methods

Peter Gorm Larsen, Andreas Prinz, Colin F. Snook, and Hamed Yaghoubi Shahir

In the discussion, we agreed that all the methods are state-based and have a semantically similar idea of state. The same is true for the initial state. Moreover, all the methods use some kind of state transitions. There is not much alignment related to the notation of concepts and the (detailed) semantics of concepts. This leads to problems with different semantics for visually the same concepts and the same semantics for visually different concepts. The main difference is often the development process used, and the example we had showed this clearly. In our group, we worked with functional formulation first, which did not work out too well. Afterwards, a more relational approach was followed making our solution more similar to the other solutions.

6.3 Defining ASMs as Event-B Machines

Egon Börger and Laurent Voisin

6.3.1 Abstract State Machines

A *(sequential) ASM M* is a set of rules of form

$$\textbf{if } \mathit{cond} \textbf{ then } \mathit{Updates}$$

where *cond* is a first-order (typically set theory) formula and *Updates* is a set of function updates $f(t_1, \ldots, t_n) := t$ with parameterised locations (array variables) $(f, (val_1, \ldots, val_n))$ and terms $t_1, \ldots, t_n, t$. It is allowed to **choose** among or to generalise **forall** elements of a set in a rule. Also other usual notations like **if then else**, **let**, etc. and calling (recursive) sub-machines are allowed.

The states of an ASM are (Tarski) structures with static and dynamic part. In each *step* M executes all its rules, performing simultaneously all the updates of each of its rules whose *cond* in this state is true.

Input is taken as part of the initialisation of a machine M and after each machine step by a step of the *environment* which updates the so-called monitored locations of the machine state. For shared locations a protocol is required to avoid conflicting updates.

An *(async) ASM* is a set of agents each equipped with a sequential ASM.

6.3.2 Event-B Machines

Except for notational differences Event-B machines have the same states as ASMs, to be precise structures of a given signature with a static part (called 'context') of sets s ('universes'), constants c, properties(c, s) ('axioms' and 'theorems' to be proved within the context) and a dynamic part of (pairwise distinct) variables v, invariants (predicates the variables must be proved to satisfy in every state that is reachable from an initial state) and the environment which is part of the model (closed system).

Inputting is done by non-determinacy. The initialisation is described via a special event with guard true.

Events are without loss of generality instances of the following general form (stated in ASM notation):

if **forsome** p $P(p, v)$ **then**
  choose p **with** $P(p, v)$ **in** $action(p, v)$
where
  $action(p, v) =$ **choose** v' **with** $Q(v, v', p)$
    $v := v'$ // NB.v, v' generally are vectors

A step of an Event-B machine with a finite number of events $event_i$ $(1 \leq i \leq n)$ can be described in ASM terms as follows:

choose i **with** $guard_i$ **in** $event_i$ // interleaving view
  where
    $event_i =$ **if** $guard_i$ **then** $rule_i$

A constraint requires that no parallel update is allowed for the same variable.

6.3.3 Sequential ASMs encoded as Event-B Machines

To simplify the exposition of the coding idea assume that each **choose** construct is replaced by an explicit selection function.

A brute force transformation of (sequential) ASMs rewrites the set of parallel rules in such a way that all truth-functional combinations of rule conditions are listed explicitly and guard the respective function updates. This comes up to rewrite the given ASM into a flat ASM with only rules of form **if** *cond* **then** *Updates* where all guards are pairwise disjoint. As a consequence, one can interpret the resulting ASM semantically as an Event-B machine where in each state only one 'event' can be selected. All non-determinism is hidden in the selection functions.

- replace

  if $cond_1$ **then** $Updates_1$
  if $cond_2$ **then** $Updates_2$

- by the equivalent machine

  if $cond_1$ **and** $cond_2$ **then**
  $Updates_1$
  $Updates_2$
  if $cond_1$ **and not** $cond_2$ **then** $Updates_1$
  if $cond_2$ **and not** $cond_1$ **then** $Updates_2$

A more economical encoding goes as follows:

if c_1 **then** $f(s_1) := t_1$
if c_2 **then** $f(s_2) := t_2$

is encoded by

when c_1 **or** c_2 **then**
$f := f \oplus$
$\{(a, v) \mid c_1$ **and** $(a, v) = (s_1, t_1)$
or c_2 **and** $(a, v) = (s_2, t_2)\}$

To prove: f is a (partial) function.

Similarly the **forall** construct can be encoded via a set update.

We still have to clarify whether and how to deal with recursive sub-machines and with the **undef** concept in ASMs.

6.3.4 Where the differences are

The main difference is in the refinement notions (see [1] and [2, Ch.3.2.1]). The Event-B concept is tailored to provide as much as possible interaction-free refinement correctness verifications. The ASM refinement concept is driven first of all by modelling concerns to describe design ideas in an accurate and as smooth as possible manner, delegating mathematical verification (where considered to be needed) to a second step.

References

1 J.-R. Abrial. *Modeling in Event-B: System and Software Engineering.* Cambridge University Press, Cambridge, 2010.
2 E. Börger and R. F. Stärk. *Abstract State Machines. A Method for High-Level System Design and Analysis.* Springer, 2003.

6.4 Closing the Gap between Business Process Models and their Implementation: Towards Certified BPMs

Albert Fleischmann and Egon Börger

The gap between on the one side the users' understanding of Business Process Models (BPMs), even if described using standardised languages like BPMN, and on the other side the run behaviour of model implementations is still with us. We explain how Abstract State Machines (ASMs), tailored as a domain specific (to a large extent diagrammatic) modelling language, allow the BP experts to design BPMs with the help of a graphical editor in such a way that the underlying ASM models constitute a reliable precise contract – a contract which guarantees to the BP domain experts that the application-domain focussed understanding of the BPMs they design is also a correct understanding of the code behaviour provided by the implementation of the models by software experts. This opens the way to the development of certifiably correct BPMs and their implementations. We instantiate the claim by ASM models for the behavioural meaning of the graphical notations used in Metasonic's industrial BPM tool suite.

6.5 Comparison of Methods and Tools: the Report of the ASM Group

Elvinia Riccobene, Paolo Arcaini, Marcel Dausend, Albert Fleischmann, Angelo Gargantini, Vincenzo Gervasi, Uwe Glässer, Alexander Raschke, Gerhard Schellhorn, Klaus-Dieter Schewe, Qing Wang, and Hamed Yaghoubi Shahir

This report presents the results of the ASM group discussion, held at the second day of the seminar, regarding the comparison of methods and tools on specific questions and reflection points.

The following questions were arguments of the discussion:

- Why were there variations of the models? Why did you go for one model rather than another? Which modelling process did you use?
- Compare with other models you have seen: Commonalities and Differences.
- List weak points of your method, tools, models.
- How far is the model style pre-determined by the tool or adapted to the tool?
- Wish list for new features, tools, etc.; especially those you have seen from other groups.
- Brainstorm: Ideas and suggestions for Integrating tools and methods; make more concrete suggestions, how they would profit, refer to other tools and methods.

For the Dijkstra et al.'s case study, three ASM models were presented. They all *share* the common operational semantics of the ASMs, but present some *variations* due to the different goals which they were developed for. (i) For *requirement elicitation*, an ASM model was developed simply translating the text of the Dijkstra description in terms of transition rules capturing the behaviour of the system at a very high level of abstraction. This model is not "correct" and "complete". Rather, it tries on purpose to expose errors, ambiguities, or incompleteness in the original text. Correctness and completeness can be reached through an iterative process reasoning on requirements. This high level model capturing the informal requirements is usually called "ground model". The ASM ground model of the Dijkstra et al.'s case study results into an asynchronous multi-agent ASM. (ii) For *model validation*, two ASM models were developed and encoded in the languages of the two simulation engines available for ASMs: the CoreASM and the AsmetaS simulator of the ASMETA framework. These two models can be obtained from the ground model by refinement into the tool languages. They present distributed solutions with the same transition rules, they almost use the same syntax (untyped for CoreASM, typed for Asmeta), but they present different ways to refine rules in order to make the model executable. (iii) For *verification* purposes, a specification in KIV was developed using a global model to facilitate the proof of invariants.

Regarding the *modelling process* usually followed when developing an ASM model, the user starts from the textual description of the informal requirements and tries to develop a Ground Model capturing the intended behaviour of the algorithm or system at a high level of abstraction. Then (s)he refines predicates and macros and makes signature precise and complete. The model can be then validated by the use of simulation engines, and other tools, as the model advisor, can be used to discover inconsistencies, redundancy, etc., helping to fix problems. More complex properties can be later proved by using theorem provers (KIV, PVS) or model checkers (AsmetaSMV, CoreASM2SMV).

Discussing *commonalities and differences* of the ASM method with respect to the other formal methods presented at the seminar, the group agreed on the following *commonalities*: state-based nature of the approach, operational models suitable for animation and verification; and *differences* (commonly intended as advantages of the method): ability to formalise ALL behaviour requirements, and requirements traceability. Both these qualities, not completely present in the other methods, are fundamental to communicate with stakeholders. Among the differences, the necessity was recognised to improve the way to perform model animation that is not supported by the ASM tools at the moment.

As other *weak aspects* of the ASM method and its supporting tools, the group agreed on (i) the absence of a way to express temporal properties (only expressions in CTL/LTL with finite sets are supported); (ii) the lack of integration between different tools for model simulation (CoreASM and Asmeta); (iii) the different ways of data encoding; (iv) the lack of interoperability among encoding into different verification languages (KIV, Promela/SPIN, SMV, PVS).

Regarding the question *if the model style is pre-determined by the tool or adapted to the tool*, the group agreed that modelling style is not constrained by the tools, but a certain adaptation is required to move from the ground model to an executable model, and more effort is required to use verification tools.

On the *wish list* for new features, tools, and other issues seen from other groups, the ASM community agreed on having (i) a higher level notation to facilitate tool integration, and (ii) a way to express temporal features.

On the last question regarding *ideas and suggestions for integrating tools and methods*, all people agreed that is more reasonable and feasible trying to achieve an *integrated use* of different methods and tools, instead of going towards tool integration.

7 Post-Seminar Collaboration Activities

In the sessions on day 3 and day 4 many concrete collaborations and integration efforts have been started. In particular,

- Visualisation & Integration: Jens, Peter, Vincenzo
- UML-B to ASM: Elvinia, Paolo, Colin
- Constraints Solver Integration into VDM: Jens, Peter
- Pro-B Integration into TLA Toolbox and TLC Tool: Dominik, Stephan, Hernan
- CZT to Pro-B (Pro-Z + TLA$^+$): Leo
- Proof GUI: Laurent, Peter
- Destecs Integration: Peter, Jens
- Comparison of Model Checking between ASM and B: Paolo, Michael

- Comparison of Model Development between ASM and B: Marc, Angelo
- Common Language for ASM: Marcel, Alex, Vincenzo, Elvinia, Angelo, Andreas
- Case Study: Uwe, Hamed
- Modelling and Validation of Distributed Situation Analysis (ASM and Event-B): Uwe, Narek, Hamed
- ASM to Event-B Translation: Laurent, Egon

8 Plenum Discussions

The organisers had planned several plenary discussions in a first draft. They had been replaced by plenum discussions by the time the seminar started. The main reason for this was to avoid turning too many participants into "spectators". We believe, this worked well for a small seminar. Plenum discussions may be less effective in large seminars.

In this seminar a plenum discussion was held each time after the groups reunited to present their results. If the time planned for the discussion was not sufficient, the corresponding slot was extended and the programme adapted.

9 Final Programme

Day 1

09:00–09:30	Opening
09:30–10:30	Short presentation of participants (3 slides per presentations, all presentations in 1 hour!)
10:30–12:00	Modelling case study in groups
12:15–13:30	Lunch
13:30–15:30	Modelling case study in groups
15:30–16:00	Coffee break
16:00–18:00	Presentation of the models to illustrate the different methods and tools

Day 2

09:00–10:30	Comparison of methods and tools based on the models of day 1
11:00–12:00	Presentation of results of comparison
12:15–13:30	Lunch
13:30–15:30	Discussion continued (45 min) & 3 working groups on integration ($\sim$ 60 min): Methodology, Abstract Syntax, Low-level integration
15:30–16:00	Coffee break
16:00–17:30	Discussion of working group results
17:30–18:00	Talk: Egon Börger: A proposal for including communication into Abstract State Machines

Day 3

09:00–10:30	Modelling in two formalisms: The FM'99 ATM modelling challenge
11:00–11:45	Talks: Paolo Arcaini & Kirsten Winter
11:45–12:15	Planning of integration sessions for next day
12:15–13:30	Lunch
13:30–17:30	Excursion: A rainy walk

Day 4

09:00–10:00	Wrap up ATM case study of day 3
10:00–12:00	Presentation of solutions of the ATM case study
12:00–12:15	ASM to B: transformation (Egon Börger)
12:15–13:30	Lunch
13:30–15:00	Research and tool talks on various topics (by Kirsten Winter, Laurent Voisin, Marcel Dausend, Leo Freitas)
15:30–17:30	Technical interchange meetings and discussion about architectures between individual groups
17:30–18:00	Talk: Flash File System Verification (Gerhard Schellhorn)
19:30–21:00	Plenum discussion on joint book and other project outcomes (report for Seminar 13372)

Day 5

9:00–11:00	Closing: Conclusions, Proceedings, Book

The rest of the day was left unplanned because many participants left early due to travel arrangements.

Participants

- Paolo Arcaini
University of Bergamo, IT
- Jens Bendisposto
Heinrich-Heine-Universität Düsseldorf, DE
- Egon Börger
University of Pisa, IT
- Marcel Dausend
Universität Ulm, DE
- David Deharbe
Federal University of Rio Grande do Norte, BR
- Albert Fleischmann
Metasonic AG – Pfaffenhofen, DE
- Marc Frappier
University of Sherbrooke, CA
- Leo Freitas
Newcastle University, GB
- Angelo Gargantini
University of Bergamo, IT
- Vincenzo Gervasi
University of Pisa, IT
- Uwe Glässer
Simon Fraser Univ. – B.C., CA
- Stefan Hallerstede
Aarhus University, DK
- Dominik Hansen
Heinrich-Heine-Universität Düsseldorf, DE
- Ian Hayes
The Univ. of Queensland, AU
- Cliff B. Jones
Newcastle University, GB
- Markus Alexander Kuppe
Universität Hamburg, DE
- Peter Gorm Larsen
Aarhus University, DK
- Thierry Lecomte
CLEARSY – Aix-en-Provence, FR
- Michael Leuschel
Heinrich-Heine-Universität Düsseldorf, DE
- Fernando Mejia
Alstom Transport – Saint-Quen, FR
- Stephan Merz
LORIA – Nancy, FR
- Andreas Prinz
Univ. of Agder – Grimstad, NO
- Alexander Raschke
Universität Ulm, DE
- Elvinia Riccobene
University of Milan, IT
- Gerhard Schellhorn
Universität Augsburg, DE
- Klaus-Dieter Schewe
Software Competence Center – Hagenberg, AT
- Colin F. Snook
University of Southampton, GB
- Hernán Vanzetto
INRIA – Nancy – Grand Est, FR
- Laurent Voisin
SYSTEREL Aix en Provence, FR
- Qing Wang
Australian National Univ., AU
- Kirsten Winter
The Univ. of Queensland, AU
- Hamed Yaghoubi Shahir
Simon Fraser Univ. – B.C., CA

Report from Dagstuhl Seminar 13381

Algorithms and Scheduling Techniques for Exascale Systems

Edited by
Henri Casanova[1], Yves Robert[2], and Uwe Schwiegelshohn[3]

1 **University of Hawaii at Manoa, US,** `henric@hawaii.edu`
2 **ENS – Lyon, FR,** `Yves.Robert@ens-lyon.fr`
3 **TU Dortmund, DE,** `uwe.schwiegelshohn@udo.edu`

Abstract

Exascale systems to be deployed in the near future will come with deep hierarchical parallelism, will exhibit various levels of heterogeneity, will be prone to frequent component failures, and will face tight power consumption constraints. The notion of application performance in these systems becomes multi-criteria, with fault-tolerance and power consumption metrics to be considered in addition to sheer compute speed. As a result, many of the proven algorithmic techniques used in parallel computing for decades will not be effective in Exascale systems unless they are adapted or in some cases radically changed. The Dagstuhl seminar "Algorithms and Scheduling Techniques for Exascale Systems" was aimed at sharing open problems, new results, and prospective ideas broadly connected to the Exascale problem. This report provides a brief executive summary of the seminar and lists all the presented material.

Seminar 15.–20. September, 2013 – www.dagstuhl.de/13381
1998 ACM Subject Classification D.4 Operating Systems, C.1Processor Architectures, F.2 Analysis of Algorithms and Problem Complexity, D.1.3 Concurrent Programming, F.2.2 Nonnumerical Algorithms and Problems, I.2.8 Problem Solving, Control Methods, and Search
Keywords and phrases Exascale computing, high-performance computing and networking, fault-tolerance, power management, scheduling, numerical linear algebra
Digital Object Identifier 10.4230/DagRep.3.9.106

1 Executive Summary

Henri Casanova
Yves Robert
Uwe Schwiegelshohn

Hardware manufacturers are currently deploying machines with sustained petascale performance while already looking forward to produce Exascale machines. Exascale systems are likely to contain 10^5 to 10^6 processors, each processor itself being equipped with more than 100 cores, and possibly 10^3 to 10^4 GPU cores. These systems already reach such a degree of sophistication and complexity that the conventional approach of hardware goes first and applications follow is likely to fail. Furthermore, application performance is no longer solely defined by time-to-solution but also by power consumption and resilience to fault. Many conferences and workshops are dedicated to the architecture and systems issues pertaining to Exascale computing. Instead, in this seminar we have discussed algorithmic issues (application parallelization, application scheduling, resource management, etc.) that must be addresses to make Exascale computing a tenable proposition. As seen in many of the presentations during the seminar, core elements or principles of existing applications must be modified so that they can form the building blocks of new Exascale applications while new methods specifically targeting Exascale systems must be developed for new application areas.

The presentations during the seminar covered a wide range of topics. Some of these topics were directly targeted to various aspects of "the Exascale problem". Some topics were targeted to components of the problem, e.g., efficient execution of application kernels on a heterogeneous many-core node. Finally, yet other topics were in broader, and often more theoretical, parallel and distributed computing contexts with less immediate but possibly large impact on the future of Exascale computing. Overall, the topics presented and discussed during the workshop can be roughly categorized as follows, noting that at least half the presentations spanned more than one of these topics:

- **Fault-tolerance.** Fault-tolerance is a major concern at large scale and several presentations focused on the limitations of current checkpoint-restart fault-tolerance techniques, providing analytical studies to quantify expected performance of these solutions and comparing them to proposed new solutions. These new solutions included, for instance, the use of algorithm-specific checkpointing combined with system-level checkpointing, or the use of imperfect fault predictors.
- **Multi-criteria optimization.** A large number of presentations presented multi-criteria optimization problems, including one traditional performance metric (throughput, makespan) and one (2-criteria) or two (3-criteria) metrics relating to power consumption and/or reliability. Several works studied the use of techniques such as DVFS to trade-off performance for a lower power consumption. These multi-criteria problems were formalized, and various theoretical and practical results were obtained in attempts to solve these problems. Two main approaches were followed: (i) optimizing one metric w.r.t. constraints on the other metric(s); or (ii) obtaining Pareto-optimal solutions or determining the entire Pareto front.
- **Multiple cores.** A handful of presentations focused on the above optimization problems not on large-scale platforms but on many-core nodes with shared memory, i.e., the intended individual components of future Exascale platforms. These nodes consist of possibly heterogeneous cores, accelerators (GPUs, etc.) connected via busses and on-chip networks.
- **Novel scheduling results.** A large number of presentations included novel findings regarding the complexity of scheduling problems. These scheduling problems are of general interest for various models of parallel computation, as motivated by the above topics. Results consisted of p-time optimal algorithms, new NP-completeness results, approximation algorithms, and efficient heuristics.
- **Exascale scientific computing.** Several presentations focused on particular scientific applications (e.g., PDE solvers) or scientific kernels (e.g., matrix multiplication), and discussed how age-old algorithms should be adapted to exploit Exascale platforms with heterogeneous components, hierarchical networks, and the need to have both efficient and rare communication primitive invocations. One presentation presented recent experience with scalable performance monitoring and performance debugging, capabilities that will be crucial in the practice of Exascale computing.
- **Programming models for Exascale.** A handful of presentations spoke to the need for novel programming models at large scale. These presentations spanned the spectrum from very (e.g., actual implementations of programming models usable today, proposals to

enhance current programming standards) to theoretical (e.g., a new theoretical approach to assess the efficiency of techniques such as work stealing and least-loaded-machine-first scheduling when the number of compute nodes tends to infinity).

- **Resource and application management.** A handful of presentations discussed Exascale computing in the context of cloud computing. In other words, these presenters made a case for applying/evolving some of the concepts currently applied in cloud deployments to future Exascale platforms (e.g., service level agreements, virtualization, resource economy). A number of open problems were identified when trying to make these two "worlds" collide.

Although the presentations at the seminar were very diverse in scope, ranging from practice to theory, an interesting observation is that many works do establish strong links between practice (e.g., particular applications, programming models) and theory (e.g., abstract scheduling problems and results). For instance, it was found that the age-old numerical linear algebra topic, far from being well-understood, in fact gives rise to a range of interrelated and interesting practical and theoretical problems that must be solved conjointly to achieve efficiency at large scale. Such observations make it plain that forums that blends practice and theory, as is the case with this seminar, are very much needed.

The seminar brought together 41 researchers from Austria, France, Germany, Ireland, Morocco, Netherlands, New Zealand, Switzerland, U.K, and U.S.A. with interests and expertise in different aspect of parallel and distributed computing. Among participants there was a good mix of senior researchers, junior researchers, postdoctoral researchers, and Ph.D. students. Altogether there were 36 presentations over the 5 days of the seminar, organized in morning and late-afternoon sessions, plus an open problem session. The program was as usual a compromise between allowing sufficient time for participants to present their work, while also providing unstructured periods that were used by participants to pursue ongoing collaborations as well as to foster new ones. The feedback provided by the participants show that the goals of the seminar, namely to circulate new ideas and create new collaborations, were met to a large extent.

The organizers and participants wish to thank the staff and the management of Schloss Dagstuhl for their assistance and support in the arrangement of a very successful and productive event.

2 Table of Contents

3 Overview of Talks

3.1 Adaptive Energy and Throughput Scheduling for Dynamic Systems

Ismail Assayad (University Hassan II – Casablanca, MA)

Joint work of Assayad, Ismail; Girault, Alain

Owing to unpredictable changes of the workload variation, optimally running multiple applications in terms of throughput and energy consumption on multi-core architectures is a huge challenge. We propose a novel adaptive approach capable of handling dynamism of the set of applications. For each such application, a set of non-dominated (Pareto) schedules are computed at design-time. Then, upon the starting or ending of an application, a lightweight adaptive run-time scheduler re-configures the mapping of the live applications according to the available resources (i.e., the available cores of the multi-core chip). This novel scheduling approach is adaptive, and thus able to change the mapping of applications during their execution, under throughput and power-constrained modes. This is a work in progress.

3.2 Approximation algorithms for energy, reliability and makespan optimization problems

Guillaume Aupy (ENS – Lyon, FR)

Joint work of Aupy, Guillaume; Benoit, Anne
Main reference G. Aupy, A. Benoit, "Approximation algorithms for energy, reliability and makespan optimization problems," INRIA Research report RR-8107, October 2012.
URL http://hal.inria.fr/hal-00742754

In this paper, we consider the problem of scheduling an application on a parallel computational platform. The application is a particular task graph, either a linear chain of tasks, or a set of independent tasks. The platform is made of identical processors, whose speed can be dynamically modified. It is also subject to failures: if a processor is slowed down to decrease the energy consumption, it has a higher chance to fail. Therefore, the scheduling problem requires to re-execute or replicate tasks (i.e., execute twice a same task, either on the same processor, or on two distinct processors), in order to increase the reliability. It is a tri-criteria problem: the goal is to minimize the energy consumption, while enforcing a bound on the total execution time (the makespan), and a constraint on the reliability of each task.

Our main contribution is to propose approximation algorithms for these particular classes of task graphs. For linear chains, we design a fully polynomial time approximation scheme. However, we show that there exists no constant factor approximation algorithm for independent tasks, unless P=NP, and we are able in this case to propose an approximation algorithm with a relaxation on the makespan constraint.

3.3 What Makes Affinity-Based Schedulers So Efficient?

Olivier Beaumont (INRIA – Bordeaux, FR)

Joint work of Beaumont, Olivier; Marchal, Loris
Main reference O. Beaumont, L. Marchal, "What Makes Affinity-Based Schedulers So Efficient?," hal-00875487, 2013.
URL http://hal.inria.fr/hal-00875487

The tremendous increase in the size and heterogeneity of supercomputers makes it very difficult to predict the performance of a scheduling algorithm. Therefore, dynamic solutions, where scheduling decisions are made at runtime have overpassed static allocation strategies. The simplicity and efficiency of dynamic schedulers such as Hadoop are a key of the success of the MapReduce framework. Dynamic schedulers such as StarPU, PaRSEC or StarSs are also developed for more constrained computations, e.g. task graphs coming from linear algebra. To make their decisions, these runtime systems make use of some static information, such as the distance of tasks to the critical path or the affinity between tasks and computing resources (CPU, GPU,...) and of dynamic information, such as where input data are actually located. In this paper, we concentrate on two elementary linear algebra kernels, namely the outer product and the matrix multiplication. For each problem, we propose several dynamic strategies that can be used at runtime and we provide an analytic study of their theoretical performance. We prove that the theoretical analysis provides very good estimate of the amount of communications induced by a dynamic strategy, thus enabling to choose among them for a given problem and architecture.

3.4 Energy-aware scheduling

Anne Benoit (ENS – Lyon, FR)

Joint work of Aupy, Guillaume; Benoit, Anne; Renaud-Goud, Paul; Robert, Yves

In this talk, I survey recent works on energy-efficient scheduling. The goal is to minimize the energy consumption of a schedule, given some performance constraints, for instance a bound on the total execution time. I first revisit the greedy algorithm for independent tasks in this context. Then I present problems accounting for the reliability of a schedule: if a failure may occur, then replication or checkpoint is used to achieve a reliable schedule. The goal remains the same, i.e., minimize the energy consumption under performance constraints.

3.5 Using Runtimes to ease the way to Exascale

George Bosilca (University of Tennessee, US)

Joint work of Bosilca, G. and Bouteiller, A. and Danalis, A. and Hérault, T. and Luszczek, P. and Dongarra, J.
URL http://icl.utk.edu/~bosilca/

Novel system designs with steeply escalating resource counts, burgeoning heterogeneity, and increasing memory accesses unpredictability, calls for novel programming paradigms.

PaRSEC, our answer to this challenge, is a layered approach that provides a clear separation of concerns between architecture, algorithm, and data distribution by allowing the algorithm to be decoupled from the data distribution and the underlying hardware. Developers can focus solely on the algorithmic level without the constraints involved with programming for hardware trends.

References

1 Bosilca, G. and Bouteiller, A. and Danalis, A. and Hérault, T. and Luszczek, P. and Dongarra, J., *D*ense Linear Algebra on Distributed Heterogeneous Hardware with a Symbolic DAG Approach, in Scalable Computing and Communications: Theory and Practice, John Wiley & Sons, ISBN 978-1-118-16265-1, March 2013

3.6 Enabling the production deployment of advanced Fault Tolerance techniques: Fault Tolerant MPI

Aurélien Bouteiller (University of Tennessee, US)

As Supercomputers are entering an era of massive parallelism where the frequency of faults is increasing, the MPI Standard remains distressingly vague on the consequence of failures on MPI communications. Advanced fault tolerance techniques have the potential to prevent full scale application restart and therefore lower the cost incurred for each failure, but they demand from MPI the capability to detect failures and resume communications afterward. In this talk, we present a set of extensions to MPI that allow restoring communication capabilities, while maintaining the extreme level of performance to which MPI users have become accustomed. The general design is to avoid impacting existing interface behavior, and constrain recovery actions to a limited number of new interfaces. The performance impact (or lack thereof) on MPI communication performance is outlined before we use the Wang-Landau protein folding application as an example to illustrate how to employ the new interfaces to effectively express advanced fault tolerance techniques.

3.7 Scheduling Data Sensor Retrieval for Boolean Tree Query Processing

Henri Casanova (University of Hawaii at Manoa, US)

The processing of queries expressed as trees of boolean operators applied to predicates on sensor data streams has several applications in mobile computing. Sensor data must be retrieved from the sensors, which incurs a cost, e.g., an energy expense that depletes the battery of the mobile query processing device. The objective is to determine the order in which predicates should be evaluated so as to shortcut part of the query evaluation and minimize the expected cost. This problem has been studied assuming that each data stream occurs at a single predicate. In this work we remove this assumption since it does not necessarily hold for real-world queries. Our main results are a novel optimal algorithm for single-level trees and a proof of NP-completeness for DNF trees. For DNF trees, however, we show that there is an optimal predicate evaluation order that corresponds to a depth-first traversal. This result provides inspiration for a class of heuristics. We show that one of these heuristics largely outperforms other sensible heuristics, including a heuristic proposed in previous work.

3.8 Does the cache matter? When? Why? and How Much?

Anthony Danalis (University of Tennessee, US)

It is a well known fact that the cache hierarchy can have a profound effect on the performance of codes with heavy memory usage. However, it seems that several misconceptions regarding the effects of the cache on application performance are deeply rooted in the community. In the first part of this talk I will examine the effects of the cache hierarchy on two extreme cases of applications, in terms of memory access patterns. On one extreme we have memory bound codes with no, or little data reuse, such as simple buffer traversals. On the other extreme we have codes with heavy computation and good reuse patterns, such as matrix- matrix multiply. Using experimental results, I will first demonstrate that the cache is neither very easy to utilize, nor very easy to trick into pathological behavior. In the reminder of the talk I will use parallel codes to examine the interaction between the cache hierarchy and the remainder of the memory subsystem – i.e., NUMA memory bus. The algorithm used in this part of the talk will be matrix-matrix multiply and specifically the highly sophisticated implementation that exists in the Intel MKL library. Using experimental results collected from custom parallel benchmarks built on top of this matrix multiply kernel I will demonstrate that the dominant factor affecting the performance of such parallel codes is not the cache friendliness of the scheduling, or the size of the data with respect to the cache hierarchy, but rather the load that the scheduling is putting on the memory bus and the cross-traffic generated.

3.9 Comparison of push and pull strategies in server farms

Bruno Gaujal (INRIA Rhône-Alpes, FR)

Joint work of Gast, Nicolas; Gaujal, Bruno;
Main reference N. Gast, B. Gaujal, "Markov chains with discontinuous drifts have differential inclusion limits," Performance Evaluation, 69(12):623–642, 2012.
URL http://dx.doi.org/10.1016/j.peva.2012.07.003

Let us consider a model of a server farm composed of N identical servers. Jobs arrive according to a Poisson process with rate Nr and have a size exponentially distributed with mean 1. Each server can buffer up to B jobs.

We consider two strategies that improve load balancing:

- pull strategy – we add a centralized server that serves jobs at rate Np. It chooses to serve a job from the cluster with longest queue first. For fair comparison, we consider that the total computing capacity remains N (the new speed of the servers is set to $1 - p$). This model is similar to the model studied in by Tsitsiklis & Xu in 2011.
- push strategy – with probability q, a job is pushed to the cluster with the shortest queue. With probability $1 - q$, it is routed to a server at random.

We use an asymptotic model by letting N go to infinity (mean field approximation) to measure the performance of the system under the three strategies (push, pull or none).

When the load of the system approaches one (r goes to one), the expected response time of jobs for the fully distributed system (with no pull nor push) verifies: $E[W] = O(1/(1-r))$.

With an arbitrary small degree of pull ($p > 0$), then the response times are drastically smaller: $E[W] = O(log1/(1-r))$.

With a small degree of push, ($q > 0$), the expected response times remain bounded: $E[W] < 1/q$.

3.10 Exascale techniques for Numerics for PDEs (Part II)

Dominik Goeddeke (TU Dortmund, DE)

We motivate the concept of hardware-oriented numerics to address the challenges imposed on the simulation of real-life phenomena by increasingly heterogeneous and parallel hardware: Only by integrating hardware aspects directly in the design of novel numerical solution methods for PDEs, the conflicting goals of scalability and total efficiency can be reached. Our examples include CFD problems, low-power architectures, exposing parallelism and resilience.

3.11 Sparse QR factorization for multicore architectures over runtime systems

Abdou Guermouche (University of Bordeaux, FR)

Joint work of E. Agullo, A. Buttari, A. Guermouche and F. Lopez
Main reference E. Agullo, A. Buttari, A. Guermouche, F. Lopez, "Multifrontal QR Factorization for Multicore Architectures over Runtime Systems," in Proc. of the 19th Int'l Conf. on Parallel Processing (Euro-Par'13), LNCS, Vol. 8097, pp. 521–532, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-40047-6_53

To face the advent of multicore processors and the ever increasing complexity of hardware architectures, programming models based on DAG parallelism regained popularity in the high performance, scientific computing community. Modern runtime systems offer a programming interface that complies with this paradigm and powerful engines for scheduling the tasks into which the application is decomposed. These tools have already proved their effectiveness on a number of dense linear algebra applications. This paper evaluates the usability of runtime systems for complex applications, namely, sparse matrix multifrontal factorizations which constitute extremely irregular workloads, with tasks of different granularities and characteristics and with a variable memory consumption. Experimental results on real-life matrices show that it is possible to achieve the same efficiency as with an ad hoc scheduler which relies on the knowledge of the algorithm. A detailed analysis shows the performance behavior of the resulting code and possible ways of improving the effectiveness of runtime systems.

3.12 Unified Model for Assessing Checkpointing Protocols at Extreme-Scale

Amina Guermouche (University of Versailles, FR)

This talk presents a unified model for several well-known checkpoint/restart protocols. The proposed model is generic enough to encompass both extremes of the checkpoint/restart space, from coordinated approaches to a variety of uncoordinated checkpoint strategies (with message logging). We identify a set of crucial parameters, instantiate them and compare the expected efficiency of the fault tolerant protocols, for a given application/platform pair. We then propose a detailed analysis of several scenarios, including some of the most powerful currently available HPC platforms, as well as anticipated Exascale designs. The results of this analytical comparison are corroborated by a comprehensive set of simulations. Altogether, they outline comparative behaviors of checkpoint strategies at very large scale, thereby providing insight that is hardly accessible to direct experimentation.

3.13 ABFT & Checkpoint-Composite Strategy

Thomas Hérault (University of Tennessee, US)

Joint work of Hérault, Thomas; Bosilca, George; Bouteiller, Aurélien; Robert, Yves; Dongarra, Jack
Main reference G. Bosilca, A. Bouteiller, T. Hérault, Y. Robert, J. Dongarra, "Assessing the impact of ABFT and checkpoint composite strategies," Research Report ICL-UT-13-03, University of Tennessee, 2013.
URL http://icl.cs.utk.edu/news_pub/submissions/brainstorm.pdf

Algorithm-specific fault tolerant approaches promise unparalleled scalability and performance in failure-prone environments. With the advances in the theoretical and practical understanding of algorithmic traits enabling such approaches, a growing number of frequently used algorithms (including all widely used factorization kernels) have been proven capable of such properties. These algorithms provide a temporal section of the execution when the data is protected by its own intrinsic properties, and can be algorithmically recomputed without the need of checkpoints. However, while typical scientific applications spend a significant fraction of their execution time in library calls that can be ABFT-protected, they interleave sections that are difficult or even impossible to protect with ABFT. As a consequence, the only fault-tolerance approach that is currently used for these applications is checkpoint/restart. In this presentation, I presented an algorithm, a model and a simulator to investigate the behavior of a composite protocol, that alternates between ABFT and checkpoint/restart protection for effective protection of each phase of an iterative application composed of ABFT-aware and ABFT- unaware sections. In this work, we highlight this approach drastically increases the performance delivered by the system, especially at scale, by providing means to rarefy the checkpoints while simultaneously decreasing the volume of data needed to be saved in the checkpoints.

3.14 Tree traversals with task-memory affinities

Julien Herrmann (ENS – Lyon, FR)

Joint work of Herrmann, Julien; Robert, Yves; Marchal, Loris
Main reference J. Herrmann, Y. Robert, L. Marchal, "Model and Complexity Results for Tree Traversals on Hybrid Platforms," in Proc. of the 19th Int'l Conf. on Parallel Processing (Euro-Par'13), LNCS, Vol. 8097, pp. 647–658, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-40047-6_65

We study the complexity of traversing tree-shaped workflows whose tasks require large I/O files. We target a heterogeneous architecture with two resource types, each with a different memory, such as a multicore node equipped with a dedicated accelerator (FPGA or GPU). The tasks in the workflow are colored according to their type and can be processed if all there input and output files can be stored in the corresponding memory. The amount of used memory of each type at a given execution step strongly depends upon the ordering in which the tasks are executed, and upon when communications between both memories are scheduled. The objective is to determine an efficient traversal that minimizes the maximum amount of memory of each type needed to traverse the whole tree. In this paper, we establish the complexity of this two-memory scheduling problem, and provide inapproximability results. In addition, we design several heuristics, based on both post- order and general traversals, and we evaluate them on a comprehensive set of tree graphs, including random trees as well as assembly trees arising in the context of sparse matrix factorizations.

3.15 On the Scalability of Moldable Task Scheduling Algorithms

Sascha Hunold (TU Wien, AT)

Parallel task scheduling is now more relevant than ever, especially under the moldable task model, in which tasks are allocated a fixed number of processors before execution. A common assumption for scheduling algorithms is that the speedup function of moldable tasks is either non-decreasing, sub-linear or concave. However, in practice the resulting speedup of parallel programs on current hardware with deep memory hierarchies is most often neither non-decreasing nor concave. We present an algorithm for scheduling moldable tasks with precedence constraints for the makespan objective and arbitrary speedup functions. We show through simulation that the algorithm not only creates competitive schedules for moldable tasks with arbitrary speedup functions, but also outperforms other published heuristics and approximation algorithms for non-decreasing speedup functions.

3.16 Communication and Topology-aware Load Balancing in Charm++ with TreeMatch

Emmanuel Jeannot (INRIA – Bordeaux, FR)

Joint work of Jeannot, Emmanuel; Meneses-Rojas, Esteban; Mercier, Guillaume; Tessier, François; Zheng, Gengbin
Main reference E. Jeannot, E. Meneses, G. Mercier, F. Tessier, G. Zheng, "Communication and Topology-aware Load Balancing in Charm++ with TreeMatch," in Proc. of IEEE Cluster 2013, IEEE, 2013.

Programming multicore or manycore architectures is a hard challenge particularly if one wants to fully take advantage of their computing power. Moreover, a hierarchical topology implies that communication performance is heterogeneous and this characteristic should also be exploited. We developed two load balancers for Charm++ that take into account both aspects depending on the fact that the application is compute-bound or communication-bound. This work is based on our TREEMATCH library that compute process placement in order to reduce an application communication cost based on the hardware topology. We show that the proposed load-balancing scheme manages to improve the execution times for the two classes of parallel applications.

3.17 Many-task computing for the rest of us

Thilo Kielmann (Vrije Universiteit Amsterdam, NL)

Joint work of Oprescu, Ana-Maria Oprescu; Kielmann, Thilo; Leahu, Haralambie
Main reference A.-M. Oprescu, T. Kielmann, H. Leahu, "Budget Estimation and Control for Bag-of-Tasks Scheduling in Clouds," Parallel Processing Letters, Vol. 21, No. 2, pp. 219–243, June 2011.
URL http://dx.doi.org/10.1142/S0129626411000175

Exascale machines come with unprecedented challenges for applications that wish to exploit their potential. Efficient communication for tightly-coupled codes, energy efficiency, and fault tolerance are the ones that come to ones mind immediately. Whereas exascale machines still have to be built, there are already large-scale machines in daily commercial exploitation: the data centers of large cloud providers such as Google and Amazon. Can we learn something from these data centers for the exploitation of exascale supercomputers?

In this talk I outline the many-task computing (MTC) model that strives to bridge the gap between static, tightly-coupled applications (as with MPI) and the embarrassingly parallel model of high-throughput computing (as with Condor and BOINC). With MTC, individual tasks (both sequential and parallel) are orchestrated to form an application while typically communicating through shared file systems. I outline the task farming service of our ConPaaS platform (www.conpaas.eu) that, as an example of MTC, implements bag-of-tasks execution on cloud infrastructures, minimizing the overall makespan while respecting a user-selected monetary budget. It does so by sampling the bag of tasks on different cloud machine types, and by predicting the overall makespan and cost for different combinations of instances from available machine types.

The possibly provocative hypothesis of this talk is the applicability of such an execution model for exascale machines, as it provides a loosely-coupled, scalable, and fault-tolerant execution model, while introducing a financial incentive for application users to only use as many resources as their applications really need. Constructing a price model based on energy consumption (and capabilities) of nodes in a supercomputer should give incentive to resource-conscious use of such supercomputers.

3.18 On Variants of the Hierarchically Structured Bin Packing Problem

Thomas Lambert (ENS – Lyon, FR)

Joint work of Lambert, Thomas ; Marchal, Loris; Ucar, Bora

We deal with a variant of the Bin Packing problem: the Hierarchically Structured Bin Packing. The inputs of this problem are a tree and an integer B and the output is a packing of the leaves of the tree into bins such that every bin as at most B leaves. The objective function is to minimize the total dispersal number. This number is equal to the sum, over all internal nodes, of the number of bins into which the descendants of an internal node are packed. We investigate four variants of this problem. We discuss inapproximability results, an exact algorithm with polynomial time complexity for constant arity trees, and simulation results. At the end, we identify a new variant of our problem with two trees and present two inapproximability results.

3.19 Hierarchical Approach to Improve Performance of Legacy Scientific Applications on Large-Scale Platforms

Alexey Lastovetsky (University College Dublin, IE)

Joint work of Quintin, Jean-Noel; Hasanov, Khalid;Lastovetsky, Alexey;
Main reference J.-N. Quintin, K. Hasanov, A. Lastovetsky, "Hierarchical Parallel Matrix Multiplication on Large-Scale Distributed Memory Platforms," arXiv:1306.4161v1 [cs.DC], 2013.
URL http://arxiv.org/abs/1306.4161v1

Many state-of-the-art parallel algorithms, which are widely used in scientific applications executed on high-end computing systems, were designed in 20th century with relatively small-scale parallelism in mind. Indeed, while in 1990s a system with few hundred cores was considered a powerful supercomputer, modern top supercomputers have millions of cores. In this talk, we present a hierarchical approach to optimization of message- passing parallel algorithms for execution on large-scale distributed-memory systems. The idea is to reduce the communication cost by introducing hierarchy and hence more parallelism in the communication scheme. We apply this approach to SUMMA, the state-of-the-art parallel algorithm for matrix-matrix multiplication, and demonstrate both theoretically and experimentally that the modified Hierarchical SUMMA significantly improves the communication cost and the overall performance on large-scale platforms.

3.20 A greedy algorithm for optimally pipelining a reduction

Bradley Lowery (University of Colorado, US)

Joint work of Lowery, Bradley; Langou, Julien;
Main reference B. Lowery, J. Langou, "A greedy algorithm for optimally pipelining a reduction," arXiv:1310.4645v1 [cs.DC], 2013.
URL http://arxiv.org/abs/1310.4645v1

Collective communications are ubiquitous in parallel applications. We present two new algorithms for performing a reduction. The operation associated with our reduction needs to be associative and commutative. The two algorithms are developed under two different communication models (unidirectional and bidirectional). Both algorithms use a greedy scheduling scheme. For a unidirectional, fully connected network, we prove that our greedy algorithm is optimal when some realistic assumptions are respected. Previous algorithms fit the same assumptions and are only appropriate for some given configurations. Our algorithm is optimal for all configurations. We note that there are some configuration where our greedy algorithm significantly outperform any existing algorithms. This result represents a contribution to the state-of-the art. For a bidirectional, fully connected network, we present a different greedy algorithm. We verify by experimental simulations that our algorithm matches the time complexity of an optimal broadcast (with addition of the computation). Beside reversing an optimal broadcast algorithm, the greedy algorithm is the first known reduction algorithm to experimentally attain this time complexity. Simulations show that this greedy algorithm performs well in practice, outperforming any state-of-the-art reduction algorithms. Positive experiments on a parallel distributed machine are also presented.

References

1 Bradley Lowery and Julien Langou, "A greedy algorithm for optimally pipelining a reduction," arXiv:1310.4645, 2013.

3.21 Efficient Bi-objective Optimization of Energy and Makespan for Exascale Heterogeneous Computing Systems

Anthony A. Maciejewski (Colorado State University, US)

Joint work of Tarplee, Kyle M.; Friese, Ryan; Maciejewski, Anthony A.; Siegel, Howard Jay
Main reference K. Tarplee, R. Friese, A. A. Maciejewski, H. J. Siegel, "Efficient and Scalable Computation of the Energy and Makespan Pareto Front for Heterogeneous Computing Systems," in Proc. of the 6th Workshop on Computational Optimization (WCO'13)/Proceedings of the 2013 Federated Conf. on Computer Science and Information Systems (FedCSIS'13), pp. 401–408, Krakow, Poland, Sept. 8–11, 2013.
URL http://www.engr.colostate.edu/~aam/pdf/conferences/152.pdf

Future exascale systems will have to simultaneously reduce energy consumption while they strive to maximize performance. Because these are conflicting objectives, system administrators will need to evaluate the tradeoffs of different potential resource allocations. In this work we present a novel algorithm that provides an efficient method for computing the Pareto front of optimal solutions to the bi-objective problem of minimizing energy and makespan for a bag of tasks allocated to a set of heterogeneous compute resources.

3.22 Scheduling tree-shaped task graphs to minimize memory and makespan

Loris Marchal (ENS – Lyon, FR)

Joint work of Eyraud-Dubois, Lionel; Marchal, Loris; Sinnen, Oliver; Vivien, Frédéric
Main reference L. Marchal, O. Sinnen, F. Vivien, "Scheduling Tree-Shaped Task Graphs to Minimize Memory and Makespan," IPDPS 2013: 839–850; available as pre-print: arXiv:1210.2580v1 [cs.DC], 2012.
URL http://arxiv.org/abs/1210.2580v1

This work investigates the execution of tree-shaped task graphs using multiple processors. Each edge of such a tree represents a large data. A task can only be executed if all input and output data fit into memory. Such trees arise in the multifrontal method of sparse matrix factorization. The maximum amount of memory needed depends on the execution order of the tasks. With one processor, the problem of finding the tree traversal with minimum required memory was well studied and optimal polynomial algorithms has been proposed. Here, we extend the problem by considering multiple processors. With the multiple processors comes the additional objective to minimize the makespan. Not surprisingly, this problem proves to be much harder. We study its computational complexity and provide an inapproximability result even for unit weight trees. Several heuristics are proposed, especially for the realistic problem of minimizing the makespan under a strong memory constraint. They are analyzed in an extensive experimental evaluation using realistic trees.

3.23 Liveness Evaluation for Cyclo-Static DataFlow Graphs

Alix Munier (UPMC – Paris, FR)

Joint work of Nenazouz, Mohamed; Bodin, Bryon; Hujsa, Thomas; Munier-Kordon, Alix
Main reference M. Benazouz, B. Bodin, T. Hujsa, A. Munier-Kordon, "Liveness evaluation of a cyclo-static DataFlow graph," in Proc. of the 50th Design Automation Conf. (DAC'13), ACM, 2013.
URL http://dx.doi.org/10.1145/2463209.2488736

Static DataFlow Graphs (CSDFG in short) is a formalism commonly used to model parallel applications composed by actors communicating through buffers. The liveness of a CSDFG ensures that all actors can be executed infinitely often. This property is clearly fundamental for the design of embedded applications. This talk aims to present first an original sufficient condition of liveness for a CSDFG. Two algorithms of polynomial-time for checking the liveness are then derived and compared to a symbolic execution of the graph. The performance of our methods are comparable to those existing in the literature for industrial applications. However, they are far more effective on randomly generated instances, ensuring their scalability for future more complex applications and their possible implementation in a compiler.

3.24 Planning for ExaScale Systems: The Challenge to be Prepared

Wolfgang E. Nagel (TU Dresden, DE)

Joint work of Nagel, Wolfgang E.; Hackenberg, Daniel; Juckeland, Guido; Brunst, Holger; Bungartz, Hans-Joachim

ExaScale systems are expected to arrive between 2018 and 2020, and they will have lots of cores, may be arranged in a heterogeneous way. The challenge will be to have programs that scale to hundreds of millions of parallel threads, and runtime environments that will efficiently support such a high level of parallelism with load imbalances, many parallel I/O requests, possibly heterogeneous cores and more often than today failing hardware components. There will be strong need to have software ready which provide solutions in such complex environments, and, at least in Germany, despite the activities from BMBF (HPC calls) and DFG (SPPEXA) we still invest much more in "racks" and not enough in "brains".

The talk presents actual developments on the scalability of performance tools and describes research issues, which have to be solved to get solutions even on PFlops systems today. It also shows how algorithm development and tuning can benefit from these achievements. In a case study, results are presented for the code PIConGPU which has been optimized on Titan at ORNL. With more than 7 PFlops (64 Bit) and about 1,5 PFlops (32 Bit) measured on that machine, PIConGPU is one of the six finalists for the Gordon Bell Price Award in 2013.

3.25 Task Scheduling to Extend Platform Useful Life using Prognostics

Jean-Marc Nicod (FEMTO-ST Institute – Besancon, FR)

Joint work of Nicod, Jean-Marc; Obretch, Léo; Varnier, Christophe
Main reference J.-M. Nicod, L. Obrecht, C. Varnier, "Task scheduling to extend platform useful life using prognostics," in Proc. of the International IFAC Conference on Manufacturing Modelling, Management and Control (MIM'2013), Russian Federation, 2013.
URL http://hal.inria.fr/hal-00838457

In this talk, we study the problem of optimizing the useful life of a heterogeneous distributed platform which has to produce a given production throughput rate. The machines perform independent tasks and may be configured with different profiles involving variable throughput rates. At each time the sum of the machine throughputs that are currently running determine the global throughput of the platform. Moreover, each machine of the platform is supposed to be monitored and a prognostic module gives a remaining useful life (RUL) depending on both its past and future usage (throughput rate). The problem that is tackled in this talk is to configure the platform so as to reach the demand as long as possible.

Our approach consists in discretizing the time in period and a configuration is chosen at the beginning of each period. We prove that the problem can be optimally solved in polynomial time in the homogeneous case and that this problem becomes NP-Complete since the the throughput is heterogeneous on the machine. We propose a Integer Linear Programming (ILP) model to find such a configuration for a fixed time horizon. Due to the ILP, the largest horizon can be computed for small instances of the problem. For larger instances, we propose polynomial time heuristics to reach a horizon. Exhaustive simulations show that the heuristic solutions are close to the optimal in particular case where the optimal horizon can be computed (5% in average). For other platforms with a very large number of machines, simulations assess the efficiency of our heuristics. The distance to the theoretical maximal value is never up to 20%.

3.26 Energy-aware Execution of Fork-Join-based Task Parallelism

Thomas Rauber (Universität Bayreuth, DE)

Joint work of Rauber, Thomas; Rünger, Gudula
Main reference T. Rauber, G. Rünger, "Energy-aware Execution of Fork-Join-based Task Parallelism," in Proc. of the 20th IEEE Int'l Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'12), pp. 231–240, IEEE, 2012.
URL http://dx.doi.org/10.1109/MASCOTS.2012.35

In this talk, we use an analytical energy model based on frequency scaling to model the energy consumption of tasks in a fork-join pattern of parallelism. In particular, tasks that may be executed concurrently to each other are considered, and the resulting energy consumption for different processor assignments is investigated. Frequency scaling factors that lead to a minimum energy consumption are derived and used in task-based scheduling algorithms. An experimental evaluation provides simulations for a large number of randomly generated task sets as well as energy measurements on a Intel Sandy Bridge architecture using a complex application from numerical analysis.

3.27 Reliable Service Allocation in Clouds with Memory and Capacity Constraints

Paul Renaud-Goud (INRIA – Bordeaux, FR)

Joint work of Beaumont, Olivier; Eyraud-Dubois, Lionel; Pesnaud, Pierre; Renaud-Goud, Paul
Main reference O. Beaumont, L. Eyraud-Dubois, P. Pesneau, P. Renaud-Goud, "Reliable Service Allocation in Clouds with Memory and Capacity Constraints," in Proc. of the 6th Workshop on Resiliency in High Performance Computing (Resilience) in Clusters, Clouds, and Grids in conjunction with the 19th Int'l European Conf. on Parallel and Distributed Computing (Euro-Par 2013), 2013.

We consider allocation problems that arise in the context of service allocation in Clouds. More specifically, on the one part we assume that each Physical Machine (denoted as PM) is offering resources (memory, CPU, disk, network). On the other part, we assume that each application in the IaaS Cloud comes as a set of services running as Virtual Machines (VMs) on top of the set of PMs. In turn, each service requires a given quantity of each resource on each machine where it runs (memory footprint, CPU, disk, network). Moreover, there exists a Service Level Agreement (SLA) between the Cloud provider and the client that can be expressed as follows: the client requires a minimal number of service instances which must be alive at the end of the day, with a given reliability (that can be converted into penalties paid by the provider). In this context, the goal for the Cloud provider is to find an allocation of VMs onto PMs so as to satisfy, at minimal cost, both capacity and reliability constraints for each service. In this paper, we propose a simple model for reliability constraints and we prove that it is possible to derive efficient heuristics.

3.28 Some thoughts on exascale and scheduling

Rizos Sakellariou (University of Manchester, GB)

The talk argues that past ways of thinking about High-Performance-Computing affect our way of viewing exascale computing. However, as progress is not necessarily linear we may need to focus more deeply on the key issues on exascale computing, such as fault-tolerance, communication and energy. The talk makes reference to recent results related to energy-aware resource provisioning [1] and argues that real energy measurements and a more holistic view may be necessary to understand energy consumption.

References

1 Ilia Pietri, Maciej Malawski, Gideon Juve, Ewa Deelman, Jarek Nabrzyski, Rizos Sakellariou. Energy Constrained Provisioning for Scientific Workflow Ensembles. In CGC'13 (Proceedings of the 2013 IEEE Third International Conference on Cloud and Green Computing, Karlsruhe, Germany, September 30-October 2), pp. 34-41.

3.29 Parallel Dataflow Graph Coloring

Erik Saule (University of North Carolina at Charlotte, US)

In this talk, we investigate the shared memory graph coloring problem. Existing methods such as speculative first fit coloring requires to make multiple passes on the graph so as to achieve pleasant parallelism. To reduce the execution to a single pass on the graph, a more synchronized algorithm is required. We present the dataflow coloring algorithm where a vertex v can only be colored once its neighboors $u < v$ have been colored. We show that the execution of this algorithm behaves like the output of a list scheduling algorithm on a DAG obtained by an orientation of the edges of the colored graph. However additional edges are added to the DAG in order to take the effect of the parallel runtime system into account. To solve the problem of bad orientation of the edges which can lead to a high critical path, we show experimentally that picking a randomized reordering of the vertices improves the amount of achieved concurrency.

3.30 Resource Management in an IaaS Environment

Uwe Schwiegelshohn (TU Dortmund, DE)

This talk starts by describing constraints for resource management in an Infrastructure-as-a-Service (IaaS) market economy. Then user requests are divided into the categories interactive, user-facing, and batch. Also the slack factor is suggested as a quantitative metric for a service guarantee in this scenario. It is explained that an IaaS provider is particularly interested in optimizing the utilization of his computing resources while satisfying the expectations of his users. As similar problems have already been addressed in the real-time scheduling area, relevant previous algorithmic results are presented. However, the application of these results in practice does not produce satisfactory results in general. Therefore, a new approach to generate provider offers is defined. This approach considers the previously discussed IaaS constraints and allows an efficient resource management. Theoretical analysis is used to determine the performance of the resulting scheduling methods and their variations. The methods include preemptive deadline scheduling on machines with different speed.

3.31 Stochastic-Based Deadline-Aware and Energy-Constrained Robust Static Resource Management

Howard Jay Siegel (Colorado State University, US)

Joint work of Mark Oxley, Sudeep Pasricha, Howard Jay Siegel, and Anthony A. Maciejewski
Main reference Mark Oxley, Sudeep Pasricha, Howard Jay Siegel, and Anthony A. Maciejewski, "Energy and Deadline Constrained Robust Stochastic Static Resource Allocation," 1st Workshop on Power and Energy Aspects of Computation (PEAC'13), in the Proc. of the 10th Int'l Conference on Parallel Processing and Applied Mathematics (PPAM'13), 2013; to appear.
URL http://www.engr.colostate.edu/~hj/conferences/338.pdf

We study the problem of robust static (off-line) resource management for multicore-based heterogeneous parallel computing systems. We consider allocating resources to a known set of independent tasks (the "bag-of- tasks" workload model). We use a stochastic model (random variable) for the execution time of each task on each type of core in the system. Our goal is to design static resource management techniques that maximize the probability of finishing each task by a common deadline (our robustness metric) while ensuring the resource allocation meets a constraint on the probability of obeying a given energy budget. We describe our design and evaluation of a new resource allocation heuristic based on Tabu Search, where we exploit heterogeneity and employ dynamic voltage and frequency scaling (DVFS) by incorporating problem-domain specific local search techniques.

3.32 Tools and Resources for Task Scheduling Research

Oliver Sinnen (University of Auckland, NZ)

Scheduling of task graphs (DAGs) on parallel systems with communication delay ($P|prec, c_{ij}|C_max$) is a classical scheduling problem. It is NP-hard in the strong sense and part of many computing problems. Given its importance, there is still a strong interest of the research community in this problem. In this talk tools and resources for research of this problem are presented. This includes a visual tool for the analysis and manual manipulation of schedules, optimal solvers for small to medium sized problem instances, and a database of optimal schedules for a large set of task graphs on different numbers of processors. The tools and resources are available at http://www.ece.auckland.ac.nz/~parallel/OptimalTaskScheduling/.

3.33 Job Scheduling Using Successive Linear Approximations of a Sparse Model

Veronika Sonigo (University of Franche-Comté – Besancon, FR)

Joint work of Rehn-Sonigo, Veronika; Chrétien, Stéphane; Nicod, Jean Marc; Toch, Lamiel; Philippe, Laurent;
Main reference S. Chrétien, J.-M. Nicod, L. Philippe, V. Rehn-Sonigo, L. Toch, "Job Scheduling Using Successive Linear Programming Approximations of a Sparse Model," in Proc. of the 18th Int'l Conf. on Parallel Processing (Euro-Par'12), LNCS, Vol. 7484, pp. 116–127, Springer, 2012.
URL http://dx.doi.org/10.1007/978-3-642-32820-6_14
URL http://hal.inria.fr/hal-00789217

In this talk we tackle the well-known problem of scheduling a collection of parallel jobs on a set of processors either in a cluster or in a multiprocessor computer. For the makespan objective, i.e., the completion time of the last job, this problem has been shown to be NP-Hard and several heuristics have already been proposed to minimize the execution time. We introduce a novel approach based on successive linear programming (LP) approximations of a sparse model. The idea is to relax an integer linear program and use lp norm-based operators to force the solver to find almost-integer solutions that can be assimilated to an integer solution. We consider the case where jobs are either rigid or moldable. A rigid parallel job is performed with a predefined number of processors while a moldable job can define the number of processors that it is using just before it starts its execution. We compare the scheduling approach with the classic Largest Task First list based algorithm and we show that our approach provides good results for small instances of the problem. The contributions of this paper are both the integration of mathematical methods in the scheduling world and the design of a promising approach which gives good results for scheduling problems with less than a hundred processors.

3.34 Exascale techniques for Numerics for PDEs (Part I)

Stefan Turek (TU Dortmund, DE)

We motivate the concept of hardware-oriented numerics to address the challenges imposed on the simulation of real-life phenomena by increasingly heterogeneous and parallel hardware: Only by integrating hardware aspects directly in the design of novel numerical solution methods for PDEs, the conflicting goals of scalability and total efficiency can be reached. Our examples include CFD problems, low-power architectures, exposing parallelism and resilience.

3.35 Semi-matching algorithms for scheduling parallel tasks under resource constraints

Bora Ucar (ENS – Lyon, FR)

Joint work of
Main reference A. Benoit, J. Langguth, B. Ucar, "Semi-matching algorithms for scheduling parallel tasks under resource constraints," accepted to be published in IPDPSW 2013.
URL http://hal.inria.fr/hal-00738393

We study the problem of minimum makespan scheduling when tasks are restricted to subsets of the processors (resource constraints), and require either one or multiple distinct processors to be executed (parallel tasks). This problem is related to the minimum makespan scheduling problem on unrelated machines, as well as to the concurrent job shop problem, and it amounts to finding a semi- matching in bipartite graphs or hypergraphs. The problem is known to be NP- complete for bipartite graphs with general vertex (task) weights, and solvable in polynomial time for unweighted graphs (i.e., unit-weight tasks). We prove that the problem is NP-complete for hypergraphs even in the unweighted case. We design several greedy algorithms of low complexity to solve two versions of the problem, and assess their performance through a set of exhaustive simulations. Even though there is no approximation guarantee for these low-complexity algorithms, they return solutions close to the optimal (or a known lower bound) in average.

3.36 Checkpointing and fault prediction

Frédéric Vivien (ENS – Lyon, FR)

Joint work of Aupy, Guillaume; Robert, Yves; Vivien, Frédéric; Zaidouni, Dounia
Main reference G. Aupy, Y. Robert, F. Vivien, D. Zaidouni, "Checkpointing strategies with prediction windows," in Proc. of the IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'13), 2013.
URL http://hal.inria.fr/hal-00847622

This work deals with the impact of fault prediction techniques on checkpointing strategies. We consider fault-prediction systems that do not provide exact prediction dates, but instead time intervals during which faults are predicted to strike. These intervals dramatically complicate the analysis of the checkpointing strategies. We propose a new approach based upon two periodic modes, a regular mode outside prediction windows, and a proactive mode inside prediction windows, whenever the size of these windows is large enough. We are able to compute the best period for any size of the prediction windows, thereby deriving the scheduling strategy that minimizes platform waste. In addition, the results of the analytical study are nicely corroborated by a comprehensive set of simulations, which demonstrate the validity of the model and the accuracy of the approach.

4 Open Problems

A short "Open Problems" session was held on the Thursday morning. Three open problems were introduced by three participants. The first two problems were theoretical in nature, i.e., mathematical problems directly connected to simple yet fundamental scheduling problems. Interestingly, one of these two problems was actually solved by yet another workshop participant that evening. The other problems received some attention but no solution was achieved and it is not clear that such a solution can be computed. The third problem was much larger in scope and targeted to the resource management models in cloud computing infrastructures. In other words, given that current resource abstractions provided by cloud infrastructures can be vastly sub-optimal especially given the heterogeneity of the underlying physical infrastructure, which new abstractions should be used and how? This problem led to a lively discussion that, it is fair to say, left the group with more (interesting) questions than answers.

Participants

- Ismail Assayad
University Hassan II – Casablanca, MA
- Guillaume Aupy
ENS – Lyon, FR
- Olivier Beaumont
INRIA – Bordeaux, FR
- Anne Benoit
ENS – Lyon, FR
- George Bosilca
University of Tennessee, US
- Aurélien Bouteiller
University of Tennessee, US
- Heinrich Braun
SAP AG – Walldorf, DE
- Henri Casanova
Univ. of Hawaii at Manoa, US
- Anthony Danalis
University of Tennessee, US
- Carsten Franke
ABB – Baden-Dättwil, CH
- Bruno Gaujal
INRIA Rhône-Alpes, FR
- Dominik Göddeke
TU Dortmund, DE
- Christian Grimme
Universität Münster, DE
- Abdou Guermouche
University of Bordeaux, FR
- Amina Guermouche
University of Versailles, FR
- Thomas Hérault
University of Tennessee, US
- Julien Herrmann
ENS – Lyon, FR
- Sascha Hunold
TU Wien, AT
- Emmanuel Jeannot
INRIA – Bordeaux, FR
- Thilo Kielmann
Vrije Univ. Amsterdam, NL
- Thomas Lambert
ENS – Lyon, FR
- Alexey Lastovetsky
University College Dublin, IE
- Bradley Lowery
University of Colorado, US
- Anthony A. Maciejewski
Colorado State University, US
- Loris Marchal
ENS – Lyon, FR
- Alix Munier
UPMC – Paris, FR
- Wolfgang E. Nagel
TU Dresden, DE
- Jean-Marc Nicod
FEMTO-ST Institute – Besancon, FR
- Thomas Rauber
Universität Bayreuth, DE
- Paul Renaud-Goud
INRIA – Bordeaux, FR
- Yves Robert
ENS – Lyon, FR
- Gudula Rünger
TU Chemnitz, DE
- Rizos Sakellariou
University of Manchester, GB
- Erik Saule
University of North Carolina at Charlotte, US
- Uwe Schwiegelshohn
TU Dortmund, DE
- Howard Jay Siegel
Colorado State University, US
- Oliver Sinnen
University of Auckland, NZ
- Veronika Sonigo
University of Franche-Comté – Besancon, FR
- Stefan Turek
TU Dortmund, DE
- Bora Ucar
ENS – Lyon, FR
- Frédéric Vivien
ENS – Lyon, FR

Report from Dagstuhl Seminar 13382

Collaboration and learning through live coding

Edited by

Alan Blackwell[1], Alex McLean[2], James Noble[3], and Julian Rohrhuber[4]

1 **University of Cambridge, GB,** alan.blackwell@cl.cam.ac.uk
2 **University of Leeds, GB,** a.mclean@leeds.ac.uk
3 **Victoria University of Wellington, NZ,** kjx@ecs.vuw.ac.nz
4 **Robert Schumann Hochschule für Musik – Düsseldorf, DE,** julian.rohrhuber@musikundmedien.net

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13382 "Collaboration and learning through live coding". Live coding is improvised interactive programming, typically to create electronic music and other digital media, done live with an audience. Our seminar was motivated by the phenomenon and experience of live coding. Our conviction was that those represent an important and broad, but seldom articulated, set of opportunities for computer science and the arts and humanities. The seminar participants included a broad range of scholars, researchers, and practitioners spanning fields from music theory to software engineering. We held live coding performances, and facilitated discussions on three main perspectives, the humanities, computing education, and software engineering. The main outcome of our seminar was better understanding of the potential of live coding for informing cross-disciplinary scholarship and practice, connecting the arts, cultural studies, and computing.

Seminar 15.–20. September, 2013 – www.dagstuhl.de/13382
1998 ACM Subject Classification J.5 [Computer Applications]: Arts and humanities—Performing arts, K.3.2 [Computers and Education]: Computer and Information Science Education—Computer science education, D.2.1 [Software Engineering]: Requirements/Specifications—Methodologies
Keywords and phrases Live coding, Collaboration, Learning, Improvised interactive programming, Computer music, Algorithmic composition, TOPLAP
Digital Object Identifier 10.4230/DagRep.3.9.130
Edited in cooperation with Jochen Arne Otto (ZKM | Center for Art and Media Karlsruhe, DE)

1 Executive Summary

Robert Biddle
Alex McLean
Alan Blackwell
Julian Rohrhuber

The goal of this seminar was to understand and develop the emerging practice, characteristics and opportunities in live coding, with an emphasis on three perspectives: the humanities, computing education, and software engineering. The opening days of the seminar were broadly structured to provide thematic introductions followed by facilitated discussions on each of these three perspectives. These were interspersed with live coding performances and experiments, in order to ensure that theoretical concerns remained grounded within this discipline that fundamentally blurs the separation of concerns between theory and practice.

The second half of the seminar was problem-oriented, resulting in concrete progress on specific technical topics, together with development of a research roadmap, publications and policy strategy to realise the significant benefits that live coding promises in a number of fields. Finally, in the spirit of both practice as a form of theory and theory as a form of practice, the seminar included some exciting musical experiences – an Algorave club night in London, with performances by delegates who were traveling from other countries on their way to the seminar; an inter-continental collaborative performance hosted jointly with the IEEE VL/HCC conference in San Jose; a conceptual proposal for an interactive sound installation in the Schloss Dagstuhl garden; and live-coded jam sessions in venues ranging from the woods of the old castle, to evening cabaret entertainment in the beautiful Dagstuhl music room.

Our main findings in relation to the three contrasting research perspectives were as follows:

1. Live coding illuminates the ways in which programming can be an artistic practice, software-as-art, going beyond a mere supporting role, and illustrating that software is itself a cultural artefact, not simply an infrastructure commodity. We see many opportunities for nuanced, cross-disciplinary contributions to the digital humanities, for example in a revitalisation of the historical connection between computation and weaving, insights into the role of practice and experiment, and an enrichment of the notion of computation itself. Indeed, as computing becomes embedded in culture, the live, everyday authorship of computation becomes a socio-political question of freedom of speech and empowerment.
2. Live coding can play an important role in computing education, because it allows programming to be demonstrated and learned in a simple but authentic context. At the same time, it can support an *affective* teaching strategy where learners are not only motivated by the production of sound, visuals and other phenomena, but are also clear on the distinctly human activity which produces them. Thereby, however, it maintains a sense of discovery of something unanticipated and not prefigured. Of particular importance for learning is the potential for deeper engagement with the non trivial nature of computing, rather than an occupation with the operation of end-user application software.
3. Live coding offers new insights with regard to software engineering processes. The history of software engineering process can be seen as a move from heavyweight lock-step approaches to more agile approaches with fast cycles of development and feedback. At their heart, the new approaches rely on collaboration, as developers, designers, and customers work together to steer the process toward mutual success. Live coding demonstrates this kind of approach in a compelling way, with simple tools, a short time frame, but still allowing improvisational collaboration between performers and various audiences.

Perhaps more significant than any of these individual considerations is an ambitious holistic vision: that live coding can entirely change the way we think about programming. Indeed, the common experience articulated at the workshop is that live coding exemplifies both the power and the excitement of programming – in a small space, in a short time, available and accessible to anyone. Live coding exposes the *soul* of programming.

Our next steps are a series of collaborative workshops and programs to articulate and demonstrate this collection vision of a broad and expanding role for live coding.

This report is a collection of texts that were direct outcomes of the seminar rather than having been set up in advance. Taking the role of editor seriously, Jochen Arne Otto has

diligently collected, arranged and revised the heterogeneous materials, a synthesis without which such a concise paper would not have been possible. While taking responsibility for all remaining mistakes and omissions, we would like to thank Jochen very much for his commitment to this difficult task.

2 Table of Contents

3 Overview of Talks

3.1 Live Coding as a Model for Cultural Practice

Geoff Cox (Aarhus University, DK)

Could a precedent for live coding be seen in Wolfgang von Kempelen's 18th century chess-playing automaton, a description of which introduces Walter Benjamin's essay "On the Concept of History"? The machine's trickery demonstrates that the dynamic of history and that of the machine is fake; and the task of the historical materialist is to reveal the inner workings as ideological constructions.

This relates somewhat to my interest in live coding: how it seems prototypical of contemporary labour dynamics and what is hidden from view – inasmuch as labour has become more performative and linguistic in character (and precarious, with reference to Amazon's Mechanical Turk – a crowdsourced online workplace – that exhibits so-called "artificial artificial intelligence"). Live coding makes apparent coding practice and embodiment (labour) and the visibility of code, coding and coders ("show us your screens", they proclaim).

Expressed ever so briefly, my point is that in a culture where behaviour is increasingly prescribed by various scripts, scores, programs (and these are enclosed), live coding offers a way to understand these dynamics, and accordingly suggests ways out of various automatisms and determinisms associated with code.

3.2 Live Coding in Education

Mark Guzdial (Georgia Institute of Technology – Atlanta, US)

In one of the first talks at the Dagstuhl Seminar on Live Coding, Sam Aaron asked, "Why aren't kids programming?" It's fun, it's creative, and it's an inexpensive activity that's easily accessible. Why isn't it happening?

The problem is one of perception and access. As Simon Peyton-Jones of *Computing At School* has pointed out, most people define "Computer Science" as "a niche university subject for socially-challenged males," when most computer scientists see it as "a foundational discipline, like maths or physics, that every child should learn." In the United States, computer science is one of the least-subscribed topics in science, technology, engineering, and mathematics (STEM) in secondary schools. For the 30,000 high schools in the United States, there are only some 2,000 high school computer science teachers. There is very little access. Even if more students *wanted* to study computer science, it's not available.

Computer science in the United States is predominantly male and white or Asian. The United States is 13.1% Black and 16.9% Hispanic. In most of the states (37 out of 50), fewer than 14 Black students per state took the Advanced Placement Exam in Computer Science (similar to an A-Level) – not percentage, 14 *individuals*. California, the largest and most Hispanic state in the US (38% of population), has a rate of Hispanic students taking the AP CS exam of less than 10%.

How do we improve access and change perceptions about computer science for students?

- First, we create computer science classes that succeed in minority-serving high schools. Jane Margolis at the the University of California Los Angeles has led the way in this strategy with the *Exploring CS* (http://exploringcs.org) that they have had adopted successfully in many California schools.
- Second, we change how we frame and describe computer science. Betsy DiSalvo at Georgia Tech has had success with her Glitch Game Testers project where she engages Black males with video game testing as a way of getting them to work with computing technology in a way that they find attractive. Her results are astounding, with the majority of her students moving on to study computing in higher-education.

Live coding can play a role in these strategies. Live coding gives an audience opportunity to see programming and to see a programmer, perhaps for the first time in their lives. They see programming in a live coding situation not as a solitary, asocial activity, but as a creative endeavor that creates a valuable product, music. Like Glitch, live coding can be engaging and draws students into exploring technology.

We need research to explore the value of live coding in changing perceptions about computer science, in engaging new students in exploring programming, and in the kinds of programming environments that are successful as both live coding environments and successful pedagogical tools. We need to explore how live coding can be adopted by a broad range of students. Teachers must be comfortable with the technology to invite it into their classrooms, so we need to explore how to make live coding inviting for teachers. Live coding can be a useful tool in improving access and changing perceptions about computing.

3.3 Live Coding, Software Engineering and Collaboration

Robert Biddle (Carleton University – Ottawa, CA)

One of the topics of interest in our seminar was software engineering, and I was asked to give a brief overview to help our discussions. Our focus was collaboration, so I decided to focus on software engineering process, and especially on agile software development processes.

I began with the “waterfall” diagram from Royce’s classic paper [9] often used (unfairly [4]) as advocating a strict step-by-step process model. The steps in Royce’s diagram are requirements, analysis, design, coding, testing, and operations. I outlined the issues that arise in a strict flow, such as the need for early look-ahead to coding concerns, or the need to go back to design or analysis when requirements evolve or limitations are recognized. I also discussed the appeal of the waterfall, the vexing, but for some reason enduring, drift toward the strict flow between separated phases that can still be seen today. The role of Taylorism or “scientific management” has been recognized for some time [8] as affecting software process, and that may be the explanation.

This brought us to agile methods, and I introduced the basics of both Extreme Programming (XP) [2] and Scrum [10]. I emphasized the iterative nature of both processes, and explained the connection between iterative processes and collaboration, the way in which a dialog is fostered when work can be begun and then changed or refined subject to feedback and discussion. I emphasized the practices in XP and Scrum that facilitate this beneficial loop. In XP, for example, I discussed the “Planning Game”, having a “Customer On-Site”, the idea of “Test Driven Development”, how “Pair Programming” works, and the role of the

"Daily Standup". For Scrum I discussed the roles, the "Product Owner", the "Team", the "Scrum Master", and the flow from "Backlog" through "Sprint Planning" and "Sprint".

I then moved on to discuss some topics where my research group has focused, typically through field studies of software development teams. The main topic I addressed was collaboration between programmers and interaction designers. Aside from embedded software, this collaboration is crucial to the success of software, but little has been said, either in academic work or in industry reports, about how this collaboration should work. I first discussed the work of Miller about interleaved processes [6], and the success it brought in their application domain. I then outlined our own findings. One common concept we identified was "alignment work", where various practices, some formal, some less so, have the beneficial effect of cross-checking understanding between the groups involved in creating software [3]. Another important finding was the role of artefacts of various kinds in facilitating that alignment: stories, lists, sketches, and so on. These all serve to keep different viewpoints aligned, even as the software begins to take shape [2].

To conclude, I reviewed some of the challenges that arise in encouraging collaboration in software engineering processes. One I identified was that software development needs to connect to multiple other processes: business processes, organizational change, and so on. The nature of software is that it serves other purposes, and it can be difficult to link the software development process in to all the processes that are relevant. The other challenge that I addressed is that raised by other innovative ideas in software development. One, for example, is "software craftsmanship" [5], which emphasizes the technical skills and practices of programmers, and which I feel can de-emphasize the importance of collaboration with interaction designers, customers, and others involved in creating successful software. Another recently prominent idea is that of "lean" processes [7] that, inspired by success in manufacturing, emphases the elimination of waste. Like software craftsmanship, this is a good idea, but can be misunderstood: practices involving iteration and collaboration can too easily be seen as superfluous.

References

1 Kent Beck. *Extreme Programming Explained: Embrace Change.* Addison-Wesley Professional, first edition, 1999.

2 Judith M. Brown, Gitte Lindgaard, and Robert Biddle. Collaborative events and shared artefacts: Agile interaction designers and developers working toward common aims. In *Proceedings of the 2011 Agile Conference*, AGILE'11, pages 87–96, Washington, DC, USA, 2011. IEEE Computer Society.

3 Judith M. Brown, Gitte Lindgaard, and Robert Biddle. Joint implicit alignment work of interaction designers and software developers. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, NordiCHI'12, pages 693–702, New York, NY, USA, 2012. ACM.

4 Craig Larman and Victor R. Basili. Iterative and incremental development: A brief history. *Computer*, 36(6):47–56, June 2003.

5 Pete McBreen. *Software Craftsmanship: The New Imperative.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

6 Lynn Miller. Case study of customer input for a successful product. In *Proceedings of the Agile Development Conference*, ADC'05, pages 225–234, Washington, DC, USA, 2005. IEEE Computer Society.

7 Mary Poppendieck and Tom Poppendieck. *Lean Software Development: An Agile Toolkit.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

8 Hugh Robinson, Patrick A. V. Hall, Fiona Hovenden, and Janet Rachel. Postmodern software development. *Comput. J.*, 41(6):363–375, 1998.

9 W. W. Royce. Managing the development of large software systems: Concepts and techniques. In *Proceedings of the 9th International Conference on Software Engineering*, ICSE'87, pages 328–338, Los Alamitos, CA, USA, 1987. IEEE Computer Society Press.
10 Ken Schwaber and Mike Beedle. *Agile Software Development with Scrum.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

4 Group Discussions

4.1 Critical Engineering and Software Tools

Roly Perera (University of Edinburgh, GB)

The title of this session was inspired by the Critical Engineering Manifesto, *http://criticalengineering.org*.

Codifying Practices in Tools

In human crafts, tools and practices co-evolve. Tools come to assimilate and embody the practices they support; software engineering is no exception. *Test-driven development* (TDD), for example, evolved as a methodology alongside the development of unit-testing tools which support TDD. When considering tooling for live coding, it may be useful to look at other programming domains where time plays a central role, such as animation. We may also look to traditional software engineering for ideas or jargon – such as a *spike*, a functionally "narrow but deep" prototype of a system. Moreover different performance scenarios (say teaching vs. public performance) may imply quite different tool requirements.

Since effective tools co-emerge with patterns of usage, programmers need to be able to "code without fear", unhindered by worries about the consequences of exploratory actions. In current environments, many user actions are transient rather than persistent (i.e. leave no record of what was done), and this and other uncertainties may be a source of anxiety which undermines exploration.

Programming environments for live coding need to be open-ended and extensible, like Smalltalk's *read-eval-print* loop (REPL). Open-endedness means the environment places as few constraints as possible on how it is used. Extensible means that code snippets and other reusable fragments of code – perhaps a library of synthesisers – can be made easily accessible from the programming environment. Such small, composable micro-features may also be a useful granularity for teaching and sharing.

Both performative live coding and other forms of live programming emphasise the importance of short feedback cycles. Anecdotally, any programmer knows that it is often hard to see how something works by just staring at code. But there is a tension here: if short feedback cycles alleviate cognitive load, then necessarily they also reduce the need for skills that manage that cognitive load. And we should not presume that these skills are unimportant to the creative process.

Researchers interested in tools and practices need small, focused research problems – *drosophilae* for live programming research. The form that these easily digestible problems might actually take remains unclear. A programme for live coding research may challenge foundational assumptions about programming languages: whether languages are visual or textual, or even whether tools and languages can be cleanly separated.

New Models of Collaboration

New collaborative tools have the potential to bring live coding to a wider audience, develop communities of live coders, and make the whole endeavour more mainstream, through exposure.

Live coding systems can be less opaque and more communicative than traditional sequencing software like Cubase – more *authentic*, in the sense of conveying what is actually going on. This is due to the plurality of roles for source code in live coding. Primarily it is used by the performer to construct the performance, but importantly it is also often a key *part of the performance*. Observing how the code changes and its relationship to the music can elicit an aesthetic response just as can the auditory component of the music. Moreover it can help the audience appreciate simply that *it is possible* to create such a performance live, which they may not have previously considered. Finally, the code may also be a (partial) *record* of the performance, useful for teaching or other communicative activities after the performance. Thus as with normal programming, the comprehensibility of the code is important, although typically less so for the live audience, where comprehending everything which is going on may not be practical or even desirable.

The performative emphasis of live coding may be able to inform more traditional software engineering. For example music is often learned by observing virtuosity, as in a music masterclass, but this has only recently begun to happen with traditional programming. Online videos in which an expert programmer works through a problem are now common. Indeed performative videos are fast becoming a primary medium for disseminating both practical and theoretical knowledge: see for example the recreational maths videos of Vi Hart of the Khan Academy. Given this potential, recorded coding performances should perhaps be considered a first-class output – part of the "codebase", even.

The internet is transforming not only knowledge dissemination, but collaborative activities in general. New collaborative systems tend to be flatter, decentralised (peer-to-peer) and distributed, rather than hierarchical, centralised and local. GitHub, Wikipedia, and StackExchange are just three paradigmatic examples of the "new school". As new collaborative practices emerge, we should expect these to be supported and codified by new tools. The new structures of course also have shortcomings, just as more traditional collaborative regimes, such as orchestras, have their own unique strengths. Not only that, new structures that start out simple or democratic often become baroque or draconian, as some argue has happened to Wikipedia.

Although the internet democratises along some dimensions, it has the potential to alienate along others, such as gender and computational literacy. American scholar Henry Jenkins has written about the importance of new media literacy to life in a participatory culture. Because of the pace of change, age can be a powerful source of disenfranchisement. Conversely, young people often achieve a very high level of fluency with computer technology, as exemplified by the thirteen million Minecraft users worldwide.

Hero Culture in Programming

Even outside performing arts, communities have their celebrity "performers". In software development, these are usually influential industry practitioners and commentators. Often their role is to *simplify a message*; such figures were instrumental in the adoption of object-oriented programming and agile methods. In programming language research, and the academic community in general, gurus of this kind may be less important.

Another role for a luminary is to act as a "benevolent dictator" on a project, avoiding the pitfalls of design-by-committee. (Fred Brooks, in *The Mythical Man Month*, compared a development team to a surgical team, lead by a Chief Surgeon, although his analogy seems quaint by the standards of modern lightweight methods.)

Every field also needs its visionaries. Recently Bret Victor set out a compelling vision of programming, showcased not through working implementations but via a series of carefully crafted demos. In an influential essay "Learnable Programming", he explicitly distanced himself from a simplistic construal of live coding as the live updating of the output of a program, emphasising instead the importance of being able to "see" all of the computation, and arguing that programming today is like painting with a blindfold on.

As with any performance, a live coding audience focuses their attention on the performer or ensemble. An algorave places the programmer–musician centre stage, as a traditional club night does a DJ. This focus on the virtuoso is potentially in tension with the collaborative and participatory potential of live coding. While some live coders feel that a performance should – in principle at any rate – be reproducible by a member of the audience, others are happy with a clearer distinction between virtuoso performer and audience member.

4.2 Algorithmic Complementarity, or the Impossibility of "Live" Coding

Julian Rohrhuber (Robert Schumann Hochschule für Musik – Düsseldorf, DE)

Origin and Motivation of the Concept

Live coding tries to closely superpose a contradictory pair of entities, both of which are implied in usual understanding of the term the term *program*: a structural description for something to happen in the future and the temporal unfolding of its consequences. By opening up the conventional black boxes in computer use, live programming apparently sets out with the promise to finally allow immediate and complete access or control. I have proposed the term *Algorithmic complementarity*[1] to explicitly name an easily overlooked but fundamental impossibility of making this superposition complete.

The concept of *complementarity* is derived in analogy to complementarity as it has been used in the philosophy of physics and psychology [1, 2, 3] where it refers to a number of phenomena (like subatomic particles or bistable figures) that have in common that their complete description requires two points of view which cannot be fully reconciled. I think the disadvantages of borrowing such a fairly loaded term from a different domain is outweighed by its potential to shine a different light on live programming. It also may point toward the theoretical relevance of this practise in a more general consideration of art and science, which is something that has always been an inspiration for me. [4, 5, 6]

For live coding, I propose that the fundamental impossibility of complete and transparent access to a programmed process is a positive constraint rather than a hindering obstacle.

[1] It was introduced last year at the International Symposium on History and Philosophy of Programming: Algorithmic complementarity. Some thoughts on experimental programming and the history of live coding. Cf. http://www.computing-conference.ugent.be/file/12.

Live coding demonstrates this impossibility in general, and finds local solutions to it, which results in the necessarily variety of existing systems.

Recounting the Dagstuhl Session on Algorithmic Complementarity

At the Dagstuhl seminar, we had one session on algorithmic complementarity, where we tried to better understand the core of the problem (and the conditions under which it ceases to be a problem). In the following, I'll try to recount these steps.

1. Let's assume a very simple algorithm that is formulated as a function of time, like $y = sin(t)$, and that y has non-trivial consequences, like controlling a second, more interesting process.
2. At a certain point in time t_1, the programmer realises that the outcome of this function should be different, e.g. that it should actually have double frequency.
3. She changes the code, e.g. to $y = sin(2t)$.

A live coding system tries to correlate such changes in the description with changes in the running process, treating the code as its most accurate representation. So what should be the appropriate semantics of such a change? This depends on an interpretation of the formalism. There are a number of possibilities, which fall in two classes: state picture and causal picture.

- The *state picture* interprets the change as a change that happens to the state in which the running system is at that very moment (t_1). By consequence, it will either start the process $y = sin(2t)$, setting $t_1 = 0$, or it may instead interpret the change as a parameter change of an assumed implicit parameter a, like in $y = sin(at)$. Thus, the state picture takes the observable reasons into account which motivate a given change. The new code doesn't describe the system from its beginning, though: the causal relation between the process and the program text is fragmented.
- The *causal picture* is interested instead in maintaining the consistency of the description at all times. By consequence, it will retroactively interpret the whole past in terms of the change at t_1. This means that one could restart the program from the beginning and at t_1 one would get the same result. The resulting state of the system, however, may not at all be related to the observation that led to the change in the first place – y may end up to be a very different value, and given that it is connected to some non trivial process, this may make the whole intervention meaningless. Maintaining the causal picture fragments the state picture.

In analogy with the corresponding phenomena in physics, it is argued that the two pictures cannot be reconciled in a single one.

4.2.1 Preliminary Conclusions

In such a way one can conclude (at least this is the hypothesis) that there is an incommensurability of two complementary interpretations of rewriting a part of a running program. As Roly Perera emphasized that when one unfolds all consequences of such a change and keeps all textual changes represented as changes,[2] a different set of problems appears: program and

[2] e.g. in a function like:

$$y = \begin{cases} sin(t) & \text{if } t < t_1 \\ sin(2t) & \text{otherwise} \end{cases}$$

result now are both fragmented. The resulting unfolding becomes an interesting navigational process, which in itself might not be representable on the same level.

Live coding is a non trivial activity anyhow, as in many cases (like sound synthesis), the relation between code and result is necessarily nonobvious – the resulting surprises or frictions with intuition or convention are a large part of its benefit (equally for learning, art, and science). So in a way, algorithmic complementarity is but one aspect of a much broader issue. It seems though as if algorithmic complementarity, if it really exists, is a fairly fundamental obstacle – an empty center perhaps, that prevents live coding from becoming a merely "technical" problem.

References

1 Atmanspacher, Harald, Thomas Filk, and Hartmann Römer. Complementarity in Bistable Perception. See [2], pp. 135–150.
2 Atmanspacher, Harald and Hans Primas (Eds.) *Recasting Reality. Wolfgang Pauli's Philosophical Ideas and Contemporary Science.* New York, 2009.
3 James, William (1890). *The Principles of Psychology. Volume One.* New York.
4 Rohrhuber, Julian, Alberto de Campo, and Renate Wieser. Algorithms today – Notes on Language Design for Just In Time Programming. In *Proceedings of International Computer Music Conference*, Barcelona, pp. 455–458. ICMC, 2005.
5 Rohrhuber, Julian. Implications of Unfolding. In Seifert, Uwe, Jin Hyun Kim, and Anthony Moore (Eds.), *Paradoxes of Interactivity.* Bielefeld, 2008.
6 Rohrhuber, Julian. Das Rechtzeitige. Doppelte Extension und formales Experiment. In Volmar, Axel (Ed.), *Zeitkritische Medienprozesse.* Berlin, 2009.

4.3 Tidal: Representing Time with Pure Functions

Alex McLean (University of Leeds, GB)

Despite being the most common noun in the English language, *time* can be quite difficult to talk about. Sometimes, the subject of time has been purposefully avoided in computer science, its discussion seen as a kind of anthropomorphism of pure mathematics, or simply a matter for optimisation. Over the past ten years, since the development of Live Programming as pedagogic and exploratory method [2], as live performance practice [1], and with the recent exploratory work of Bret Victor, the subject of time has been brought back on the agenda not only in computation but also in the act of programming itself.

A focus of one session was the music representation of Tidal, a Domain Specific Language (DSL) embedded in the Haskell language. Its representation is built upon the following principles;

Time is both cyclic and linear

In music we hear both cyclic repetitions and linear progressions, and so an expressive representation should allow us to work with both conceptions of time, at the same time. In other words, the representation should enable structure to be described at multiple timescales.

Pattern is both continuous and discrete

Our electronic computers, as fundamentally digital technology, naturally lend themselves to discrete representations. But we need not stop there; we experience the world, including music, through mental imagery as well as discrete events, and so our musical representations should allow for continuous shape as well as discrete sequences.

Live coding representations are not just designed for describing music, but also for thinking about, improvising with and composing music

It is not enough to be able to encode any piece of music theoretically, the language of composition needs to be adaptable to a particular style, and extremely quick to work with and explore.

Representations for composing music should be perceptually salient

That is, the encoding of pattern should be structured in a way that could *potentially* be reverse-engineered in Human perception, just by listening to the output. That said, ambiguity is generally important in Music, being open to multiple possible interpretations by the creative listener.

In the following we focus on the first two principles, which are served through the following features of Tidal's representation of time.

```
type Time = Rational
```

Tidal represents time simply as a monotonically increasing, rational number. This time value represents the current metric cycle[3] number, so that a time value of "8/3" would be the point that is two thirds through the third cycle. This straightforward notion is enough to allow us to think about and manipulate time either in progressive linear, or repeating cyclic terms.

Tidal represents both continuous and discrete patterns with the same higher order type signature:

```
type Arc = (Time, Time)
type Event a = (Arc, a)
data Pattern a = Pattern {arc :: Arc -> [Event a]}
```

Simply put, a pattern is a function from time, to events. This allows pattern transformations to be expressed as combinators, which may operate on time separately from the contents of each event, or vice versa.

Because Time is a rational number with arbitrary precision, a Pattern cannot be efficiently represented and queried as a function from a single time value; we would have to sample a pattern through multiple queries to retrieve all the events, and depending on the rate of sampling that we choose, may miss some events. Instead, a Pattern is represented as a function from a time *range*, for which we use the term Arc, in sympathy with the notion of time as a cycle. In practice then, a Pattern is a function from a given Arc of time, to a set of events which occur during that arc. Furthermore, each Event is returned with its own Arc, representing when the Event begins and ends.

[3] In musical terms, a *meter* is the underlying structure of a rhythm.

This representation works well for representing long, complex patterns. Firstly, it allows random access, so that we only need to calculate the particular section of a pattern we are currently interested in. Secondly, it allows patterns to be manipulated without calculating the events themselves, using a higher order combinator function.

Furthermore, by virtue of the use of Arc based queries, this representation works for notionally continuous as well as discrete patterns. For continuously varying patterns, we simply take the midpoint of the Arc of the query as the value to return.

In Tidal, the Pattern type is an instance of the Applicative functor and Monad classes, which allows them to be composed in a highly expressive manner, serving the latter two design principles listed earlier. Visual examples are provided in the paper "The Textural X" [3].

References

1 Nick Collins, Alex McLean, Julian Rohrhuber, and Adrian Ward. Live coding in laptop performance. *Organised Sound*, 8(03):321–330, 2003.

2 Christopher M. Hancock. *Real-time programming and the big ideas of computational literacy.* PhD thesis, Massachusetts Institute of Technology, 2003.

3 Alex McLean. The textural X. In *Proceedings of xCoAx2013: Computation Communication Aesthetics and X*, pages 81–88, 2013.

4.4 Explicit vs. Implicit and External vs. Internal (Representations of) Time

Julian Rohrhuber (Robert Schumann Hochschule für Musik – Düsseldorf, DE)

Ignited by Alex McLean's introduction of his live coding system Tidal, and by the excellent moderation of Thomas Green, there was a spontaneous session on the question of time: how is time represented in different systems and is there a fundamental difference in those representations? On an implementation level, timing can be a complicated issue, not least because it is equally relevant for layers of computation that concern the micro and the macro scale, which each may have very different constraints in terms of efficiency and expressivity.

Implementation itself wasn't so much the topic of the discussion, however. What was more central was the way the variable *time* becomes part of the representation of the relation between a script and a process. It became apparent that:

1. some systems do not make explicit reference to a "state of time", some make use of it excessively, even treating time as a first class object,
2. some systems locate the changing time state outside (usually a single value), some locate it inside (often multiplying the time references).

As the central domain of live coding are temporal phenomena and programs that unfold *over* time, this pair of differences, *explicit time* vs. *implicit time* and *external time* vs. *internal time* has a lot of ramifications in the trade offs between expressive power and expressive clarity. It is no wonder therefore, that the ontologies and computational aspects of time have been focus of numerous publications in the live coding community. Specific solutions were discussed at the seminar, such as cyclic time (which in a sense is an intermediary between changing time and static time), temporal recursion, knitting with time, and multiparadigmatic systems.

Also we exchanged knowledge about different ways of specifying sequences of events, in particular considering the difference between polymetric and polyrhythmic interpretations of a time step.

4.5 Stress and Cognitive Load

Alex McLean (University of Leeds, GB)

It goes almost without saying that live coding music in front of an audience can be stressful. Now that live coding is popular in conference presentations, as well as in educational settings, this issue of stress has wider significance. We formed a large breakout group to explore this issue from a number of perspectives.

Firstly, we noted that cognitive load is not necessarily to be avoided. When participating in music and other creative tasks, we enter a state of 'flow' [2]; deep engagement which requires a happy medium somewhere between frustratingly difficult on one side, and distractingly boring on the other. In other words, to support exploration and curiosity, we want *some* cognitive load, but not so much that things become too stressful to concentrate.

The Cognitive Dimensions of Notations framework [1] is well established in the Psychology of Programming field, and in our discussion we made good use of its terms in talking about the design trade-offs involved in cognitive load. The dimension of *hard mental operations* is one; live coding environments should be designed so that mental processing is targeted on the semantic rather than notational level. This relates to *closeness of mapping*; if the notation corresponds closely to the problem world, i.e. supports Domain Specific Language (DSL), then we are more able to engage the 'right kind' of cognitive load, and become more absorbed in programming as a live experience. Lack of *hidden dependencies*, low notational *viscosity* and of course facilities for *progressive evaluation* are other highly relevant design aims which can be articulated using the cognitive dimensions.

In conventional programming situations, the comprehensibility of a program is of great importance, but in live coding performances the requirements are different: we established that, in the opinion of most of the live coders present, the audience's ability to comprehend the code as it was displayed was not paramount. Thus some of the aspects highlighted by the cognitive dimensions framework had a lesser role than in conventional programming. Nevertheless, the various languages used by live coders differ in many important ways, and the interfaces used to represent and interact with the code are similarly very varied. These large differences between probably have significant impact on how the code evolves in performance and on audience reception. For those interested in notational design *per se* detailed comparisons would be fascinating, but for most members of the live coding community, it seemed that there was more interest in what could be achieved with a particular tool than in abstract comparisons between tools.

We discussed behavioural differences of programmers when coding under the stress of tight time constraints. For example, whereas copy-and-paste is normally seen as bad programming practice, when a programmer only has a matter of seconds to effect a change, then it becomes a more attractive technique. This is especially true in live coding tasks such as improvised music; maintainability is hardly a factor if the code is discarded at the end of a performance.

There is some relationship between cognitive load, and ease of learning. The popularly known “learning curve” of programming is related to the creative path of flow, and to be widely accessible, live coding environments need to support a good path for learners, including by supporting collaborative practices which provide social scaffolds for learning. Indeed, the possibility of learning during performance was raised; creating new functions during preparation, and exploring the possibilities of these functions in front of an audience.

A particularly interesting issue is that understanding your own code is not the same thing as being able to change it. In live coding of music, being able to quickly make a change, and then quickly adjust it in response to the musical result, requires knowing where you can make a change, and what kind of change you can make, but does not require fully understanding everything else about the code. One possible design response that arose is to make it easier to locate the section of code that is producing a particular aspect of the the sounds and patterns you can hear. This is to some extent being explored through visual environments such as SchemeBricks and Texture, but perhaps much more could be done to visualise data flow and functional transformations within the code environment.

References

1. Blackwell, Alan, and Thomas Green. Notational Systems – the Cognitive Dimensions of Notations framework. In *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*, ed. J. M. Carroll, 103–134. Morgan Kaufmann.
2. Nash, C. and Blackwell, A.F. Liveness and Flow in Notation Use. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME) 2012*, pp. 76–81.

5 Practical Implementation Session

5.1 Setting Up a Shared Temporal Frame

David Ogborn (McMaster University – Hamilton, CA)

In a session on Wednesday afternoon, the live coding performers in the group addressed the “infrastructural” issue of a common temporal frame shared via standard networking technologies. There are many systems (ranging from simple ad hoc constructions through to more elaborate, maintained and independent software projects) for synchronizing beats and other types of musical data. The infrastructural issue is not a lack of ways of sharing timing information, but rather an excess of such methods. Each individual, or group, or in some cases broader community, has one or more synchronization techniques. A lack of agreed upon protocols around these techniques hampers collaboration, especially when it is a case of co-performance involving multiple, distinct live coding software environments.

Much of the shared temporal frame session was devoted to exploring and hacking around the EspGrid software. EspGrid was developed by seminar participant David Ogborn in the specific context of a large, live coding laptop orchestra that uses multiple languages/-performance environments (the Cybernetic Orchestra at McMaster University in Hamilton, Canada). EspGrid responds to the multiplicity of performance environments by producing and maintaining synchronization elements independently of those performance environments (it is a separate application, developed in Objective-C and linked against the Cocoa or GNUstep libraries). This separation from the environments in which live coders perform allows

EspGrid to implement synchronization techniques of arbitrary complexity while shielding live coding performers from that complexity. It also allows different synchronization techniques to be tested and compared efficiently, as the specific synchronization technique can be changed independently of the code actions carried out by an individual performer.

While a common and relatively language-neutral software utility like EspGrid addresses some of the challenges of synchronization, it does not in and of itself abstract away the diversity of representations of time and metre. This was directly evident in the hacking that ensued during this session: the live coding performers, working very quickly, attempted to connect their habitual and preferred "internal" representations of time and metre to the public interface presented by EspGrid, with varying degrees of success. While the live coding performers performed this work of adaptation, the other seminar participants collected observations about the nature of the process unfolding before them.

Following the Dagstuhl seminar and informed by its results, a group of seminar participants have continued to work on the definition and dissemination of a common protocol for shared clocks and metres. Both the discussion at Dagstuhl and its later continuation have informed recent updates of the EspGrid software.

6 Live Coding: Some History and Implementations

6.1 Personal Accounts on the History of Live Coding

Renate Wieser (University of Paderborn, DE), Julian Rohrhuber (Robert Schumann Hochschule für Musik – Düsseldorf, DE)

In the following, we briefly sketch a few historical aspects of live coding, bearing in mind that such a description (besides being too short anyway) cannot be satisfactory – after all, as it is said, "the contemporary witness is the natural enemy of the historian"[2] How and when was live coding discovered? If this question that can be answered at all, it is from a personal perspective. Once recognised, however, live coding turns out to have never been entirely absent.

One marker for the recognition was, together with the first publication,[1] the first meeting *changing grammars*, organised at the University of Fine Arts of Hamburg (HfbK) in winter 2004. In a sense, this institution was a fairly natural environment for live coding: computers had already been an integral part of art education for a long time, with Kurd Alsleben as one of the protagonists of early computer art. Various cooperations with the computer science department of Hamburg University existed, as well as a number of computer labs with an emphasis on experimenting, network art and free software.

Certainly, this conference was a decisive moment for the form live coding has today. Firstly, it showed that the idea of rewriting programs at runtime wasn't relevant only to a few individuals, but was a problem with many facets and faces and could lead to many different approaches. This variety has increased ever since and was an important topic at the Dagstuhl seminar. Secondly, it brought together two important understandings of the word "live", which remain central today:

1. programming as *public* thought
2. reprogramming a program at *runtime*.

By delocating programming to unusual places like night clubs[3] and by deliberately entering the paradoxes of using programs to interact with programs, at that time, both aspects had been realised in a rough, radical, and experimental way. What “public” means could have been anything from a large concert to an informal conversation or even an abstract concept.

The further development of live coding in many ways went beyond any expectations at that time. Then already, however, we can recognise a tendency toward the unobvious, weak, hard to convey, which, together with exposing this practice to a partly uninformed public, includes a certain inclination to joke, failure and slapstick. Live coding could have been oriented more toward virtuoso culture, but it seems that the more conceptualist orientation has (gladly) remained till today. Also situating the live programming practice at disciplinary interstices, in particular between art, theory, science and education, was so common that it was hardly talked about – it wasn’t an explicitly transdisciplinary endeavour. The domain of sound, in its non-visual and irreducibly temporal character, was an important condition for this development.The founding of TOPLAP (one surprisingly non-temporary outcome of this meeting) led the way to a broader community and a consciously open definition of the practice and its terminology (including live coding, live programming, just in time programming, conversational programming, interactive programming).

Even if already recognised initially, in the subsequent years, at least two aspects of live coding came to the surface: firstly, the fact that many ideas had been already existing in computer science, but had been occluded by an over-emphasis of the computer as a tool simulation. After the peak of interactive programming in the late 1970s it was almost completely forgotten (while sometimes implicitly practiced) in mainstream computer science. When Adrian Kuhn at Dagstuhl reported from his recent live programming workshop that the participants were either quite young or relatively old, this may point toward the fact that the realisation of the relevance of live programming is indeed a retrospective one.

Secondly, it became rather obvious that the recognition and formulation of the central ideas – programming as public thought and programming as rewriting of running programs – would necessitate and give rise to a multiplicity of different approaches, languages and disciplinary contexts.

References

1 Nick Collins, Alex McLean, Julian Rohrhuber, and Adrian Ward. Live Coding in Laptop Performance. *Organized Sound*, (8):321–330, 2003.
2 Wolfgang Kraushaar. Der Zeitzeuge als Feind des Historikers? Neuerscheinungen zur 68er-Bewegung. *Mittelweg 36*, (8):49–72, 1999.
3 Alex McLean. Hacking perl in nightclubs.

6.2 Extempore

Andrew Sorensen (Queensland University of Technology – Brisbane, AU)

What makes Extempore reasonably unique in the current computer music landscape, is that Extempore is efficient enough, and general enough, to make its own audio-stack implementation available for on-the-fly code modification at runtime. The complete audio stack, right down to the audio device driver callback, is written in Extempore and can be extended, modified, or completely swapped out, on-the-fly at runtime.

Extempore is a general purpose programming environment that hosts two programming languages, Scheme, and XTLang, and a standard library whose growing scope includes, 2D and 3D graphics, audio signal processing, mathematics, I/O and networking, concurrency, high performance computing, abstract data types, etc.

Extempore's Scheme interpreter is a derivation of the TinyScheme interpreter modified for real-time use. Extempore's Scheme interpreter was lifted, in whole, from the Impromptu programming environment, and in this sense is somewhat of a legacy component in the Extempore ecosystem. XTLang is a new, general purpose, high performance programming language designed for both the 'low-level' programming tasks often associated with 'systems' languages like C and C++, as well as the higher level programming often associated with 'scripting' languages. In many respects Extempore derives directly from C, supporting low level type expressivity, first class pointer semantics, manual memory management and a static type system. However, Extempore also borrows heavily from functional languages like ML, supporting first class closures, tail recursion, parametric polymorphism (aka reified generics) and type inferencing. Most importantly XTLang supports the same level of on-the-fly code compilation and runtime substitution commonly expected of so called 'scripting' languages. XTLang uses a homoiconic s-expression syntax similar to Scheme, and supports Common Lisp style macros.

Extempore's two languages are tightly integrated and share two of the environment's most important architectural features – a real-time scheduler for precise control over code start and execution times, as well as a flexible distributed concurrency semantics. These two qualities combine to provide the programmer with imperative control in both time and space. Providing imperative control over wall-clock-time and distribution-in-space makes Extempore well suited for tasks whose domain is the physical environment, with all of the causal consequences that this entails. Extempore's mantra is "real-time in real-time".

As a live coding language, Extempore supports the high-level expressivity favored by many live coding environments, but also provides the low-level expressivity and high-performance required to code down-to-the-metal.

6.3 Fluxus

Dave Griffiths (FoAM – Kernow, GB)

Fluxus is a game engine designed for rapid prototyping and live coding of audio/visual 3D worlds. It's been used for live coding performance, immersive interactive games, architectural visualisations and teaching of music and programming concepts from children in the Brazilian rainforest to teenagers in Barrow in Furness in the UK.

Being built from the Racket language, a dialect of Scheme, Fluxus is an application of programming language research being carried out by the extensive academic community around that project.

Fluxus is also used as a foundation to support the research of new visual programming concepts for live coding performance. Betablocker is an interpreter for live coding assembler in graphical form with a game pad – prototyped in Fluxus and later developed as a Gameboy

DS game. Daisy Chain is a self modifying petri-net live coding language where the performer creates colourful dynamic graphs to sequence musical patterns. Scheme bricks is an interface for programming Scheme graphically, with close integration with audio for performer and audience feedback tightly connecting a bespoke code description and it's sonic results.

In recent years Fluxus has moved into browsers, mobile and games console technology, providing a diversification of live coding into new areas such as citizen science games and re-programmable tools for researchers working in remote locations.

6.4 ixi lang

Thor Magnusson (University of Sussex – Brighton, GB)

ixi lang is a live coding programming language whose interpreter is built in SuperCollider, thus giving access to the underlying power of that environment. However, unlike SuperCollider, the aim with the language is to create expressive constraints. Inspired by operator overloading in C++, the live coding systems of Alex McLean, and esoteric languages such as Whitespace and Brainfuck (see http://www.esolangs.org), ixi lang was designed as a high-level system affording certain types of musical patterns, but excluding others. As such, the system itself becomes a compositional form. Here, constraints inherent in the language are seen as providing freedom from complexity, yet defining a large enough search space for musicians other than the author to explore and express themselves. The language is very simple and intuitive for audience members. Code can be written that changes other code (and updates the code in the same document), which allows for complex structures and changes over time that are not directly called by the performers.

ixi lang affords a specific set of musical activities. It provides a scaffold for externalising musical thinking and, through its simplicity, attempts to ease the live coder's cognitive load. As a live-coding system, it goes further than most common live-coding environments in providing a simple, high-level platform for musical expression. As the system is written in SuperCollider, regular SuperCollider code can be written in the same document, allowing the user to tap into the extensive scope of SuperCollider itself. Learning the affordances of ixi lang as presented in its language constructs might take less than an hour, but getting an overview of the system's constraints can take many long sessions of practice.

6.5 SuperCollider and the Just In Time Programming Library

Julian Rohrhuber (Robert Schumann Hochschule für Musik – Düsseldorf, DE)

A significant technological threshold in sound programming was the moment when sound could be calculated faster than real time: while before the late 1990s, programs resembled factories that would produce sound files, suddenly, a programmer could construct a program that became something like a conversation partner in a musical situation. In such a way, both graphical and physical interfaces became central orientation points in programming.

This significance of this type of interactivity has also structured how the relation between the variable (parameters) and the static (the program) has been conceived of. Today, programming still defines itself mostly as a preparational task in which this relation between change and its boundary conditions is specified.

The programming language SuperCollider [1, 2] was an interesting intervention at this point – its granularity of description of acoustic processes was tuned in such a way that the separation between parameter and program became increasingly porous. Code became concise enough to resemble gestures, poems, or interfaces, so that it became thinkable that the separation between programming activity and music making might have merely been a conventional one that had been kept in place by the existing practice of setting up graphical user interfaces.

JITLib,[3] which has later become integral part of SuperCollider, set out to remove the remaining stumbling blocks which had so far hindered this idea to become an actual practice. While live coding already at that time was fairly easy to implement in a purely imperative world (where state simply is changed), in a mixed world between pure functions and objects, there were a number of problems to solve. The different parts of the library have the purpose to make it possible to experiment with the non-trivial relation between changes in the code to respective changes in sound.[4]

This experimentation has thus been made an integral part of various types of contexts of sound programming: learning, composition, performance, research, exploration. One of the outcomes of integrating the programming process into these activities is a very general insight: because programs are plans, or propositions for the future, changing such a plan in its unfolding cannot be immediate and transparent at the same time. From the beginning, Just-in-time-programming has therefore been a tongue in cheek cover term for the truth: a pro–gram is always too late.[5]

References

1 McCartney, James (1996). SuperCollider: a new real time synthesis language. In *ICMC Proceedings*.

2 McCartney, James (2002). Rethinking the computer music language: SuperCollider. *Computer Music Journal* (26), 61–68.

3 Rohrhuber, Julian and Alberto de Campo (2011). Just in time programming. In Collins, Nick, Scott Wilson, and David Cottle (Eds.), *The SuperCollider Book*. Cambridge, Massachusetts.

4 Rohrhuber, Julian and Alberto de Campo (2009). Improvising Formalisation: Conversational Programming and Live Coding. In Assayag, Gérard and Andrew Gerzso (Eds.), *New Computational Paradigms for Computer Music*.

5 Rohrhuber, Julian, Alberto de Campo, and Renate Wieser (2005). Algorithms today – Notes on Language Design for Just In Time Programming. In *Proceedings of International Computer Music Conference*, Barcelona, pp. 455–458. ICMC.

6.6 Overtone

Samuel Aaron (University of Cambridge, GB)

Overtone is an open source toolkit designed to facilitate the exploration of new musical ideas such as audio synthesis and sampling, instrument design, composition, live-coding and collaborative jamming [2]. It combines the SuperCollider audio synthesis engine, with Clojure, a JVM language with a specific emphasis on concurrency, immutability and functional programming.

Overtone supports multiple simultaneous users and provides first level language constructs to support the safe coordination of concurrent modifications in performance time. This allows new functionality to be hot-swapped over old. Overtone's support for lazy sequences enables infinite streams of events to succinctly model compositional structures such as chord progressions and recursive melodies. First class support for immutable data structures, as well as lock-free concurrency constructs, allow concurrent processing of incoming musical events, including storage and query, without having to block any running musical processes [1]. Finally, being hosted on the JVM provides Overtone with wrapper free access to a wealth of previously written Java libraries for interaction with external systems.

References

1 Samuel Aaron and Jenny Judge *New Possibilities For Social Digital Music-Making Arising From The Storage Of History* Proceedings Of The International Computer Music Conference, pages 228–235, 2012.
2 Samuel Aaron, A F Blackwell, Richard Hoadley and Tim Regan *A principled approach to developing new languages for live coding* International Conference on New Interfaces for Musical Expression, pages 381–386, 2011.

6.7 Republic: Collaborative Live Coding 2003–2013

Alberto de Campo (Universität der Künste – Berlin, DE)

Republic is an extension library for the SuperCollider language designed to allow highly collaborative forms of live coding. In a symmetrical network of participants, everyone can write and evaluate scripts which generate musical processes that can distribute sounds across all machines.

The library tries to minimize the difference between local and delocated events, i.e. between sound played on one's own computer and sound played on any other. Besides code and sound, a third layer of communication is chatting and shouting (i.e. hard-to-miss chat messages), which may serve to discuss the evolving performance, to make suggestions for next directions to take, send previously agreed-on cues, or sometimes try to distract each other.

Republic has been developed for the musical practice of the ensemble powerbooks_unplugged (since 2003), which evolved from a series of workshops called Warteraum ("waiting room") which attempted to collide the concepts of granular synthesis and music in symmetrical networks [1]. It has since been used in academic teaching in several ways: the ensemble Republic111 at the Berlin University of the Arts has been performing based on the Republic infrastructure since 2009, and the authors have given numerous workshops in many different contexts.

Over time, performance practice with Republic has tended towards more use of just-in-time programming as implemented in JITLib [2]: going from layers of independent, free-running shortish processes to fewer running processes and precise interventions into them allow more detailed shaping of the collaborative performance flow. Furthermore, the atomicity of JITLib-style programming has made reuse and adaptation of code lines by other participants easier, more likely to work immediately, and thus more rewarding and enjoyable. This can also lead to considerable uncertainties about causal relations of the concurrent audible processes (who of the delocalised players is responsible for which delocalised stream of sounds?), and about how to attribute authorship of the diverging versions of the scripts being written into and taken from the performance history [3].

It has turned out to be a useful introduction to programming for a wide range of artists: students of media arts, sound engineering, fine arts, music, and other fields often are encouraged to studying programming as an artistic practice they may want to integrate into their working repertoire, but very often show some reluctance, and at times even fear of coding. Reading very short programs that create a structure in time, hearing that structure unfold, and hearing how one's own changes to the code cause corresponding changes in the unfolding musical structure is a very immediate experience of the potential of programming.

By extension, such experiences allow deeper understanding of the role of programming in determining the behaviours of the countless devices and services that pervade today's technological societies. In fact, such experiences can illuminate the value of programming as a cultural technique that is both too important and too interesting to leave it to those only interested in the engineering side of the practice.

References

1 Julian Rohrhuber and Alberto de Campo. Waiting and Uncertainty in Computer Music Networks. In *Proceedings of the ICMC 2004, Miami*, 2004.

2 Julian Rohrhuber and Alberto de Campo. Just in time programming. In Nick Collins, Scott Wilson, and David Cottle, editors, *The SuperCollider Book*. MIT Press, Cambridge, Massachusetts, 2011.

3 Julian Rohrhuber, Alberto de Campo, Renate Wieser, Jan-Kees van Kampen, Echo Ho, and Hannes Hölzl. Purloined letters and distributed persons. In *Music in the Global Village Conference*, Budapest, December 2007.

6.8 Sonic Pi

Samuel Aaron (University of Cambridge, GB)

Sonic Pi is a simple Raspberry Pi based environment designed to introduce basic computer science concepts through the creation of sounds. This approach aims to extend the creative opportunities for young Raspberry Pi users, engaging them through a direct emphasis on ownership and self agency within a programming context. Sonic Pi draws from many of the ideas within Overtone, presenting a text interface for manipulating the SuperCollider audio synthesis engine [1]. The text interface is implemented as a Domain Specific Language embedded within Ruby. Students interact with the language through a simplified text editor that provides two basic controls: play and stop. Upon pressing the play button, the text is evaluated and sound is created. The stop button halts any currently executing process. This simple interaction cycle provides an extremely low barrier to entry for beginners, with their first program being as simple as typing `play 60` , then pressing the play button.

Sonic Pi has been developed through close collaboration with UK school teachers. It is accompanied by a scheme of work targeted towards introductory Computer Science within the newly created curriculum. Sonic Pi has also been used to explore the creative potential of technology within the arts sector. A project with artists, Defining Pi, explored the collaborative extension of the linguistic capabilities of Sonic Pi to directly support new art works. Current work focuses upon adding more liveness to the system, to enable the Raspberry Pi to be treated as a musical instrument, and in doing so bridge the computer science and music curricula.

References

1 Samuel Aaron and Alan F Blackwell *From Sonic Pi to Overtone: Creative Musical Experiences with Domain-Specific and Functional Languages.* FARM 2013, ACM

7 Personal Reflections

7.1 Clickety-Click: Live Coding and Software Engineering

Robert Biddle (Carleton University, CA)

The seminar was an eye-opening experience for me, not in the sense of discovery, but rather of re-discovery, or perhaps more correctly of recognizing the importance of something so commonplace. At the workshop we considered various foci involving Live Coding: music performance, the humanities, computing education, and software engineering. I loved the music performances, and was inspired by the humanities perspectives, and the computing education expositions roused my missionary zeal. But I will not comment further on these. I am going to concentrate on the discussions on software engineering, as I saw them unfold at the workshop.

Agility and Collaboration

By using the term “software engineering”, we did not adopt any strong scoping: not any professional or political positioning. Instead we simply meant computer programming for some purpose. This activity is no longer new, and a few of us have seen it develop over half a century. And it’s hardly a niche activity, and indeed it is now the main work for millions of people, the foundation of giant organizations, and supports the infrastructure of much of the industrial world. But there is still uncertainty about how well we understand the nature of programming as an activity. When we must articulate that nature, we are still quickly drawn to exemplars: engineering, manufacturing, science, craft, and so on.

When the buzz about “agile” software engineering began, what was impressive was how it reflected actual experience. When Kent Beck first talked publicly about Extreme Programming in 1997, it felt refreshing to hear someone talk out loud about programmer experience, rather than invoking a model of what should work, based on some other activity. In truth, however, there has been such reflection throughout the history of programming, and there is so much commentary on the activity of programming that it is hard to keep track. The material of programming allows many potential structures, and it seems the same is true for the activity. Moreover, in all the ecosystems involved, from technology to business, novelty itself appears to have value, even if only for the disruption it causes.

To begin our discussions on live coding and software engineering, we had a brief presentation on software processes, with a particular focus on collaboration and agile software engineering; the presentation is summarized elsewhere in this report (§3.3). The emphasis was on ideas in agile development that facilitate collaboration, such as pair programming, having the customer onsite, rapid cycles of development and acceptance, and willingness to change. Indeed, the subtitle of Beck's book [2] was "Embrace Change". In explaining how the collaboration between customers and programmers might work, one ideal often suggested was if the customer and programmer sat together, the customer could make suggestions and the programmer could implement them immediately for review and exploration. The customer might say "how about . . . ", and the programmer would change the program: "clickety-click". This term, *clickety-click*, came to represent, in our discussions, the special character of software engineering exhibited in live coding.

The Blank Page

Hemingway called the blank page the "white bull". It stares at you, defiant, daring you to attempt writing. In live coding we reflected on how performers typically began with a blank screen. No program was running, and indeed no program existed. We made connections to doing this when teaching, or when demonstrating some programming to colleagues. The intent is to make the context clear and simple, to make it clear that there is no hidden process at work, that everything of interest will be introduced explicitly, and only that code explicitly introduced will be responsible for any behaviour observed. The aim therefore seems to be clarity, but also the establishment of common ground: my computer begins in the same state as yours. This emphasizes an important point for some performers: anyone can do this the same way. This supports the pedagogical aspect of live coding, and some performers articulated that perspective: they want their audience to understand that the process is visible and reproducible: anyone can make music just like this. As the magician says, there's nothing up my sleeve.

We reflected on elements of the magician's showmanship. It is manifestly impressive to audiences that music is created from a blank page. On the other hand, we also discussed the potential for skepticism. After all, a magician makes an effort to demonstrate a lack of hidden mechanisms. But then from nothing they produce scarves, or doves, or rabbits. It is not their intent to show that anyone can do this, but rather suggest to mystify, to demonstrate "magic". Might an audience feel live coding is making a similar claim? Or might they regard the feat as a different kind of demonstration, of mastery of the complicated skills and knowledge involved in programming? This might then resemble other kinds of music performance, where the audience does not learn how to perform similarly, yet does not ascribe the feats to magic, but nonetheless is appreciative of the live demonstration of skill and knowledge.

In considering the effects of the blank screen, we also acknowledged that no screen is ever really blank. There is much there: the computer itself, the programming environment, code libraries and frameworks, audio samples, network access, and all kinds of hardware, software, and content. So if the blank screen beginning demonstrates a beginning from zero, we have to ask what "zero" means. It can still, at least, mean common ground: an understanding that anyone else could start from the same place. If the audience is knowledgeable, then that would work directly. But even if the audience is not so knowledgeable, then the claim might still have some impact: knowing that in principle they might be able to do the same. This kind of rhetorical device can also misdirected, however: the Formula 1 car may be labelled "Renault", but it is not the same Renault one can buy, and might have nothing at all in common, because naming rights can be bought and sold. So the live coding audience may still be skeptical, and this is worth further thought.

This discussion of skepticism raises some issues about how to present the nature of live coding. Live coding is about music and computing, art and science, whereas performers of "magic", despite their appeal, are practicing deception. It seems important for live coding to be understood as eschewing deception. It is important for the audience to recognize this, and also for them to recognize the consequential and essential *vulnerability* of the live coding performer.

Small is Beautiful

In a live coding performance, the screen doesn't stay blank for long. But while the performer may spend the entire performance typing, the size of the code never grows very large. Indeed, it often seems to be stay about the size of the screen, meaning in the range of 20 to 50 lines of code. Sometimes it is longer, but still not very long. Of course, if the screen starts blank, the amount of code will be limited by the duration of the performance, and if a line of code takes a minute to imagine, type, and tweak, then an hour long performance only has at most 60 lines. Yet this still impresses, inspires, keeps an audience rapt. The brevity intensifies the effect, as does beginning from a blank screen, perhaps because the link between cause (the programming) and effect (the music) seems so clear, so immediate. I must admit, I've always liked small programs that do big things. The first programming language I learned, in 1968, was APL, known for enabling complex behaviour in a single line (albeit sometimes a long one). We take more care with the format and understandability of our code nowadays, but brevity still has appeal. Live coding reminds us of that appeal, and that power.

Small programs can do big things. Of course small programs can be made hard to understand – see the work on code obfuscation. But large programs are seldom really easy to understand. With a good supporting language and environment, small programs seem to have advantages in ease of creation, comprehension, correctness, adaptability, and manageability. This doesn't always match up well with ease at the user interaction level: we discussed those early Apple Macintosh programs that were so appealingly easy to use, but concealed complex code in Macintosh Pascal. Will we ever resolve this conflict: can software be simple at both the user and the developer level?

But in live coding there seems to be an emphasis on keeping the operational elements more visible. There seems to be an explicit intent to show the workings while demonstrating their results. We discussed how this was essential when the program or the command line was the only way to interact with computers. While having a higher barrier to understanding, this leads to a deeper understanding. We reflected on how it later became commonplace for computer literacy to become a more superficial kind of literacy. One appeal of live coding, we thought, was the return to that earlier age. Some people even referred to those who grew up in the GUI era as "lost generation", who learned how to use computers without learning how they worked. We also discussed how, on the other hand, graphical interfaces can also be used to expose inner workings: Alan Blackwell talked about his Palimpsest language [5], and Dave Griffiths demonstrated the visual language he uses in his live coding performances (Scheme Bricks) [8].

Reduce, Reuse, Recycle

In a live coding performance, the program is not only created live, but it is then changed over and over again. The program may be small, but it is not static. It seems there are many kinds of changes that are common. One is accretion, where code is added to create more complex behaviour. Indeed, even the initial creation from a blank screen might be seen this way, where the blank screen indicates a running program that simply does nothing; when more code is added, so the behaviour begins to be visible, or rather audible. Sometimes code is taken away, simplified, or deleted, and the music becomes less complex. And this is how some performances end, returning to an empty program and silence. Other performances ended when a change broke the code – those are also important because they show the performance risks are real, and familiar!

Sometimes code is created by being copied from one place and pasted in another. In live coding, the code typically comes from elsewhere in the current program. In real life, code comes from other local files and sometimes from the wild Internet. (This approach has been called "Scrapheap Programming", where search engines help find code to be repurposed, adapted, and fitted into place [9], but this is not typical a live coding performance.) Other functionality typically involves simpler textual operations: we type keystrokes and enter text, can delete and insert, search and replace, and so on. (Of course, there are more refined approaches to reuse, and indeed the history of programming and programming languages can be seen as efforts to advance ways of supporting reuse [3].) We make macros or functions with parameters and call them; we create classes and instantiate them; or objects and clone them; we create frameworks and populate them. And all these also work with accretion: we add code, we rename, we split, we merge, we refactor. And we did see some of this in live coding, albeit on a small scale.

An important thing to notice here is the interplay between the programming language and the environment. They are not really separate, and how well they fit together is a significant usability issue. This important relationship between notation and environment is explored in the Cognitive Dimensions of Notation framework by Green, Petre, and Blackwell, [6, 7, 4]. In live coding this is all at work: the language, the environment used together. The small size of the programs raised some interesting questions. With such small size, do we need such powerful tools? Or are the programs only small because of the power of the tools? I think I saw evidence for both arguments. At the workshop, Thomas Green also emphasized the need to support secondary notation, "visual cues which are not part of formal notation": interestingly we saw little use of that in live coding.

Reflecting on how the code develops and changes in live coding, we discussed the potential role of version control systems such as Subversion or Git. As well as helping manage code in different versions and from different programmers, these tools also make it possible to try out code ideas with the easy ability to change one's mind. In this way they provide a kind of safety net, like an "undo" facility, which in turn encourages exploration. Some of the performers explained that they did use such systems, and one suggested it feels like it allows "programming without fear".

Time and Tide

All programs can change what they do. They do one thing, then another, they select which of several things to do, and they do things over and over. They change what they do depending on input, whether static data or live input. With techniques like reflection, they can change their own code. All of this can happen rapidly.

In live coding, we see a different kind of change. The programmer changes the code while the program is running, based on their human perception of time. The changes happen continually, and always on the human scale of time. It would have been possible for the programmer to create the program up-front to cause the same behaviour, and indeed that would more closely resemble conventional programming. In live coding this is not done. Instead of complex coding done entirely before execution, live coding specifically means coding during execution. But it means more: the coding is part of the execution. The programmer is literally part of the machine.

In one of our sessions at Dagstuhl it was suggested that what performers did was tweaking. It often did seem like that: making small changes to get something just right. Except that it never stayed just right: or "just right" changed as the performance progressed. It was exploratory programming, and the direction of the exploration changed as the performance progressed. The tweaking *was* the performance. The tweaking, the act of seeking "just right" and then moving on: that was the experience being intentionally shared between the performers and the audience.

Live Free or . . .

A live coding performance is compelling, and we discussed the experience as being reawakened to an almost forgotten sense of what programming can be. Live coding speaks to the programming soul. For more conventional software engineering, this raises questions about why this happens and what it suggests for improving the status quo. Our speculative answers to these questions come from the points above. To begin with the last point, much comes from the sharing of the experience of exploration.

As with any live performance, there is an anticipation, a tension, a continual awareness of uncertainty. Some live performances are like tightrope-walking, and the uncertainly is whether the performance will be successful. More relevant to live-coding is the uncertainty about what direction the performance will take. And most relevant is where there is a sense that the performers are communing with each other and with the audience. Having a role in creation, or co-creation, engenders a heightened awareness of what happens. Espen Aarseth suggests this is why games form a kind of "ergodic" medium, because one must work for the results [1].

If the main effect in live coding is the shared experience, it is also important to recognize that the other characteristics all support this main effect. To react is to change, and to react fast is to change fast. To support that, therefore, it is important to be able to change software fast. That means we need the tools, the language and the environment, to make that happen. The mechanisms involved, copy-paste, calling functions with parameters, refactoring, need to be at our fingertips. To making the scoping clear, it helps if our code is small, and to make sure that scoping is evident, there is even an advantage in starting from a blank page.

It is this experience that is possibly the key lesson of live coding for software engineering. In particular, if agile software engineering is about collaborating and about willingness to change, then the kind of shared experience shown in live coding is just what is needed. All kinds of techniques lead in the same direction. In interaction design, low-fidelity prototypes allow us to try things out and change them fast, in software engineering, test-driven development and automated testing, also lead to rapid feedback. It's what made the first spreadsheet software so compelling, because the automatic and rapid recalculation encourages explorations that asked "what if?" It reminds me of Shneiderman's articulation [10] about direct manipulation: "rapid, incremental, reversible operations whose impact on the object of interest is immediately visible." Tanimoto wrote reflectively on this effect and its implications, calling it "liveness"

[11]. It has taken decades, and it seems both necessary and worthwhile to keep looking at this.

To work together we need to be able to make changes fast in response to ideas, we need to be able to experience their effects. How fast? *Clickety-click.* That fast.

References

1 Espen J Aarseth. *Cybertext: perspectives on ergodic literature.* JHU Press, 1997.
2 Kent Beck. *Extreme Programming Explained: Embrace Change.* Addison-Wesley Professional, first edition, 1999.
3 Robert Biddle, Angela Martin, and James Noble. No name: just notes on software reuse. *ACM Sigplan Notices*, 38(12):76–96, 2003.
4 Alan F Blackwell. Ten years of cognitive dimensions in visual languages and computing: Guest editor's introduction to special issue. *Journal of Visual Languages & Computing*, 17(4):285–287, 2006.
5 Alan F. Blackwell and Ignatios Charalampidis. Practice-led design and evaluation of a live visual constraint language. Technical Report UCAM-CL-TR-833, University of Cambridge, Computer Laboratory, May 2013.
6 Thomas RG Green. Cognitive dimensions of notations. *People and computers V*, pages 443–460, 1989.
7 T.R.G. Green and M. Petre. Usability analysis of visual programming environments: A 'cognitive dimensions' framework. *Journal of Visual Languages & Computing*, 7(2):131–174, 1996.
8 Alex McLean, Dave Griffiths, Nick Collins, and Geraint Wiggins. Visualisation of Live Code. In *Proceedings of Electronic Visualisation and the Arts London 2010*, pages 26–30, 2010.
9 James Noble and Robert Biddle. Notes on postmodern programming. In *Proceedings of the Onward Track at OOPSLA*, volume 2, pages 49–71. Citeseer, 2002.
10 Ben Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16(8):57–69, 1983. cited By (since 1996)359.
11 Steven L Tanimoto. Viva: A visual language for image processing. *Journal of Visual Languages & Computing*, 1(2):127–139, 1990.

7.2 Cultural-Epistemological Aspects of Live Coding

Geoff Cox (Aarhus University, DK)

Many of the introductions on the first day of the workshop suggested similar ideas that I addressed in my talk: in celebration of messiness in programming (Biddle); in esoteric and ad hoc language development (Blackwell, Brown, Aaron); and examples like SmallTalk or Scratch (Green, Griffiths, Kuhn); histories of interactive/conversational programming and the collaborative dimension (Rohrhuber, Romero); in emphasising real-time or just-in-time programming (de Campo, Rohrhuber, Sorensen); and the importance of the time dimension of programming and execution (Magnusson, McLean, Perera); as well as more conceptual concerns with issues of control and computational thinking, references to defamiliarisation and new (non-human) aesthetics, and even craft traditions – techne or poeisis (Rohrhuber, Wieser). Indeed live coding embraces the unexpected, the accident and an "aesthetics of failure" (perhaps even what some commentators these days refer to as the "post-digital" condition).

Live programming is the more accepted term in the communities of software engineering and programming language development. Robert Biddle developed the idea of software development and language design through a short history of extreme programming and Scrum methods, and with examples of agile programming, to stress the significance of environments where the machine and programmers are writeable (e.g. executable scratchpad or xiki, a wiki with executable code) and the shift back to text-based languages (command line rather than GUI). It is interesting to note historical comparisons here and possibilities to conceive of programming as breaking with correctness and allowing for the programmer to reflect on how they think and feel while programming. The body is more in the frame, and together with the running program produce an embodied conception of software as a whole.

How do we understand live coding in terms of its genealogy (e.g. TOPLAP), the development of the field, the community that has emerged? This is not simply a development method but something more. Does it constitute a field or discourse? Where does it emerge from in terms of existing disciplines (computer science or music)? What do the analogies to interactive or conversational programming reveal?

Temporality is also clearly a key concern that live coding is able to highlight: a central issue in both computing and music-making but perhaps a neglected one at least in computational cultures. With live coding the source/plan and its execution run at the same time. A focused discussion ran on the issue of time with reference to the paradox that time is both linear and cyclical (cf. §4.3). Does procedural logic allow for a different understanding of time, and perhaps a different understanding of the creative process, where the programmer and program are running in the very dynamic processes of history (cf. Benjamin's "Jetztzeit"/nowness)?

This brings me back to my interest to further emphasise the political and epistemological aspects (someone mentioned Turkle's "epistemological pluralism"): that live coding produces new knowledge through artistic and technical experimentation; and open up analogies to other notation practices more broadly (where scores and execution have resonances in terms of action and agency). Social context also is important, inasmuch as new formations of publicness can be perceived – here making reference to the writings of Hannah Arendt and her understanding of the political. In Two Bits, Chris Kelty argues that the free software movement is an example of what he calls a "recursive public", extending Arendt's definition of a public through speech and action, to incorporate technical and legal infrastructures. Publicness is constituted not simply by speaking, writing, arguing, and protesting but also through modification of the domain or platform through which these practices are enacted. Live coding seems to fit the description and recombinations of human and non-human (or more-than human) bodies, and new materialisms.

7.3 Dagstuhl: Live Coding and Memorable Discussions

Dave Griffiths (FoAM – Kernow, GB)

Our seminar included people from the fields of Software Engineering, Computer Science Education as well as plenty of practising live coders and multidisciplinary researchers.

Discussion was wide ranging and intense at times, and the first job was to sufficiently explain what live coding actually was – which turned out to require performances in different settings:

1. Explanatory demo style live coding: talking through it as you do it.
2. Meeting room coffee break gigs: with a closely attentive audience.
3. The music room: relaxed evening events with beer and wine.

So Dagstuhl's music room was immediately useful in providing a more "normal" live coding situation. It was of course more stressful than usual, knowing that you were being critically appraised in this way by world experts in related fields! However it paid off hugely as we had some wonderful interpretations from these different viewpoints.

One of the most important for me was the framing of live coding in terms of the roots of software engineering. Robert Biddle, Professor of Human–Computer Interaction at Carleton University put it into context for us. In 1968 NATO held a "Software Components Conference" in order to tackle a perceived gap in programming expertise with the Soviet Union.

This conference (attended by many of the "big names" of programming in later years) led to many patterns of thought that pervade the design of computers and software – a tendency for deeply hierarchical command structures in order to keep control of the arising complexity, and a distrust of more ad hoc solutions or any hint of making things up as we go along. In more recent times we can see a fight against this in the rise of agile programming methodologies, and it was interesting to look at live coding as a part of this story too. For example it provides a way to accept and demonstrate the "power to think and feel" that programming gives us as humans. The big question is accessibility, in a ubiquitously computational world – how can this reach wider groups of people?

Ellen Harlizius-Klück works with three different domains simultaneously – investigating the history of mathematics via weaving in ancient Greece. Her work includes live coding, using weaving as a performance tool – demonstrating the algorithmic potential of looms and combinations of patterns. Her work exposes the hidden shared history of textiles and computation, and this made a lot of sense to me as at the lowest level the operations of computers are not singular 0s and 1s as is often talked about, but actually in terms of transformations of whole patterns of bits.

Mark Guzdial was examining live coding through the lens of education, specifically teaching computer science. The fact that so many of us involved in the field are also teaching in schools – and already looking at ways of bringing live coding into this area, is noteworthy, as is the educational potential of doing live coding in nightclub type environments. Although here it works more on the level of showing people that humans make code, it is not a matter of pure mathematical black boxes – that can be the ground breaking realisation for a lot of people.

Something that was interesting to me was to concentrate on live coding as a specifically musical practice (rather than also a visual one) as there are many things about perceiving the process with a different sense from your description of it that are important. Julian Rohrhuber pointed out that "you can use sound in order to hear what you are doing" – the sound is the temporal execution of the code and can be a close representation of what the computer is actually doing. This time-based approach is also part of livecoding working against the notion that producing an "end result" is important, Juan A. Romero said that "if you're livecoding, you're not just coding the final note" – i.e. the process of coding is the art form.

In terms of a school teaching situation sound is also powerful, as described by Sam Aaron, live coder and creator of Sonic Pi. A child getting a music program to work for the first time in a classroom is immediately obvious to everyone else – as it is broadcast as sound, inspiring a bit of competition and ending up with a naturally collaborative learning experience.

It is impossible to cover all the discussions that we had. It was a great opportunity to examine what live coding is about now in relation to other practices, where it came from, and where it might go in the future.

7.4 Live Coding, Computer Science, and Education

Mark Guzdial (Georgia Institute of Technology, US)

Most of the attendees were live coders, but there were a number of us others who helped explore the boundary disciplines for live coding. The seminar explored the ramifications and research potential of this activity.

Robert Biddle was there to lead discussions about the software engineering implications of live coding. On the one hand, live coding feels like the antithesis of software engineering, or as one attendee put it, "an ironic response to software engineering." There is no test-driven development, no commenting, no development of abstractions (at least, not while live coding), no exploration of user needs. On the other hand, live coding (not necessarily with music) can be an important part of exploring a space. One could imagine using live coding practices as part of a conversation with a user about needs and how the programmer understands those needs.

Geoff Cox led a conversation about the humanities research directions in live coding. Geoff has a particular interest in labor, and he pointed out how live coding surfaces hidden aspects of the labor in modern society. While computing technology has become ubiquitous in the developed world, few people in our society have ever seen source code or a programmer. What does it mean for an audience to see an embodiment of a programmer, to see the labor of generating code? What's more, the audience is seeing code doing something that is not normally associated with people's notions of what code is for – making music. How does this change the audience's relation to the computing technology? The notion of programming-as-performance is an interesting and novel way of thinking about computing practice, and in sharp contrast to stereotypical perspectives of programming.

Thomas Green, Alan Blackwell, and others from the PPIG (Psychology of Programming Interest Group) community pointed to the notations that the live coders used. I've drawn on Thomas's work on the cognitive dimensions of programming and the implications of our programming representations since I was a graduate student. The language constraints for live coding are amazing. Live coders tend not to use traditional development languages like C++ or Java, but instead work in Scheme, Haskell, and a variety of domain-specific languages (like SuperCollider) – often building their own implementations. Live coders need languages that are expressive, provide for the ability to use techniques like temporal recursion, are concise, and (as one live coder put it) "let me live code at 2 am at a dance club after a couple of beers."

I was there to connect live coding to computing education, and learned about many connections from the seminar. I am fascinated by the humanities questions about introducing source code and programmers to a technologically-sophisticated audience that probably never saw the development of code. Also, I am interested in the value of rapid feedback (through music) in the performance. Does that help students understand the relationship between the code and the execution?

A Playful Live Coding Practice to Explore Syntax and Semantics

Three of the nights of the Dagstuhl Seminar on Live Coding included performances. Several of these combined live coders with analogue instruments (guitar, piano, cello, and even kazoo), which was terrific to watch.

I found one of their practices fascinating, with real potential for the computer science classroom. Alex McLean introduced it as "Mexican Roulette", because they first did it at a live coding event in Mexico City. Live coders take turns (the roulette part) at a shared computer connected to speakers at the front of the room.

- The first live coder types in some line of code generating music, and gets it running. From now on, there is music playing.
- The next live coder changes the code any way she or he wants. The music keeps playing, and changes when the second coder then evaluates the code, thus changing the process. Now the third coder comes up, and so on.
- If a live coder is unsure, just a few parameters might be changed.
- If a live coder makes a syntax error, the music continues (because the evaluation that would change the process fails), and the next coder can fix it.
- If a live coder makes a mistake (at one point, someone created quite a squeal), the next live coder can fix it. Or embellish it.

What I found most promising about this practice is that (to use Briana Morrison's phrase for this) nothing is ever *wrong* here. The game is to keep the music going and change it in interesting ways. Responsibility for the music is shared. Mistakes are part of the process, and are really up for definition. Is that a mistake, or an exploration of a new direction? This activity encourages playing with syntax and semantics, in a collaborative setting. It relies on the separation of program and process – the music is going, while the next live coder is figuring out the change. This could be used for learning any language that can be used for live coding.

Live Coders Challenge Computer Science to Think About Expression Again

Live coders think about and talk about expression, as evidenced from the conversations at Dagstuhl. They build their own languages and their own systems. They talk about the abstractions that they are using (both musical and computational, like temporal recursion), how their languages support various sound generation techniques (e.g., unit generators, synthesized instruments, sampled sounds) and musical styles. If you look at the live coders on the Dagstuhl Seminar participant list, most of them are in music programs, not computer science. Why are the musicians more willing to explore expressive notations than the computer scientists?

Lisp is alive and well in live coding. I now have a half-dozen of these systems running on my laptop. Overtone is a wonderful system based in Clojure (http://overtone.github.io/) Andrew Sorensen's Impromptu was in Scheme, as is his new environment Extempore. Andrew said that he could build 90% of this in Impromptu, but the low-level bits would have to be coded in C. He was not happy with changing his expressive tools, so he created Extempore whose lowest level parts would be compiled (via LLVM) directly to machine code. Andrew went to this effort because he cares a lot about the expressiveness of his tools.

Not everything is s-expressions. Thor Magnusson's ixi lang is remarkable. I love how he explores the use of text programming as both a notation and a feedback mechanism. When he manipulates sequences of notes or percussion patterns, whatever line he defined the sequence on changes as well.

Tidal from Alex McLean is a domain-specific language built on top of Haskell, and his new Texture system creates more of a diagramming notation. Dave Griffiths has built his live coding environment, Fluxus, in Racket which is used in Program by Design and Bootstrap CS education projects. Dave did all his live coding at Dagstuhl using his Scheme Bricks, which is a Scratch-like block language that represents Scheme forms.

Education Research Questions Around Live Coding: Vygotskian and Non-Constructionist

I saw four sets of computing education research questions in live coding. These are unusual research questions for me because they are Vygotskian and non-Constructionist.

Live coding is about performance. It is not an easy task. The live coder has to know their programming language (syntax and semantics) and music improvisation (e.g., including listening to your collaborator and composing to match), and use all that knowledge in real-time. It is not going to be a task that we start students with, but it may be a task that *watching inspires* students. Some of my research questions are about what it means to watch the performance of someone else, as opposed to being about students constructing. I've written before about the value of lectures (http://computinged.wordpress.com/2010/07/27/in-defense-of-lecture/), and I really do believe that students can learn from lectures. But not all students learn from lectures, and lectures work only if well-structured. Watching a live coding performance is different – it's about changing the audience's *affect* and *framing* with respect to coding. Can we change attitudes via a performance?

Vygotsky argued that all personal learning is first experienced at a social level. Whatever we learn must first be experienced as an interaction with others. In computing education, we think a lot about students' first experience programming, but we don't think much about how a student first sees code and first sees programming. *How can you even consider studying a domain whose main activity you have never even seen?* What is the role of that coding generating *music*, with cultural and creative overtones? The social experience introducing computing is important, and that may be something that live code can offer.

Here are four sets of research questions that I see:

1. *Making visible.* In a world with lots of technology, code and programmers are mostly invisible. What does it mean for an audience to see code to generate music and programming as a live coder? It is interesting to think about this impact for students (does it help students to think seriously about computing as something to explore in school?) and for a more general audience (how does it change adults' experience with technology?)
2. *Separating program and process.* Live coding makes clear the difference between the program and the executing process. On the first day, we saw performances from Alex McLean and Thor Magnusson, and an amazing duet between Andrew Sorensen at Dagstuhl and Ben Swift at the VL/HCC conference in San Jose using their Extempore system. These performances highlighted the difference between *program* and *process*. The live coders start an execution, and music starts playing. Meanwhile, they change the program, then re-evaluate the function, which changes the process and the music produced. There is a gap between the executing process and the text of the program, which is not something that students often see.

3. *Code for music.* How does seeing code for making music change student's perception of what code is for? We mostly introduce programming as engineering practice in Computer Science class, but live coding is pretty much the opposite of software engineering. Our biggest challenges in Computer Science Education are about getting students and teachers to even *consider* computer science. Could live coding get teachers to see computing as something beyond dry and engineering-ish? Who is attracted by live coding? Could it attract a different audience than we do now? Could we design the activity of live coding to be *more attractive* and accessible?
4. *Collaboration.* Live coding is a collaborative practice, but very different from pair programming. Everybody codes, and everybody pays attention to what the others are doing. How does the collaboration in live coding (e.g., writing music based on other live coders' music) change the perception of the asocial nature of programming?

I will end with an image that Sam Aaron showed in his introduction, a note that he got from a student in his Sonic Pi class: "Thank you for making dull lifeless computers interesting and almost reality." That captures well the potential of live coding in computing education research – that activity is interesting and the music is real.

7.5 Algorithms, Live Patterning and the Digital Craft of Weaving

Ellen Harlizius-Klück (University of Copenhagen, DK)

From the perspective of the humanities, in my case: the history of textile technology, live coding offers interesting perspectives on questions I generated in my research on patterning textiles in antiquity. In both cases, craftsmanship is a category that is to define within the research. The sociologist and philosopher Richard Sennett introduced the idea of software as craftsmanship in a first chapter of his book on the craftsman. As we heard from Robert Biddle, this idea was already strong in software engineering, long before, but not comparable with what live coders understand, when they use this term. Interestingly, Sennett subtitled one of his chapters "Ancient weavers and Linux Programmers". However, he rarely says anything about what they have in common.

Margaret Boden's book on Artificial Intelligence is more instructive in this respect, describing algorithms as executing knitting notations. But described like this, even a cooking recipe is an algorithm and the knitting example does not really convince of a similarity of knitting and computers executing programmes on a structural level.

My interest lies just in this: demonstrating that weaving was a digital and even dualistic craft from the very beginning and that the patterning principles can tell us a lot about the prediction or impossibility of prediction of patterns even if we fully control the production process. I am therefore looking for tools to code textile patterns, and live coding with such a tool would enable me to show the production of patterns and their changes on the fly when demonstrating my topics to an audience. It was interesting to see some coders already working with the ideas of threads, textures (Alex McLean) and patterned weaves (Dave Griffiths). Live coding of textile patterns at the moment can only be done with very complicated programmes that are designed for the use of modern looms and for designers who, on the level of sketching, work with vector graphics instead of the dualistic and discrete patterns that are the basis of weaves. Coding patterns on a principal level can help me in showing that such patterns are not flat but an interference effect of a three dimensional structure and a special colour order of its elements. Live coding of textile patterns also makes the algorithmic nature and the involvement of number ratios visible.

From the seminar I learned that live coding could be a way of thinking publicly, as artwork today mainly does. And it can change our perception of people working with computers as it was demonstrated by the quote that Renate Wieser presented: "Art removes objects from the automatism of perception in several ways."[1] Today programmes project the impossibility of liveliness into the future. We need a counterbalance against this attitude, a counterbalance that does not reject programming as a whole but develops possibilities of liveliness in a surrounding of codes and programmes. Also here, doing live coding as an art form, can contribute. As Renate Wieser said: Artists are stopping habitual processes and not starting new habits (which science does). Live coding therefore can be a possibility of exploring the relationship between liveliness and automation. Live coding also implies a critical perspective on engineering as the most influential language of our times. And with live coding of pattern weaving I try to make room for women in the history of computer and software development.

References

1 Victor Shklovsky. *Art as Technique.* 1917, 10 f.

7.6 A Few Notes from the Seminar

Julian Rohrhuber (Robert Schumann Hochschule für Musik – Düsseldorf, DE)

The forgotten dimension of programming which turned out to be only half forgotten

Some historical notes: In the first years, live coding developed within a specific mixed context between experimental arts, club music, philosophy, and interdisciplinary scientific research. It grew where people worked together as musicians, as artists, but also as researchers in interdisciplinary science projects. It was immediately part of a teaching curriculum, or rather its self-recognition as something special came from learning exchanges, workshops, collaboration, conversation art, lecture performance.

In such a way, computer science education has been its driving motor from the beginning – the fact that the forms and topics of teaching didn't resemble those of informatics courses shouldn't obscure the insight that this was collaborative learning efforts that centred on topics like higher level functions, polymorphism and networking protocols. From the programming languages involved, this practice inherited knowledge (both tacit and explicit) from a long history of programming languages.

A note regarding the visibility of programming: the necessity to expose the programming activity, as well as to see the programming activity as a part of either a performance or the computational process was far more obvious in this transdisciplinary context than it was for those who had contemporary training in computer sciences. As it was said a couple of times during the discussion in the seminar, the abstract and elusive character of the domain (namely sound in the art and sound as a means to explore scientific data) made it both more fruitful to do live coding (or just in time programming, as we usually call it) and more essential to achieve anything interesting in the first place. As far as education concerns: this context made it very obvious that learning itself requires the experience of programming, of thinking a domain (like sound) through the medium of a program.

Another point implied in the discussions in the seminar: programming always has a domain, a domain that is addressed (to some degree) in computer science and computer science education. In the case of pure theoretical computer science, this domain is mathematical, or formal structures. In many other cases, the domain is a mixed field of knowledge, that may involve fields as diverse as music, astronomy or accounting. Invariably, however, programming means speaking doubly, to a machine and to other humans. This is I think what Robert Biddle showed to be a constant challenge in the relation between programmer and domain expert, and also between programmers. In how far is it this involvement in a domain through the activity of programming itself, that live coding makes evident?

Maybe it could be summarised as follows: Live coding was discovered (or, if to a degree rediscovered) by computer science and by computer science education – but not in their respective disciplinary institutions. Reintroducing it now to those institutions, the question is: should this happen as a transfer of a technical tool-idea or should this also transfer at least parts of the philosophical, methodological, and institutional backdrop that was necessary for this idea to proliferate? Both seem possible futures.

Chess players

Taking up Geoff Cox' thought: live coding brings to the surface the social and labour dimension of programming: programs, like texts, are written, often in a tedious and vacillating process, involving decisions both rational and aesthetic, sometimes practical and ad hoc. Showing writing (in the broad sense) reveals the "agency behind" a program, the agency that allows to make plans rather than just to act. In such a way, live coding shows abstraction in the form of delegation (this may be just a sort of steering, if it is a more immediate translation, this may also be a pro–gramming proper in the sense of involving in changing planned setups). On the other hand, what becomes visible also is that this labour is not reducible to any core of direct action or intention, it is rather already in itself a mediated process, involving human actions, but actions which only make sense in the context of a system, a programming language, its abstract but sensual semantics, its syntactic resistance and fluency, its ability to speak about itself, to be understood. In such a way, in our context, van Kempelen's fake chess robot is half the truth: inside the hollow case, the dwarf is sitting, certainly, but what makes the dwarf a chess player? Only the context of the chess system, and the fact that the audience understands this context makes the automaton so fascinating.

In a discussion with Thomas Green, which couldn't be continued, two understandings of the term "plan" came up in the context of a chess game: Thomas referred to a plan as something that a human skilled player has in mind. I pointed out that in a program, contrary to the chess board, plans are laid out and become readable as plans. I should add now that in a chess constellation, a skilled player can certainly decipher possible plans; but it would arguably defeat the purpose of the game if all this were explicit. Thomas disagreed: what we see in a program is not a plan. The sociological and psychological semantic dimension of a plan may indeed be something entirely different than the formalised aspect of a plan which we can read in a program. Nevertheless, if the agency of planning should really play a role in live coding (and I think it does), it is a particular indistinguishability between those aspects that define live coding as something different than playing an instrument or a chess game. The fact that this agency of planning is partly obscure, potentially to the audience but also to the programmer – for instance when a program becomes readable only retroactively – is one of the reasons why it is so interesting.

Participants

- Sam Aaron
University of Cambridge, GB
- Robert Biddle
Carleton Univ. – Ottawa, CA
- Alan Blackwell
University of Cambridge, GB
- Andrew R. Brown
Griffith Univ. – Brisbane, AU
- Luke Church
University of Cambridge, GB
- Geoff Cox
Aarhus University, DK
- Alberto de Campo
Universität der Künste – Berlin, DE
- Thomas Green
University of York, GB
- Dave Griffiths
FoAM – Kernow, GB
- Mark Guzdial
Georgia Inst. of Technology, US
- Ellen Harlizius-Klück
University of Copenhagen, DK
- Shelly Knotts
Birmingham, GB
- Adrian Kuhn
University of British Columbia – Vancouver, CA
- Thor Magnusson
University of Sussex - Brighton, GB
- Alex McLean
University of Leeds, GB
- David Ogborn
McMaster Univ. – Hamilton, CA
- Jochen Arne Otto
ZKM | Center for Art and Media Karlsruhe, DE
- Roly Perera
University of Edinburgh, GB
- Julian Rohrhuber
Robert Schumann Hochschule für Musik, DE
- Juan Gabriel Alzate Romero
Hochschule für Musik – Karlsruhe, DE
- Uwe Seifert
Universität Köln, DE
- Andrew Sorensen
Queensland University of Technology – Brisbane, AU
- Jan Kees van Kampen
Robert Schumann Hochschule für Musik, DE
- Renate Wieser
Universität Paderborn, DE

Report from Dagstuhl Seminar 13391

Algorithm Engineering

Edited by

Andrew V. Goldberg[1], Giuseppe F. Italiano[2], David S. Johnson[3], and Dorothea Wagner[4]

1 **Microsoft Research – Mountain View, US,** goldberg@microsoft.com
2 **Università di Roma "Tor Vergata", IT,** italiano@disp.uniroma2.it
3 **AT&T Research – Florham Park, US,** dstiflerj@gmail.com
4 **KIT – Karlsruhe Institute of Technology, DE,** dorothea.wagner@kit.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13391 "Algorithm Engineering". The *algorithm engineering approach* consists of a cycle of algorithm design, analysis, implementation, and experimental evaluation, with the aim of bridging the gap between theory and practice in the area of algorithms. This cycle of phases is driven by falsifiable hypotheses validated by experiments. Moreover, real-world instances often have direct impact on this cycle since they often expose modeling and analysis shortcomings. Algorithm engineering touches other research areas such as algorithm theory, combinatorial optimization, computer architecture, parallel and distributed computing, high-performance computing, and operations research. Prominent success stories in algorithm engineering include the linear program solver CPLEX, the traveling salesman suite CONCORDE, speed-up techniques for shortest paths computation, for example, in route planning, as well as graph partitioning and the computation of Steiner trees. All these topics are driven by the need for efficient algorithms and libraries for problems that appear frequently in real-world applications.

In the last fifteen years, this approach to algorithmic research has gained increasing attention. There is an ACM Journal on Experimental Algorithmics and several annual conferences (WAE/ESA applied track since 1997, Alenex since 1998, and WEA/SEA since 2001) and the series of DIMACS implementation challenges where people meet to compare implementations for a specific problem. From 2007 to 2013 the German Research Foundation also funded a special priority program on algorithm engineering (DFG SPP 1307).

Seminar 23.–27. September, 2013 – www.dagstuhl.de/13391
1998 ACM Subject Classification E.1 Data Structures, E.2 Data Storage Representations, F.2 Analysis of Algorithms and Problem Complexity, G.2 Discrete Mathematics
Keywords and phrases Algorithm Engineering, Science of Algorithmics, Manycore Algorithms, Certifying Algorithms, Web Search, Large Graphs
Digital Object Identifier 10.4230/DagRep.3.9.169
Edited in cooperation with Andrea Kappes

1 Executive Summary

Andrew V. Goldberg
Giuseppe F. Italiano
David S. Johnson
Dorothea Wagner

Topics of the Seminar

The seminar covered all methodological aspects of algorithm engineering. Examples are the scientific method in algorithmics, the use of modern computer architecture in algorithmics, and certifying algorithms. These aspects were also addressed in dedicated discussion sessions.

Science of Algorithmics

One aspect of algorithm engineering is the *scientific method*, where research on algorithms is interpreted as in other disciplines such as physics and life sciences: the observation of a phenomenon that is not yet understood is investigated via falsifiable hypotheses as explanations of the phenomena, and experimental evaluations to test these hypotheses. That way not only empirical evidence on the behavior of algorithms is attained but also new theoretical insights are sought. Experimental algorithmics is already a core component of algorithm engineering from its very beginning. However, the design of reasonable experiments, the use of meaningful test instances, and reproducibility of experiments are still issues to be discussed in order to derive a common understanding and agree on a best practice.

Manycore and GPU Algorithms

Exploiting the full potential of a *modern computer* poses many interesting new challenges for algorithm engineering: ever increasing parallelism, deep memory hierarchies, and heterogeneous architectures. Algorithms should be tailored to utilize multiple cores, but also access memory efficiently, taking into account issues such as data locality. Nowadays the use of GPUs, which are increasingly common in modern servers, is an important issue for efficient algorithm implementation. This is in particular interesting for frequently used and "classical" algorithms.

Certifying Algorithms

An effective way to ensure correct results of algorithm implementations are *certifying algorithms*. The idea is to check each returned result for correctness using a simple checker. It then suffices to test or perhaps verify the checker. Making checking fast implies interesting algorithmic questions when checking is aided by certificates of correctness computed by the main algorithm.

Focus Topic: Web Search and Large Graphs

Experiences from previous Dagstuhl seminars showed that the interaction between different scientific communities stimulates methodological discussions. This exchange is in particular important for neighboring scientific communities who typically meet at separate conferences.

For this seminar, we focus on *web search, large graphs and social networks* in order to also address the scientific WWW and Social Media community. In these fields, methods from algorithm engineering are applied. However, these scientists typically don't publish at the algorithm engineering conferences mentioned above but meet and publish at conferences like the "International World Wide Web Conference", the "ACM Conference on Hypertext and Hypermedia" or the "International AAAI Conference on Weblogs and Social Media".

Search engines work with a large amount of data, making high-performance algorithms and data structures very important. Relevant problems include fast indexing, text and query processing, and relevance computation. The latter involves a large web graph. Web-enabled applications give raise to other large graphs, such as social networks, like "friend graphs" or e-mail graphs induced by message origin-destination pairs. Algorithms on such graphs are of great interest. For example, identifying interest-based sub-communities (e.g., classical music fans) enables better service experience or contextual advertisement.

Aims

The aim of this seminar was to bring together researchers with different backgrounds, e.g., from algorithm and datastructures, computational geometry, combinatorial optimization, parallel algorithms and algorithm engineering in order to strengthen and foster collaborations and to identify key research directions for the future. In particular, the seminar was intended to foster the exchange between algorithm engineering and scientists from the web search community. While the dominant goal of the seminar was the exchange of current research developments and discussion of topical subjects, it also contributed to bring algorithm engineering forward as a still evolving and expanding field in computer science. The seminar program included four dedicated discussion sessions on methodological questions, as well as research related issues like future DIMACS Implementation Challenges.

Conclusion

The organizers decided to schedule talks and discussions not grouped according to topics but provide a vivid mix of different research questions and results. According to the composition of the seminar participants, not all topics were covered equally well. For example, certifying algorithms were not addressed in detail. On Monday, Renato Werneck gave a short report on the "11th DIMACS Implementation Challenge on Steiner Tree Problems"[1] taking place in 2013/14. The program of Monday afternoon was concluded by a panel discussion on "Empirical and Theoretical Approaches to Algorithm Design: Synergies and Opportunities". The second panel discussion on "Benchmarks and reproducibility of experiments" took place on Tuesday. The third panel discussion on Thursday focused on "Promoting and advancing the field" and on Friday a discussion about "Teaching Algorithm Engineering" concluded the program.

The seminar hosted 39 participants. Besides presentations and panel discussions the program offered room for bilateral discussions and working groups. Schloss Dagstuhl and its staff provided a very convenient and stimulating environment. The seminar participants appreciated the cordial atmosphere which improved mutual understanding and inspiration. The organizers of this seminar wish to thank all those who helped make the workshop a fruitful research experience.

[1] http://dimacs11.cs.princeton.edu/home.html

2 Table of Contents

3 Overview of Talks

3.1 Algorithm Engineering Issues in Semantic Search

Hannah Bast (Universität Freiburg, DE)

I will talk about our research on semantic search over the past years, with an emphasis on the algorithm engineering perspective.

3.2 Engineering Little Algorithms into a Big System

Jon Bentley (Avaya – Basking Ridge, US)

This talk describes how a few researchers worked closely with a development team to improve the performance of an old yet very important system. It starts with an overview of the project, and then concentrates on the particular technical problem of cache design and implementation (including an interesting study of hashing). This interaction between research and development illustrates some old yet fundamental truths: defining the right problem can be most of the battle, and a little technology, properly applied, can make a big difference. (This talk describes joint work with Duffy Boyle, P. Krishnan and John Meiners.)

3.3 New Algorithms for Computing the Greedy Spanner

Kevin Buchin (TU Eindhoven, NL)

Joint work of Alewijnse, Sander P.A.; Bouts, Quirijn W.; ten Brink, Alex P.; Buchin, Kevin;
Main reference Sander P. A. Alewijnse, Quirijn W. Bouts, Alex P. ten Brink, Kevin Buchin, "Computing the Greedy Spanner in Linear Space", in Proc. of 21st Annual Europ. Symp. on Algorithms (ESA'13), LNCS, Vol. 8125, pp. 37–48, Springer, 2013; also available as pre-print on arXiv (arXiv:1306.4919v1 [cs.CG]).
URL http://dx.doi.org/10.1007/978-3-642-40450-4_4
URL http://arxiv.org/abs/1306.4919v1

The greedy spanner is a high-quality spanner: its total weight, edge count and maximal degree are asymptotically optimal and in practice significantly better than for any other spanner with reasonable construction time. Unfortunately, all known algorithms that compute the greedy spanner of n points use $\Omega(n^2)$ space, which is impractical on large instances. To the best of our knowledge, the largest instance for which the greedy spanner was computed so far has about 13,000 vertices.

We will present an $O(n)$-space algorithm that computes the same spanner for points in R^d running in $O(n^2 log^2 n)$ time for any fixed stretch factor and dimension. We discuss and evaluate a number of optimizations to its running time, which allowed us to compute the greedy spanner on a graph with a million vertices. To our knowledge, this is also the first algorithm for the greedy spanner with a near-quadratic running time guarantee that has actually been implemented.

Furthermore, we will present two more approaches to obtain efficient algorithms for constructing the greedy spanner. The first approach drastically simplifies our first algorithm. It uses much less space and runs in near-quadratic time under reasonable assumptions on the input. The second approach is input-sensitive and aims at subquadratic running time.

3.4 Real-Time On-line Analytical Processing (OLAP) On Cloud Architectures

Frank Dehne (Carleton University – Ottawa, CA)

Joint work of Dehne, Frank; Q.Kong; A.Rau-Chaplin; H.Zaboli; R.Zhou,
Main reference F. Dehne, Q. Kong, A. Rau-Chaplin, H. Zaboli, R. Zhou, "A Distributed Tree Data Structure For Real-Time OLAP On Cloud Architectures", to appear in Proc. of IEEE Int'l Conf. on Big Data (BigData'13), IEEE CS, 2013.
URL http://www.scs.carleton.ca/~dehne/publications/2-94.pdf

In contrast to queries for on-line transaction processing (OLTP) systems that typically access only a small portion of a database, on-line analytical processing (OLAP) queries may need to aggregate large portions of a database which often leads to performance issues. We introduce CR-OLAP, a Cloud based Real-time OLAP system based on a new distributed index structure for OLAP, the distributed PDCR tree, that utilizes a cloud infrastructure consisting of $(m+1)$ multi-core processors. With increasing database size, CR-OLAP dynamically increases m to maintain performance. Our distributed PDCR tree data structure supports multiple dimension hierarchies and efficient query processing on the elaborate dimension hierarchies which are so central to OLAP systems. It is particularly efficient for complex OLAP queries that need to aggregate large portions of the data warehouse, such as "report the total sales in all stores located in California and New York during the months February-May of all years". We evaluated CR-OLAP on the Amazon EC2 cloud, using the TPC-DS benchmark data set. The tests demonstrate that CR-OLAP scales well with increasing number of processors, even for complex queries. For example, on an Amazon EC2 cloud instance with eight processors, for a TPC-DS OLAP query stream on a data warehouse with 80 million tuples where every OLAP query aggregates more than 50of the database, CR-OLAP achieved a query latency of 0.3 seconds which can be considered a real time response

3.5 Customizable Route Planning in Road Networks

Daniel Delling (Microsoft Research – Mountain View, US)

In this talk, I will present the latest version of our customizable route planning engine. It can compute exact shortest paths in continental road networks (with tens of millions of road segments) and is able to incorporate realistic turn costs, custom height and weight restrictions, personal preferences, and traffic in real time. Moreover, it can efficiently answer extended queries such as computing distance tables or best points-of-interest, which are important building blocks for advanced map applications.

3.6 Multi-Modal Route Planning – An Overview

Julian Dibbelt (KIT – Karlsruhe Institute of Technology, DE)

Joint work of Delling, Daniel; Dibbelt, Julian; Pajor, Thomas; Wagner, Dorothea; Werneck, Renato F.
Main reference D. Delling, J. Dibbelt, T. Pajor, D. Wagner, R. F. Werneck, "Computing Multimodal Journeys in Practice", in Proc. of the 12th Int'l Symp. on Experimental Algorithms (SEA'13), LNCS, Vol. 7933, pp. 260–271, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-38527-8_24

This talk gives an overview of recent approaches to location-to-location route planning for multimodal networks consisting of schedule-based public transit, (unrestricted) walking, driving, and cycling. We discuss challenges arising in the multimodal scenario that go beyond the challenges of each (unimodal) subproblem. We review recent algorithmic approaches to the problem such as preprocessing techniques that enable fast queries for interactive applications and conclude by discussing future work.

3.7 The Hub Labeling Algorithm

Andrew V. Goldberg (Microsoft Research – Mountain View, US)

Given a weighted graph, a distance oracle takes as an input a pair of vertices and returns the distance between them. The labeling approach to distance oracle design is to precompute a label for every vertex so that the distances can be computed from the corresponding labels, without looking at the graph. We survey results on hub labeling (HL), a labeling algorithm that received a lot of attention recently.

HL query time and memory requirements depend on the label size. While some graphs admit small labels, one can prove that there are graphs for which the labels must be large. Computing optimal hub labels is hard, but in polynomial time one can approximate them up to a factor of O(log(n)). This can be done for the total label size (i.e., memory required to store the labels), the maximum label size (which determines the worst-case query time), and in general for an Lp norm of the vector induced by the vertex label sizes. One can also simultaneously approximate Lp and Lq norms.

Hierarchical labels are a special class of HL. For networks with small highway dimension, one can compute provably small hierarchical labels in polynomial time. On the other hand, one can prove that for some graphs hierarchical labels are significantly larger than the general ones. A heuristic for computing hierarchical labels leads to the fastest implementation of distance oracles for road networks. One can use label compression to trade off time for space, making the algorithm practical for a wider range of applications. We give experimental results showing that the heuristic hierarchical labels work well on road networks as well as some other graph classes, but not on all graphs. We also discuss efficient implementations of the provably good approximation algorithms and give experimental results.

Finally, we show that the labels can be stored in a database and HL queries can be implemented in SQL, making the algorithm accessible to SQL programmers.

3.8 Machine Learning & Optimisation: Promise and Power of Data-driven, Automated Algorithm Design

Holger H. Hoos (University of British Columbia – Vancouver, CA)

Computational tools are transforming the way we live, work and interact; they also hold the key for meeting many of the challenges that we face as individuals, communities and societies. Machine learning and optimisation techniques play a particularly important role in this context, and cleverly combined, they can revolutionise the way we solve challenging computational problems – as I will demonstrate in this talk, using examples from mixed integer programming, planning and propositional satisfiability, as well as from prominent supervised machine learning tasks. The fundamental techniques I will cover include automated algorithm selection, configuration and hyperparameter optimisation, as well as performance prediction and Bayesian optimisation. I will also motivate and explain the Programming by Optimisation paradigm for algorithm design and engineering.

3.9 A Bootstrap Approach to Analysing the Scaling of Empirical Run-time Data with Problem Size

Holger H. Hoos (University of British Columbia – Vancouver, CA)

Joint work of Hoos, Holger H.
Main reference H. H. Hoos, "A bootstrap approach to analysing the scaling of empirical run-time data with problem size", Technical Report TR-2009-16, University of British Columbia, June 2009.
URL http://www.cs.ubc.ca/~hoos/Publ/Hoos09.pdf

In this presentation, I describe a new approach for analysing the scaling of empirical run-time data of an algorithm when applied to sets of inputs of growing size. My method is based on the use of standard numerical techniques for fitting models, which are then challenged by extrapolation to larger problem sizes and statistically validated using bootstrap confidence intervals. It permits not only the automatic construction of predictive models of the given algorithm's run-time, but also the comparative evaluation of multiple hypothesis on the scaling in the form of different parametric functions. I will illustrate the method using run-time data for Concorde, a state-of-the-art complete algorithm for the travelling salesperson problem (TSP), applied to a class of well-known Euclidean TSP instances. However, the method is generally applicable to most situations where functional (or statistical) models of the dependence of a performance measure of an algorithm on a single, numerical feature of sets of inputs have to be fitted and assessed.

3.10 Fixed-Parameter and Integer Programming Approaches for Clustering Problems

Falk Hüffner (TU Berlin, DE)

We examine two NP-hard clustering problems: Colorful Components, where the goal is to delete a minimum number of edges in a vertex-colored graph such that each remaining connected component contains each color at most once; and Highly Connected Deletion, where the goal is to delete a minimum number of edges in a graph such that each remaining connected component is highly connected. Here, a graph with n vertices is called highly connected if each vertex has degree larger than $n/2$. Both problems have many applications; for example, Colorful Components can be used to find inconsistencies in the mapping of entities between databases, and Highly Connected Deletion can be used to find protein complexes. Using a combination of data reduction and Integer Linear Programming, we are able to solve sizable real-world instances, for example the Wikipedia interlanguage link network with 47 million edges.

3.11 Engineering Algorithms for Color Barcodes

Giuseppe F. Italiano (University of Rome "Tor Vergata", IT)

Joint work of Firmani, Donatella; Italiano, Giuseppe F.; Querini, Marco;

2D color barcodes have been introduced to obtain larger storage capabilities than traditional black and white barcodes. Unfortunately, the data density of color barcodes is substantially limited by the redundancy needed for correcting errors, which are due not only to geometric but also to chromatic distortions introduced by the printing and scanning process. The higher the expected error rate, the more redundancy is needed for avoiding failures in barcode reading, and thus, the lower the actual data density. Our work addresses this trade-off between reliability and data density in 2D color barcodes and aims at identifying the most effective algorithms, in terms of byte error rate and computational overhead, for decoding 2D color barcodes. In particular, we perform a thorough experimental study to identify the most suitable color classifiers for converting analog barcode cells to digital bit streams.

3.12 The TSP for random points in the unit square: an experimentaland statistical study

David S. Johnson (Madison, US)

TSP instances generated by placing cities uniformly at random within the unit square provide reasonable surrogates for the instances arising in many real-world TSP applications. They are also an interesting object of study on their own. The optimal tour length is known to grow as

$c\sqrt{n}$ for some constant c, but the exact value of c and the details of the convergence rate have not been determined. In this talk I report on an ongoing study of this and related questions with David Applegate and Bill Cook, based on experiments with millions of instances using the Concorde software package. Interesting statistical questions arise.

3.13 Graph Clustering with the Louvain Method

Andrea Kappes (KIT – Karlsruhe Institute of Technology, DE)

The Louvain algorithm is a very fast and effective metaheuristic developed for modularity based graph clustering using local vertex moves in a multilevel scheme. I will talk about how the Louvain method can be applied to other objective functions in the presence of constraints on the density of the clustering [2]. Furthermore, it can be effectively transferred to a dynamic setting [1], and to a scenario where multiple clusterings with a varying number of clusters are requested. Finally, it can be modified to take vertex attributes into account. This talk is based on joint work with Philipp Glaser, Robert Görke, Tanja Hartmann, Patricia Iglesias, Uwe Korn, Emmanuel Müller, Christian Staudt, and Dorothea Wagner.

References

1 Robert Görke, Pascal Maillard, Andrea Schumm, Christian Staudt, and Dorothea Wagner. Dynamic Graph Clustering Combining Modularity and Smoothness. *ACM Journal of Experimental Algorithmics*, 18(1):1.5:1.1–1.5:1.29, April 2013.

2 Robert Görke, Andrea Schumm, and Dorothea Wagner. Experiments on Density-Constrained Graph Clustering. In *Proceedings of the 14th Meeting on Algorithm Engineering and Experiments (ALENEX'12)*, pages 1–15. SIAM, 2012.

3.14 Seeking for the best priority queue: Lessons Learnt

Jyrki Katajainen (University of Copenhagen, DK)

Joint work of Bojesen, Jesper; Bruun, Asger; Chen, Jingsen; Edelkamp, Stefan; Elmasry, Amr; Fadel, Ramz; Jakobsen, Kim Vagn; Jensen, Claus; Rasmussen, Jens; Spork, Maz; Teuhola, Jukka; Vitale, Fabio;
Main reference J. Katajainen, "Seeking for the best priority queue: Lessons learnt", CPH STL Report 2013-3, Department of Computer Science, University of Copenhagen, 10 pp., 2013.
URL http://cphstl.dk/Report/Dagstuhl-13391/the-best.pdf

Since 2004, I and my research collaborators have been studying the performance of different priority queues for different sets of operations to be supported. At the theoretical level, we have measured the goodness in terms of the comparison complexity of different operations. As the other optimization criteria, we have considered how to reduce the number of instructions, branch mispredictions, cache misses, and element moves. At the practical level, we have used the actual running time as the key performance indicator. We have done most of our experiments on synthetic operation sequences, but we have also done some application engineering.

In this talk I will briefly survey the theoretical results obtained and I will discuss the methodological issues encountered when performing the practical experiments. As a teaser, consider the following questions:

- What is the best priority queue?
- When to implement a data structure?
- Does a factor of two matter?
- Does the space efficiency matter?
- What is the effect of different hardware phenomena?
- How to measure the practical performance?
- What to do next?

It goes without saying that I do not have complete answers to these questions.

3.15 GraphML-Time: a file format for dynamic graphs

Juergen Lerner (Universität Konstanz, DE)

GraphML (http://graphml.graphdrawing.org/) is an XML-based file format for graphs with arbitrary additional data. During the last ten years GraphML has become an established standard in the graph drawing and network analysis communities. Due to its generality and robustness, it is especially suitable as a format for making graph data available to others. In this talk I propose a GraphML extension for time-varying graph data. The new format is not restricted to represent snapshots of graphs but is also able to efficiently encode dynamic graphs evolving in continuous time. A working draft for the new GraphML version can be found at http://visone.info/wiki/index.php/GraphML-Time

3.16 The Computational Complexity of Virtual Address Translation

Kurt Mehlhorn (MPI für Informatik – Saarbrücken , DE)

Modern computers are not random access machines (RAMs). They have a memory hierarchy, multiple cores, and a virtual memory. We address the computational cost of the address translation in the virtual memory. Starting point for our work on virtual memory is the observation that the analysis of some simple algorithms (random scan of an array, binary search, heapsort) in either the RAM model or the EM model (external memory model) does not correctly predict growth rates of actual running times. We propose the VAT model (virtual address translation) to account for the cost of address translations and analyze the algorithms mentioned above and others in the model. The predictions agree with the measurements. We also analyze the VAT-cost of cache-oblivious algorithms.

3.17 Engineering High-Performance Community Detection Heuristics for Massive Graphs

Henning Meyerhenke (KIT – Karlsruhe Institute of Technology, DE)

Main reference C. L. Staudt, H. Meyerhenke, "Engineering High-Performance Community Detection Heuristics for Massive Graphs," in Proc. of the 42nd Int'l Conf. on Parallel Processing (ICPP'13), pp. 180–189, IEEE CS, 2013; also available as pre-print on arXiv (arXiv:1304.4453v1 [cs.DC]).
URL http://dx.doi.org/10.1109/ICPP.2013.27
URL http://arxiv.org/abs/1304.4453

The amount of graph-structured data has recently experienced an enormous growth in many applications. To transform such data into useful information, high-performance analytics algorithms and software tools are necessary. One common graph analytics kernel is community detection (or graph clustering). Despite extensive research on heuristic solvers for this task, only few parallel codes exist, although parallelism is often necessary to scale to the data volume of real-world applications.

We address the deficit in computing capability by a flexible and extensible clustering algorithm framework with shared-memory parallelism. Within this framework we implement our parallel variations of known sequential algorithms and combine them by an ensemble approach. In extensive experiments driven by the algorithm engineering paradigm, we identify the most successful parameters and combinations of these algorithms. The processing rate of our fastest algorithm exceeds 10M edges/second for many large graphs, making it suitable for massive data streams. Moreover, the strongest algorithm we present includes a parallelization of the well-known Louvain method and yields a very good tradeoff between quality and speed.

3.18 I/O-efficient Hierarchical Diameter Approximation

Ulrich Carsten Meyer (Goethe-Universität Frankfurt am Main, DE)

Joint work of Ajwani, Deepak; Meyer, Ulrich; Veith, David;
Main reference D. Ajwani, U. Meyer, D. Veith, "I/O-efficient Hierarchical Diameter Approximation," in Proc. of the 20th Annual Europ. Symp. on Algorithms (ESA'12), LNCS, Vol. 7501, pp. 72–83, Springer, 2012.
URL http://dx.doi.org/10.1007/978-3-642-33090-2_8

Computing diameters of huge graphs is a key challenge in complex network analysis. As long as the graphs fit into main memory, diameters can be efficiently approximated (and frequently even exactly determined) using heuristics that apply a limited number of BFS traversals. If the input graphs have to be kept and processed on external storage, even a single BFS run may cause an unacceptable amount of time-consuming I/O-operations.

In previous work we proposed the first parameterized diameter approximation algorithm with fewer I/Os than that required for exact BFS traversal. In this talk we present hierarchical extensions of this randomized approach and experimentally compare their trade-offs between actually achieved running times and approximation ratios.

We show that the hierarchical approach is frequently capable of producing surprisingly good diameter approximations in shorter time than BFS. We also provide theoretical and practical insights into worst-case input classes.

3.19 The Art of Discrete Structure Manipulation Based on BDDs and ZDDs

Shin-ichi Minato (Hokkaido University, JP)

Main reference S. Minato, "Techniques of BDD/ZDD: Brief History and Recent Activity," IEICE Transactions on Information and Systems, Vol. E96D, No. 7, pp. 1419–1429, July 2013.
URL http://hdl.handle.net/2115/53121

Discrete structure manipulation is a fundamental technique for many problems solved by computers. Recently, the BDD/ZDD has attracted a great deal of attention, because it efficiently manipulates basic data structures such as logic and sets. In order to organize an integrated method of algebraic operations for manipulating various types of discrete structures, we are executing "ERATO Project", a nation-wide research project in Japan from 2009.

Currently, one of the most interesting research topics of our project is "frontier-based method", a very efficient BDD/ZDD-based algorithm for enumerating and indexing all the subsets of a graph to satisfy various kinds of constraints. For example, we applied this technique for analyzing electricity supply networks and succeeded in generating a ZDD of all solutions for a realistic benchmark with hundreds of control switches. The total number of the solutions was as many as 10^{63}, but the size of ZDD was only 1.1 million (780MByte), and the computation time was about 20 min.

Our project aims to develop efficient software tools based on BDDs/ZDDs and more sophisticated data structures with algebraic operations for various kinds of discrete structure manipulation, which are useful for real- life applications. It would be "the Art" between Science and Engineering. In this talk, we show an overview of our project and recent topics on BDD/ZDD-based discrete structure manipulation.

References

1 Shin-ichi Minato. *Techniques of BDD/ZDD : Brief History and Recent Activity*. IEICE Transactions on Information and Systems, Vol. E96D, No. 7 pp. 1419–1429, July 2013.

3.20 Teaching Algorithm Engineering

Matthias Müller-Hannemann (Martin-Luther-Universität Halle-Wittenberg, DE)

The way how we teach Algorithm Engineering (AE) is of importance for several reasons: it shapes how young researchers entering the field will think about AE and how ideas and techniques are disseminated to other fields. This talk is intended to initiate a discussion on various aspects of teaching. To do so, we give a brief overview of courses on AE taught by colleagues and myself. Issues for discussion include best practices, suggestions for a curriculum, accompanying course material, and ideas for course projects.

3.21 Enumeration algorithms: polynomial time by branch and bound versus exponential time by ZDDs

Yoshio Okamoto (University of Electro-Communications, Tokyo, JP)

Joint work of Kiyomi, Masashi; Okamoto, Yoshio; Saitoh, Toshiki;
Main reference M. Kiyomi, Y. Okamoto, T. Saitoh, "Efficient Enumeration of the Directed Binary Perfect Phylogenies from Incomplete Data," in Proc. of the 11th Int'l Symp. on Experimental Algorithms (SEA'12), LNCS, Vol. 7276, pp. 248–259, Springer, 2012; also available as pre-print on arXiv (arXiv:1203.3284v1 [cs.DS]).
URL http://dx.doi.org/10.1007/978-3-642-30850-5_22
URL http://arxiv.org/abs/1203.3284

When we design an algorithm to enumerate (or list) all the combinatorial objects with a prescribed property, a branch-and-bound technique is quite useful, and in many cases it runs in worst-case polynomial time per output. This is a theoretically guaranteed algorithm, but it is not satisfactory for the purpose of recently growing interest in, for example, data mining and bioinformatics. We compare an approach by branch and bound, and an enumeration algorithm by zero-suppressed binary decision diagrams (ZDDs) proposed by Minato. As a case study, we consider the enumeration of directed binary perfect phylogenies when the input is incomplete. Our experiment shows that the ZDD approach outperforms the branch-and-bound algorithm. The talk is partially based on joint work with Masashi Kiyomi and Toshiki Saitoh [1].

References

1 M. Kiyomi, Y. Okamoto, and T. Saitoh, Efficient Enumeration of the Directed Binary Perfect Phylogenies from Incomplete Data. Proc. SEA 2012 (2012), pp. 248–259. DOI: 10.1007/978-3-642-30850-5_22
http://arxiv.org/abs/1203.3284arXiv:1203.3284 [cs.DS]

3.22 Concurrent data structures in streaming of massive data and overlays in the smart grid context

Marina Papatriantafilou (Chalmers UT – Göteborg, SE)

Joint work of Cederna, Daniel; Georgiadis, Georgios; Gulisano, Vincenzo; Nikolakopoulos, Yiannis; Papatriantafilou, Marina; Tsigas, Philippas
Main reference (collection of papers, see references in the abstract)

In many data gathering applications, information arrives in the form of continuous streams rather than finite data sets. Efficient one-pass and in-memory algorithms are required to cope with a high input load. Stream processing engines provide continuous queries to process data in a real-time fashion and have evolved rapidly from centralized to distributed, parallel and elastic solutions.

The presented work focuses on multiway aggregation, where large data volumes are received from multiple input streams. Multiway aggregation is crucial in contexts such as sensor networks, social media or clickstream analysis applications. We provide three enhanced aggregate operators that rely on new concurrent structures, with domain-related interface and their lock-free implementations, supporting both order-sensitive and order-insensitive aggregation functions. The design of these new data structures balances and parallelizes the work of the aggregate operator's processing steps. We provide an extensive study of the properties of the proposed aggregate operators and the new data structures. We also give empirical evidence of their superiority, running a variety of aggregation queries on two large datasets, one with data extracted from SoundCloud, a music social network, and one with data from a Smart Grid metering network.

In the context of the Smart Grid, network overlays can play a key role, for enabling the management of resources in an adaptive fashion. In this presentation we discussed our work on distributed matching with preferences for dynamic overlays with guarantees, as well as scheduling for demand-response problems in buldings and neighbourhoods in the context of the Smart Grid.

Joint work with Daniel Cederman, Giorgos Georgiadis, Vincenzo Gulisano, Yiannis Nikolakopoulos and Philippas Tsigas.

References

1 Daniel Cederman, Vincenzo Gulisano, Yiannis Nikolakopoulos, Marina Papatriantafilou and Philippas Tsigas. Concurrent Data Structures for Efficient Streaming Aggregation. Technical report, Chalmers University of Technology, 2013.
2 Georgios Georgiadis and Marina Papatriantafilou. A greedy algorithm for the unforecasted energy dispatch problem with storage in smart grids. In Proc. of the 4th Int'l Conf. on Future energy systems (e-Energy'13), ACM, pp. 273–274, 2013. DOI: 10.1145/2487166.2487203
3 Georgiadis, Giorgos, and Marina Papatriantafilou. Adaptive distributed b-matching in overlays with preferences. Experimental Algorithms. Springer, 2012. pp. 208–223.

3.23 Parallel Algorithm Reconsidered

Peter Sanders (KIT – Karlsruhe Institute of Technology, DE)

In this talk we want to motivate researchers in algorithmics in general and in algorithm engineering in particular to work more on parallel algorithms. We begin with a short history of parallel algorithmics – pointing out that a lot of work has been done in the 1980s whereas the area lies almost dormant since the mid 1990s. Since parallelism has moved from a niche application to a mandatory component of any high performance solution this should change. Also since many new developments in algorithmics have happened since then, there are many fundamental results just around the corner. This is particularly true for algorithm engineering. The second part of my talk gives several examples from my group: external sorting [6], string sorting [1], shortest paths [3, 5], multi-objective shortest paths [7], graph matching [2], graph partitioning [4, 9]. The area of communication efficient algorithms is proposed as an entire subfield of parallel algorithm research [8]. Our results there include distributed data structures for approximate membership, duplicate detection, joins, and low-dimensional linear programming. Another interesting area that deserves more attentions are work-efficient algorithms with polylogarithmic critical path length. A surprising new result here is on matrix inversion [10].

References

1 T. Bingmann and P. Sanders. Parallel string sample sort. In *21st European Symposium on Algorithme (ESA)*, volume 8125 of *LNCS*, pages 169–180. Springer, 2013.

2 M. Birn, V. Osipov, P. Sanders, C. Schulz, and N. Sitchinava. Efficient parallel and external matching. In *Europar*, LNCS. Springer, 2013.
3 A. Crauser, K. Mehlhorn, U. Meyer, and P. Sanders. Parallelizing Dijkstra's shortest path algorithm. Technical report, MPI-Informatik, 1998. in preparation.
4 M. Holtgrewe, P. Sanders, and C. Schulz. Engineering a scalable high quality graph partitioner. In *24th IEEE International Symposium on Parallel and Distributed Processing*, 2010.
5 U. Meyer and P. Sanders. Δ-stepping: A parallelizable shortest path algorithm. *Journal of Algorithms*, 49(1):114–152, 2003.
6 M. Rahn, P. Sanders, and J. Singler. Scalable distributed-memory external sorting. In *26th IEEE International Conference on Data Engineering*, pages 685–688, 2010.
7 P. Sanders and L. Mandow. Parallel label-setting multi-objective shortest path search. In *27th IEEE International Parallel & Distributed Processing Symposium*, pages 215–224, 2013.
8 P. Sanders, S. Schlag, and I. Müller. Communication efficient algorithms for fundamental big data problems. In *IEEE Int. Conf. on Big Data*, 2013.
9 P. Sanders and C. Schulz. Distributed evolutionary graph partitioning. In *ALENEX 2012*, 2012.
10 P. Sanders, J. Speck, and R. Steffen. Work-efficient matrix inversion in polylogarithmic time. In *25th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 214–221. ACM, 2013.

3.24 Facility Location Problems in Street Networks

Sabine Storandt (Universität Freiburg, DE)

Joint work of Storandt, Sabine; Funke, Stefan;
Main reference S. Storandt, S. Funke, "Enabling E-Mobility: Facility Location for Battery Loading Stations," in Proc. of the 27th AAAI Conf. on Artificial Intelligence (AAAI'13), AAAI Press, 2013.
URL http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6239
URL http://ad-publications.informatik.uni-freiburg.de/AAAI_evcover_SF_2013.pdf

The main disadvantage of current Electric Vehicles is the short cruising range due to the limited battery supply. Hence for driving longer routes a dense network of loading stations needs to be established. We investigate the problem of placing as few loading stations as possible to achieve mobility goals like global reachability, connectivity and feasibility of shortest paths. Unfortunately, it turns out that the respective optimization problems are all hard to approximate. Moreover naive algorithms require too much space and runtime for practical use. We devise algorithms which compute solutions on the basis of heuristics with incomplete knowledge and a new shortest path enumeration and representation scheme. While these algorithms do not provide any a priori approximation guarantee, we still can prove their results to be close-to-optimal for our real-world inputs with the help of instance-based lower bounds.

3.25 Consistent Updates in Software Defined Networks

Roger Wattenhofer (ETH Zürich, CH)

Computer networks are boring: most aspects are well-understood, and protocols have been standardized a long time ago. However, some people are not happy with the state of the art in networking. Why for instance do networks often waste more than half of the available bandwidth? Introducing software defined networking (SDN), an operator gets more control over its network. With more control, one can possibly achieve results that are not possible with standard protocols. In my talk, I present the SWAN project, an SDN-based effort to raise the bandwidth usage in a corporate wide area network to a mesmerizing 99%. I will focus on some algorithmic questions of the SWAN project, in particular multi-commodity flow and asynchronous updates. My presentation is based on joint work with colleagues at Microsoft Research, published at SIGCOMM 2013 and HotNets 2013.

3.26 An Exact Combinatorial Algorithm for Graph Bisection

Renato Werneck (Microsoft Research – Mountain View, US)

Joint work of Delling, Daniel; Fleischman, Daniel; Goldberg, Andrew; Razenshteyn, Ilya; Werneck, Renato F.;
Main reference D. Delling, D. Fleischman, A. Goldberg, I. Razenshteyn, R. F. Werneck, "An Exact Combinatorial Algorithm for Graph Bisection," submitted for publication.
URL http://research.microsoft.com/apps/pubs/default.aspx?id=200845

We present a novel exact algorithm for the minimum graph bisection problem, whose goal is to partition a graph into two equally-sized cells while minimizing the number of edges between them. Our algorithm is based on the branch-and-bound framework and, unlike most previous approaches, it is fully combinatorial. Besides introducing stronger lower bounds, we propose a new decomposition technique that contracts entire regions of the graph without losing optimality guarantees. In practice, our algorithm works particularly well on instances with relatively small minimum bisections, solving large real-world graphs (with tens of thousands to more than a million vertices) to optimality.

3.27 Improved Alternative Route Planning

Christos Zaroliagis (CTI & University of Patras, GR)

Joint work of Paraskevopoulos, Andreas; Zaroliagis, Christos;
Main reference A. Paraskevopoulos, C. Zaroliagis, "Improved Alternative Route Planning," in Proc. of the 13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'13), OASICS, Vol. 33, pp. 108–122, 2013.
URL http://dx.doi.org/10.4230/OASIcs.ATMOS.2013.108

We present improved methods for computing a set of alternative source-to-destination routes in road networks in the form of an alternative graph. The resulting alternative graphs are characterized by minimum path overlap, small stretch factor, as well as low size and complexity. Our approach improves upon a previous one by introducing a new pruning stage preceding any other heuristic method and by introducing a new filtering and fine-tuning of two existing methods. Our accompanying experimental study shows that the entire alternative graph can be computed pretty fast even in continental size networks.

3.28 A new algorithm for solving the All-Pairs-Shortest-Paths problem on positive, real-weighted graphs

Katharina A. Zweig (TU Kaiserslautern, DE)

Joint work of Zweig, Katharina Anna;
Main reference (unpublished work)

While better algorithms for the all-pairs-shortest-paths-problem (APSP) exist in many cases, the simplicity of Dijkstra's algorithm still makes it the most used one with a runtime of $O(nm + n^2 logn)$. Real-world networks such as street or sensor networks are known to be sparse such that the second term in the runtime of Dijkstra's algorithm is dominating. In this talk I will present a new algorithm for the APSP and show that it can be expected to run in $O(nm)$ for weighted sensor networks. If it is implemented as a distributed algorithm where every sensor computes the distance to all other nodes, the message complexity is also in $O(nm)$ and the algorithm can be implemented in an asynchronous fashion. Finally, I will give empirical evidence that the APSP can also be solved in $O(nm)$ for many street networks.

4 Panel Discussions

4.1 Benchmarks and reproducibility of experiments

Andrew V. Goldberg (Microsoft Research – Mountain View, US)

Common benchmarks are important for the field Algorithm Engineering. They allow researchers to compare experimental results. The discussion included the topics of creating the benchmarks, e.g., via the Challenges. An important issue that has been raised is benchmark evolution: As algorithms become better, benchmarks may become outdated and need to a revision. A related issue is overfitting, when over time the algorithms become tuned to the benchmark instances at the expense of other problems.

Since the Science of Algorithmics applies the scientific method to algorithm design, it is has to deal with the reproducibility and verification issues common to natural sciences. Making code available partially addresses these issues. However, this does not completely settle them: For example, the code may not faithfully implement its description in the corresponding paper. System issues also play a role, especially when large distributed systems and supercomputers with limited access are involved. Another approach to reproducibility is independent reimplementation. It is a great way to reproduce the results, and should be encouraged when happens. However, for sophisticated code, reimplementation is difficult, and often there is little incentive for reimplementation.

4.2 Promoting and Advancing the Field

David S. Johnson (Madison, US)

This panel discussion was lead by David Johnson, and considered two main topics. The first was potential topics for future DIMACS Implementation Challenges. The current Challenge on Steiner Problems, and the previous one on Clustering and Partitioning, were both based on ideas suggested at the previous Dagstuhl workshop on Algorithm Engineering. This time a number of ideas were floated, including a combination of facility location and the k-median problem, but perhaps the most popular one was to reprise the first Challenge from 1990–91, which covered Network Flows and Matching. There have been many new theoretical developments since then, with several major ones in the last two years that seem quite promising but haven't yet been adequately tested. At the first Challenge, a paradigm shift was confirmed, with the then-new "preflow-push" approach of Goldberg and Tarjan dominating the previous state of the art. Could a new paradigm shift be at hand?

The second main topic was that of prizes. At the last workshop, it was suggested that we try to get sponsors for a "Best Algorithm Engineering/Experimentation paper of the Year" award. One suggestion was to get SIGACT and EATCS to jointly sponsor such an award, as they currently do for theoretical papers with their Gödel Prize. Initial responses from the leaders of SIGACT were not encouraging, but new leaders are now in charge, and it is hoped that with a more detailed proposal we might now be successful.

Participants

- Hannah Bast
Universität Freiburg, DE
- Jon Bentley
Avaya – Basking Ridge, US
- Robert E. Bixby
Rice University – Houston, US
- Gerth Stølting Brodal
Aarhus University, DK
- Kevin Buchin
TU Eindhoven, NL
- Markus Chimani
Universität Osnabrück, DE
- Frank Dehne
Carleton Univ. – Ottawa, CA
- Daniel Delling
Microsoft Research – Mountain View, US
- Julian Dibbelt
KIT – Karlsruhe Institute of Technology, DE
- Rudolf Fleischer
German Univ. of Technology – Oman, OM
- Andrew V. Goldberg
Microsoft Research – Mountain View, US
- Holger H. Hoos
University of British Columbia – Vancouver, CA
- Falk Hüffner
TU Berlin, DE
- Giuseppe F. Italiano
University of Rome "Tor Vergata", IT
- David S. Johnson
Madison, US
- Andrea Kappes
KIT – Karlsruhe Institute of Technology, DE
- Jyrki Katajainen
University of Copenhagen, DK
- Jürgen Lerner
Universität Konstanz, DE
- Kurt Mehlhorn
MPI für Informatik – Saarbrücken, DE
- Ulrich Carsten Meyer
Goethe-Universität Frankfurt am Main, DE
- Friedhelm Meyer auf der Heide
Universität Paderborn, DE
- Henning Meyerhenke
KIT – Karlsruhe Institute of Technology, DE
- Shin-ichi Minato
Hokkaido University, JP
- Rolf H. Möhring
TU Berlin, DE
- Matthias Müller-Hannemann
Martin-Luther-Universität Halle-Wittenberg, DE
- Petra Mutzel
TU Dortmund, DE
- Patrick K. Nicholson
MPI für Informatik – Saarbrücken, DE
- Yoshio Okamoto
University of Electro-Communications – Tokyo, JP
- Marina Papatriantafilou
Chalmers UT – Göteborg, SE
- Alejandro Salinger
Universität des Saarlandes, DE
- Peter Sanders
KIT – Karlsruhe Institute of Technology, DE
- Federico Santaroni
University of Rome "Tor Vergata", IT
- Sabine Storandt
Universität Freiburg, DE
- Dorothea Wagner
KIT – Karlsruhe Institute of Technology, DE
- Roger Wattenhofer
ETH Zürich, CH
- Renato Werneck
Microsoft Research – Mountain View, US
- Christos Zaroliagis
CTI & University of Patras, GR
- Liang Zhao
KIT – Karlsruhe Institute of Technology, DE
- Katharina A. Zweig
TU Kaiserslautern, DE

Report from Dagstuhl Seminar 13392

Inter-Vehicular Communication – Quo Vadis

Edited by

Onur Altintas[1], Falko Dressler[2], Hannes Hartenstein[3], and Ozan K. Tonguz[4]

1 **TOYOTA InfoTechnology Center – Tokyo, JP**
2 **Universität Innsbruck, AT,** dressler@ccs-labs.org
3 **KIT – Karlsruhe Institute of Technology, DE,** hannes.hartenstein@kit.edu
4 **Carnegie Mellon University – Pittsburgh, US,** tonguz@ece.cmu.edu

Abstract

"Inter-Vehicular Communication – Quo Vadis?". With this question in mind, leading experts in the field of vehicular networking met in Dagstuhl to discuss the current state of the art and, most importantly, the open challenges in R&D from both an scientific and an industry point of view. After more than a decade of research on vehicular networks, the experts very seriously asked the question whether all of the initial research issues had been solved so far. It turned out that the perspective changed in the last few years, mainly thanks to the ongoing field operational tests in Europe and the U.S. The results point to new research directions and new challenges that need to be solved for a second generation of vehicular networking applications and protocols. In four working groups, the experts studied these new challenges and derived recommendations that are also very helpful for the respective funding organizations.

Seminar 22.–25. September, 2013 – www.dagstuhl.de/13392

1998 ACM Subject Classification C.2 Computer-Communication Networks, C.2.1 Network Architecture and Design – Wireless communication, C.2.2 Network Protocols, C.4 Performance of Systems

Keywords and phrases Vehiular Networking, Inter-Vehicle Communication, Intelligent Transportation Systems

Digital Object Identifier 10.4230/DagRep.3.9.190

1 Executive Summary

Onur Altintas
Falko Dressler
Hannes Hartenstein
Ozan K. Tonguz

Motivation

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

There is a long list of desirable applications that can be grouped into four categories:

- eSafety applications that try to make driving safer, e.g., road hazard warning;
- traffic efficiency applications aiming at more efficient and thus greener traffic, e.g., detection of traffic jams;
- manufacturer oriented applications, e.g., automatic software updates; and
- comfort and entertainment applications, e.g., automatic map updates or video streaming.

While there are some similarities with fields like mobile ad-hoc networks or wireless sensor networks, the specific characteristics of vehicular networks require different communication paradigms, different approaches to security and privacy, or different wireless communication systems. For example, the nodes usually do not have severe power and form factor constraints, and they might be always on. On the other hand, due to high relative speeds, wireless connections may not be stable for a longer time period and the network density is expected to vary from sparse to very dense networks. Another challenging issue is the efficient use of available infrastructure, such as road side units or even cellular networks. Furthermore, IVC has strong links to other research domains, e.g., geo-informatics as it requires very precise localization and precise maps or highly scalable simulations that are a requirement for analyzing traffic systems with hundreds or thousands of vehicles.

In the past, many specific solutions for IVC have been identified and now, industry and other stake-holders are already calling for standardization. Still, we believe that many important research questions have only been partially answered and the approaches discussed in the standardization bodies are based only on a minimum consensus of simplest solutions. Security and privacy, scalability, use of advanced communication patterns like aggregation, transmit power control, and optimal medium access are just a few of such issues.

In 2010, a first Dagstuhl Seminar (10402) was organized on the topic of inter-vehicular communication [1, 2]. The motivation was to bring together experts in this field to investigate the state of the art and to highlight where sufficient solutions already existed. The main outcome of this very inspiring seminar was that there are indeed areas within this research where scientific findings are being consolidated and adapted by industry. This was the consensus of quite intriguing discussions among participants from both industry and academia. Yet, even more aspects have been identified where substantial research is still needed. These challenges have been summarized in the Dagstuhl report [1] and in an IEEE Communications Magazine article [2].

Objectives

It was the goal of this new seminar to again bring together leading researchers both from academia and industry to discuss if and where the previously identified challenges have been adequately addressed, and to highlight where sufficient solutions exist today, where better alternatives need to be found, and also to give directions where to look for such alternatives. Furthermore, the goal of this workshop was to go on step beyond and identify where IVC can contribute to the basic foundations of computer science or where previously unconsidered foundations can contribute to IVC.

The 2010 Dagstuhl seminar promoted a "top-down" approach to inter-vehicle communications instead of the classical "bottom-up" approach. With the top-down approach, the effects of applications are first analyzed under the assumption that the communication system will be able to support the application. Thus, an "upper bound" can be presented on the benefits of IVC. In our discussions, we summarized all the scientific work that followed this approach after the previous Dagstuhl seminar and contrasted it with new insights based on field operational tests, safety application design and massively distributed operations.

In particular, we shifted the focus from basic networking principles to applicability in real world scenarios. In the last few years, first field operational tests have been conducted in the US (the Michigan field trial) as well as in Europe (SIM-TD in Germany, DRIVE C2X in Europe). Lessons learned from those tests applied to currently used models and concepts will bring new insights into the forthcoming research challenges. Among others, questions to be studied include the following still unanswered research challenges:

- Data analysis of current field operational tests: are they validating or invalidating current models?
- Safety applications: show stopper or driving force? What are the limitations in terms of latency and reliability of available communication principles for enabling critical safety support;
- From highly distributed to massively distributed operation: can vehicular networking based on DSRC/WAVE also support all the pedestrians and bicyclists?

We organized the 2013 seminar again as a discussion forum. Three invited keynote presentations were organized to stimulate discussions among the participants. In order to steer the discussions, we prepared four working groups that helps focusing on selected open research challenges. In addition, we also supported ad-hoc presentations on topics of the working groups. The following working groups have been formed and led to very interesting observations:

- Foundations – In this group, it was discussed, which fundamental insights gained in the vehicular networking research domain can be transfered to other domains of computer science. The other way around has been discussed as well, i.e., which areas of computer science might help fostering work in the vehicular networking and which may help overcoming open challenges.
- Field Operational Tests (FOTs) – This group focused on the results that already have been derived from the ongoing work in various test sites in the U.S. and in Europe. The main questions in the discussion were whether the current experiments are already sufficient to gain insights into larger scale behavior or if additional tests are needed.
- IVC Applications – In this group, the applications' perspective to IVC was discussed. In the last years, many of the developments have been done looking at lower layer networking problems. This resulted in a number of networking solutions that nicely support specific applications but cannot be integrated to a generalized networking architecture.
- Heterogeneous Networks – Possibly one of the most important and timely working groups focused on the integration of different networking technologies. This is strongly needed to develop integrated IVC solutions and also to overcome early deployment problems like the initially low penetration ratio.

Eventually, all these questions lead to the big question whether vehicular networking can now be shown to improve efficiency and safety on our streets. We are now in an era that completely changes the game in car manufacturing and road traffic management. Computer science is becoming the key element in the design of these systems. It is of utmost importance to bring in expertise from classical computer science (computer networking, simulation and modeling, operating system design) as well as from electrical engineering (digital signal processing, communication networks) as well as experts from the automotive industry and from the intelligent transportation community.

References

1 Falko Dressler, Frank Kargl, Jörg Ott, Ozan K. Tonguz, and Lars Wischhof, "Executive Summary – Inter-Vehicular Communication," in *Dagstuhl Seminar 10402 – Inter-Vehicular Communication*. Schloss Dagstuhl, Wadern, Germany: Schloss Dagstuhl, October 2010. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2011/2929/

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

2 Table of Contents

3 Overview of Talks

3.1 Studying safety applications for vehicular communications

Natalya An (KIT – Karlsruhe Institute of Technology, DE)

Joint work of An, Natalya; Hartensten, Hannes; Neudecker, Till

In this presentation we point out three current challenges in studying of safety applications. First, there is a need for common and clear definition of an application as a protocol. Second, application requirements need to be justified from a traffic safety perspective. The requirements analysis has to consider kinematics and the driver as well as minimization of false alarms. Such analysis can also be independent of communication technology and simply focus on information that is required. As the last point, we discuss the verification of safety applications on one example application. If safety application can be verified to be fail-safe, how efficient are they? We also quantify the tradeoff between safety and efficiency.

3.2 Fundamental limitations of the basic IVC system and related research questions

Andreas Festag (TU Dresden, DE)

The talk gave a brief overview of the IVC system that has been standardized in Europe and tested in FOTs, such as simTD, DRIVE C2X, etc. This system is also referred to as "basic system for initial deployment". From this reference system key limitations of the system covering physical transmission, medium access, networking, messaging, and congestion control; these limitations are linked to research challenges for future IVC taking that takes into account the (hopefully) coming deployment.

3.3 Play The Game

Raphaël Frank (University of Luxembourg, LU)

We are moving towards a cooperative world. Most people nowadays have ubiquitous Internet connectivity through their mobile devices. Social Networks have now become a part of our daily lives. They allow us to discuss and interact with virtually everybody on the plant. This platform enables a plethora of new cooperative applications including "social games". The idea here is to increase the efficiency of a system by building a game around it. In the context of vehicles, one could think of various gaming scenarios to increase road safety, decrease consumption or reduce traffic congestion by providing a score and rank to the participants.

3.4 Quod Vides? The Eyes of the Vehicle Computing Cloud

Mario Gerla (University of California – Los Angeles, US)

New vehicle applications have recently emerged in several areas ranging from navigation safety to location aware content distribution, intelligent transport, commerce and games. This diversity of applications sets the Vehicular ad Hoc Network (VANET) apart from conventional military and civilian emergency MANETs and does introduce new design challenges. In this talk we review the recently defined VANET standards, introduce emerging vehicular applications and examine the new services they can provide. A representative service scenario is urban sensing: vehicles monitor the environment, classify the events, e.g., license plates, chemical readings, radiation levels, and then generate metadata of what they observed. The metadata in turn can be uploaded to Internet servers or can be kept on board of vehicles to support future services such as forensic harvesting by Authorities. The notion of VANET Services suggests that the VANET can be viewed as a mobile service providing Cloud. In fact the VANET is an important example of a new type of Cloud, the Mobile Computing Cloud (MCC). The MCC consisting of mobile agents (people, vehicles, robots) that interact and collaborate to sense the environment, process the data, propagate the results and more generally share resources in order to produce mobile services that are not efficiently supported by the Internet Cloud. In this talk we will revisit VANET applications and services in light of this Mobile Cloud model. We will also address the cooperation between Vehicular Clouds and the Internet Cloud in the context of a vehicular traffic management application.

3.5 Do Vehicular Networks Scale? Early Lessons from Field Trials and Simulations for Academic Research

Marco Gruteser (Rutgers University – New Brunswick, US)

After the past decade of vehicular network protocol research and standardization, vehicular networks have now moved into a field trial stage. The Scalability Field Trials, which are planned and conducted by the CAMP VSC3 Consortium in cooperation with the USDOT, are trials that seek to identify a transmission control protocol for scalable V2V safety communications. Such a protocol should preserve the performance of V2V applications in both congested and uncontested communication environments. To this end, 200 DSRC equipped vehicles were driven in dense configurations on testing grounds with key network performance indicators logged. We now use the experimental data to calibrate and validate ns-3-based simulation models, which can then be used to predict performance in even denser configurations. After implementing appropriate capture and propagation models as well as correcting MAC inaccuracies our initial simulation results show good agreement with the field tests and promising results on transmission control algorithm effectiveness.

3.6 IVC – Beyond DSRC & Beyond Vehicles

Jerôme Haerri (EURECOM – Biot, FR)

IVC has been associated to DSRC for dedicated wireless communications between vehicles. In a larger context of smart cities, IVC may be extended beyond DSRC and beyond vehicles. In this talk, we investigate the feasibility of device-to-device LTE communication (LTE-Direct) in a traffic safety context. We extend LTE-Direct for periodic broadcast transmission of beacon messages containing mobility states (aka CAM in EU, BSM in US). We propose to employ a quasi-static OFDMA downlink resource allocation similar to eMBMS, where multiple eNBs reserve LTE downlink resource blocks for dedicated device-to-device communication. Considering an optimal resource allocation between all vehicles covered by the eNBs, our proposal may sustain up to 100 vehicles transmitting beacons at a rate of 10Hz. Although challenges remain to efficiently allocate the reserved OFDMA resources in a fully distributed way, our proposal is a first attempt to show that the LTE-Direct technology may be used for traffic safety applications in a larger context of mobile (pedestrian, bicycles, cars) rather than only vehicular communication.

3.7 Exploring Space – towards high-capacity inter-vehicular communications

Geert Heijenk (University of Twente, NL)

This presentation discusses the question 'Are there still research challenges in inter-vehicular communications'. The premise is that these may come from autonomous, or rather coordinated driving. I will start with a few results from an earlier project, Connect & Drive, where a system for cooperative adaptive cruise control was researched, designed, and prototyped. We project that for coordinated driving, important challenges are in the area of reliable consensus for coordinated manoeuvres, and high-rate beaconing for increased situational awareness of vehicles. I show that current systems do not suffice for these challenges. In order to increase the scalability of inter-vehicular communications, I propose to explore spatial reuse, by using cheap large-scale antenna arrays and beamforming receivers. This way, a vehicle can be equipped with a large number of receivers, each receiving from a specific (dynamically reconfigurable) direction. Given this idea, I point at important research questions, and argue that for a good understanding, the use of good analytical performance models is of paramount importance.

3.8 Are Generic Communication Systems Possible?

Frank Kargl (Universität Ulm, DE)

In this talk we raise the question how IVC communication systems can be made more flexible and future-proof. As we introduce a first generation of IVC communication systems, we also need to be prepared for an evolution that includes introduction of additional new applications, aggregation protocols, or other elements. We propose a lightweight framework were nodes in the network can introduce at deployment time new components that can alter or extend the behavior of the system. For example, an application that requires a new algorithm for aggregation data in the network could provide such a component when it is deployed in the network.

3.9 Do We Need … in IVC?

Renato Lo Cigno (University of Trento – DISI, IT)

In this short talk I briefly point out and stimulate discussion on some of the "golden fleece" that are being pursued in the ICV area for safety applications and that are probably slowing down research and jeopardize adoption. There is a hype that safety application require a networking support similar to a deep space mission, where a single failures means losing a billion dollar project. Safety in traffic, instead, means improving the current situation, dominated by human errors and where, worldwide, an estimated trillion (1012) USD are spent because of car accidents, not counting for casualties maimed people and social costs in general. Thus, I argue that the fundamental scientific questions that should be addressed by this community relate to the minimal amount of information needed for vehicles to react, and not to deliver all information, relate to devising randomized, distributed protocols and algorithms that improve the situation and integrate with sensors on board, rather than finding the one-system-fit-them-all perfect solution.

3.10 V-NDN: Vehicular Named Data Networks

Giovanni Pau (University of California – Los Angeles, US)

In this work we apply the Named Data Networking, a newly proposed Internet architecture, to networking vehicles on the run. Our design, VNDN, illustrates NDN's promising potential to providing a unifying architecture that enables networking among all computing devices independent from whether they are connected through wired infrastructure, ad hoc, or intermittent DTN. We also describe a proof-of-concept V-NDN implementation.

3.11 Platooning and Network Related Challenges: Solutions and First Results

Michele Segata (University of Trento – DISI, IT)

Platooning, the idea of cars autonomously following their leaders to form a road train, has huge potentials to improve traffic flow efficiency, driving experience on freeways, and most importantly road traffic safety. Wireless communication is a fundamental building block for this application – it is needed to manage and to maintain the platoons. However, strict constraints in terms of update frequency and reliability must be met. In this talk, we analyze the performance of information dissemination strategies for platooning based on DSRC/WAVE. In particular, we developed communication strategies exploiting synchronized communication slots as well as transmit power adaptation. We evaluate the performance of the controller under different update frequencies, showing that beacon frequency could be adapted depending on the dynamics of the platoon. Using the platooning simulator we developed, we demonstrate the effectiveness of a combined TDMA plus transmit power control scheme even in dense vehicular scenarios.

3.12 Heterogeneous Vehicular Networking

Christoph Sommer (Universität Innsbruck, AT)

Heterogeneous vehicular networks are not a new concept by far: always best connected transmission of data over one of a set of channels has been shown to be highly beneficial to increase connectivity and combat low equipment rates. We show that heterogeneous vehicular networks can do more than that: by making smart use of the different properties that, e.g., cellular and short range radio channels offer, we can provide new and better services, such as cellular assisted intersection collision avoidance.

3.13 Design of Congestion Control for Vehicle Safety Communications

Tessa Tielert (KIT – Karlsruhe Institute of Technology, DE)

In this talk, we address three questions about congestion control in vehicular safety communications. First, do we need it at all? Second, if we do need it, how should we do it? Typical "turning knobs" are transmit power and message generation rate. In addition, safety communications should provide "fairness" and "awareness". And finally, what is still to do in this field? We start by addressing the spatio-temporal requirements of awareness. We then study the potential of the communication system to minimize packet inter-reception time (IRT) at a certain sender-receiver distance and identify the parameter combinations optimizing this metric. We show that a fixed parameter setting is ineffective, transmit power control and message rate control each leave room for improvement and a joint control strategy seems to provide best results. We discuss different aspects of fairness and introduce the concept of basing fairness not on the share of bandwidth but on the achieved safety benefit. However, a concrete definition of this concept requires a clear understanding (and metrics) of safety applications' requirements, which we see as a major challenge for the future.

3.14 Potential benefits from (DSCR)-C2X – Questions from a traffic engineer

Peter Vortisch (KIT – Karlsruhe Institute of Technology, DE)

After ten years of research and development, intervehicular communication based on DSRC is ready for deployment. What benefit can they produce, from a traffic engineering perspective? Promises were made to improve safety, efficiency, fuel consumption and even travel behavior. Many of the intended applications generate warnings about some dangerous situations. This is a natural domain of DSCR, and given the lower latency compared to cellular communication, a certain benefit can certainly be generated. But for hard safety applications like collision avoidance, cars will still rely on autonomous sensors in the first place. Besides safety applications, traffic efficiency is addressed. One of the prominent application examples is communicating with traffic signals. The cars are informed about green or red times and can adapt their approach to avoid stopping and thus save fuel. The problem here is that there are not many fixed time signals left since traffic control is vehicle actuated in most intersections because of pedestrians or transit priority. And even in the case of fixed time signals, the time frame in which the given information can be useful is pretty short. In general, it is not easy to make a convincing case at the moment for DSRC systems to improve traffic efficiency significantly, partly because many applications are already "taken away" by cellular communication based systems. In interesting field for future research will be the interaction of vehicular communication and the rise of automated driving.

4 Working Groups

4.1 Heterogeneous Vehicular Networks

Claudio Casetti, Falko Dressler, Mario Gerla, Javier Gozalvez, Jerôme Haerri, Giovanni Pau, and Christoph Sommer

4.1.1 Introduction

A future trend of vehicular networks is the move away from focusing on just a single technology and towards designing systems that can make use of multiple different technologies, creating *heterogeneous vehicular networks*. Looking into the literature, however, the underlying assumptions, concepts, and even the goals of such approaches are very fuzzy. In an effort to move this research area forward by clarifying the foundations, identifying commonalities and differences of existing approaches, and outlining future research directions, a working group was formed at Dagstuhl Seminar 13392 to tackle these questions.

The group meeting kicked off by defining the concept of heterogeneous vehicular networks. In the context of networking in general, the term heterogeneous networking is sometimes used as a catch-all definition: for example, there is a clear consensus within 3GPP to define the integrated large-cell/small-cell coverage in LTE-advanced and its related issues as heterogeneous networks. Such definitions do not apply to our case.

In vehicular networking it was agreed that a *Heterogeneous Vehicular Network* is to refer to a system characterized by the integration of different technologies such as IEEE 802.11[p] DSRC together with higher layer protocols such as WAVE [1] or ITS G5 [2], IEEE 802.11[abgn] consumer WiFi [3], and 3G/4G cellular networks.

4.1.2 Motivation

One of the key motivations for considering such heterogeneous vehicular networks is the widespread availability of multiple technologies – both on today's portable devices like smart phones and in modern cars' sat nav systems or multimedia units.

Further, the team was quick to agree that – while cellular networks, such as LTE, will be a big helper during any initial rollout of short range communication technology – cellular networks will, in the medium term, not be able to offer sufficient network capacity without a drastic increase in deployment density and/or price [4, 5]. They might, in the long term, even be unable to offer sufficient capacity.

Heterogeneous vehicular networking is further motivated by the fact that each of the currently available wireless technologies offers unique benefits, but also unique drawbacks. It was argued that the reasons to have WiFi lie in the downloading of added-value content and in the creation of a truly integrated environment, which would not be limited to cars as the only road users: Indeed, WiFi would foster the integration of bicycles and pedestrians into the network. Further, because of its tailored physical layer, dedicated channel(s), and tight locality, DSRC can offer unique benefits in safety and cooperation awareness applications, due to their tight latency requirements. On the other end of the spectrum, cellular technologies are widely available, and designed for delivering large amounts of data over arbitrary distances. On the down side, they could face further hurdles when multicasting or local broadcasting is a strong requirement. Indeed, the lack of specific multicast support even in current 4G networks, coupled with multi-operator terminals, is a critical limitation [6].

The team identified two basic, opposing trends in heterogeneous vehicular networking that can be classified as follows:

(A) pushes for a generalized network stack that abstracts away from lower layers to decouple applications from the employed technology, aiming to provide *data offloading* services, or an *always best connected* experience to upper layers.

(B) follows a *best of both worlds* approach, exposing information and control of lower layers to applications, enabling them to selectively use the best fitting technology for a particular task.

4.1.3 Class A

Having multiple technologies at hand gives vehicles the option to communicate in an always best connected fashion. This allows them to efficiently combat hard to predict local shadowing and fading effects. Further, it allows them to operate even in very sparse networks, unhindered by network fragmentation or similar problems that would plague a purely DSRC based solutions early after market rollout.

Further, using multiple technologies in parallel for sending can make the delivery of 'one in a million' safety messages much more robust. It can further help thwart physical layer attacks or serve to cross-validate potentially fraudulent messages.

The discussion then moved to the use of DSRC for cellular offloading to increase capacity. The consensus was that many literature works already explored cellular offloading [7], but that the main applications seem to involve some variations of the caching-and-forwarding concept. However, in order to be effective, caching must be applied to popular content. It was remarked that there are no reliable studies of how "popular" content must be so as to turn offloading into a viable option.

In a similar vein, it is possible to use one technology to deliver a basic level of service, and another for optional, enhanced levels of service, e.g., the base layer and enhanced layers of scalable video coding [8].

4.1.4 Class B

As an alternative to the more straightforward *always best connected* abstract approach discussed previously, heterogeneous networks could also much more directly instrument multiple technologies, employing each to its full capacity and according to its particular benefits and drawbacks.

We categorized approaches in this class into two sub-classes:

(B1) chooses the underlying technology according to a *control/data* split.
Sending control information via a cellular channel, if available, can ensure that control information reaches the highest number of nodes, independent of network topology, and even kilometers in advance. Sending data via multihop DSRC can serve to ensure that the network load caused by such data exchange remains local only.
One example of such a network is the MobTorrent approach [9], which employs a cellular network for transmitting control data to WiFi access points, allowing them to prefetch and cache data to offer Internet access to vehicles.
A more recent example turns this architecture on its head, utilizing DSRC for service announcements and a cellular network for supporting infotainment data dissemination [10].

(B2) splits data according to a *local/global* decision.
Local collaboration via DSRC if necessary (and, thus, if available) can make best use of the low latency offered by this technology. Medium-scale or global collaboration via cellular networks, transmitting only aggregate information, can supplement local collaboration: it can exploit the universal availability of cellular networks without causing undue load and without suffering from its drawbacks for local communication.
One example of such a network is a clustering approach [11], which employs short range radio for near field information exchange in clusters and cellular networks for interconnecting clusters.

4.1.5 Conclusion

The group meeting adjourned after identifying three promising research directions for heterogeneous vehicular networks:

- combining technologies with long-range and short-range coverage: they have different objectives but a positive fallout is expected from their joint deployment;

- investigation of the feasibility of integrating a high number of different radio technologies into one device; investigation of Software-Defined Radio (SDR) as a potential way forward [12];
- further investigation of offloading, scheduled downloading and relaying is needed, identifying promising use cases;
- continuing development of safety protocols and applications, under the premise that, though safety may not carry much money it is the only option to make DSRC mandatory on newly-manufactured vehicles.

References

1 "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) – Multi-channel Operation," IEEE, Std 1609.4, February 2011.

2 European Telecommunications Standards Institute, "Intelligent Transport Systems (ITS); Communications Architecture," ETSI, EN 302 665 V1.1.1, September 2010.

3 "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE, Std 802.11-2007, 2007.

4 A. Vinel, "3GPP LTE Versus IEEE 802.11p/WAVE: Which Technology is Able to Support Cooperative Vehicular Safety Applications?" *Wireless Communications Letters*, vol. 1, no. 2, pp. 125–128, April 2012.

5 H. Rakouth, P. Alexander, A. Brown Jr., W. Kosiak, M. Fukushima, L. Ghosh, C. Hedges, H. Kong, S. Kopetzki, R. Siripurapu, and J. Shen, "V2X Communication Technology: Field Experience and Comparative Analysis," in *FISITA World Automotive Congress*, vol. LNEE 200. Beijing, China: Springer, November 2012, pp. 113–129.

6 C. Sommer, A. Schmidt, Y. Chen, R. German, W. Koch, and F. Dressler, "On the Feasibility of UMTS-based Traffic Information Systems," *Elsevier Ad Hoc Networks, Special Issue on Vehicular Networks*, vol. 8, no. 5, pp. 506–517, July 2010.

7 F. Malandrino, C. E. Casetti, C.-F. Chiasserini, C. Sommer, and F. Dressler, "Content Downloading in Vehicular Networks: Bringing Parked Cars Into the Picture," in *23rd IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2012).* Sydney, Australia: IEEE, September 2012, pp. 1534–1539.

8 E. Yaacoub, F. Filali, and A. Abu-Dayya, "SVC video streaming over cooperative LTE/802.11p vehicle-to-infrastructure communications," in *2013 World Congress on Computer and Information Technology (WCCIT 2013).* Sousse, Tunisia: IEEE, June 2013.

9 B. B. Chen and M. C. Chan, "MobTorrent: A Framework for Mobile Internet Access from Vehicles," in *28th IEEE Conference on Computer Communications (INFOCOM 2009).* Rio de Janeiro, Brazil: IEEE, April 2009.

10 A. Baiocchi and F. Cuomo, "Infotainment services based on push-mode dissemination in an integrated VANET and 3G architecture," *Communications and Networks, Journal of*, vol. 15, no. 2, pp. 179–190, 2013.

11 L.-C. Tung, J. Mena, M. Gerla, and C. Sommer, "A Cluster Based Architecture for Intersection Collision Avoidance Using Heterogeneous Networks," in *12th IFIP/IEEE Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2013).* Ajaccio, Corsica, France: IEEE, June 2013.

12 N. Haziza, M. Kassab, R. Knopp, J. Härri, F. Kaltenberger, P. Agostini, M. Berbineau, C. Gransart, J. Besnier, J. Ehrlich, and H. Aniss, "Multi-technology Vehicular Cooperative System Based on Software Defined Radio (SDR)," in *5th International Workshop on Communication Technologies for Vehicles (Nets4Cars-2013)*, vol. LNCS 7865. Lille, France: Springer, May 2013, pp. 84–95.

4.2 Fundamentals: IVC and Computer Science

Javier Gozalvez, Jerôme Haerri, Hannes Hartenstein, Geert Heijenk, Frank Kargl, Jonathan Petit, Bjöen Scheuermann, and Tessa Tieler

The working group on "Fundamentals: IVC and Computer Science" discussed the lasting value of achieved research results as well as potential future directions in the field of inter-vehicular communication. Two major themes 'with variations' were the dependence on a specific technology (particularly the focus on IEEE 802.11p in the last decade) and the struggling with bringing self-organizing networks to deployment/market.

The team started with a retrospective view and identified the following topics as major contributions in the last decade: analysis and design of single-hop broadcast communication and geonetworking, scalability issues (for both, small and large penetration rates) as well as corresponding security and privacy approaches. In addition, all the work also led to a strong requirements elicitation for the domains of safety and efficiency applications bringing together traffic experts, automotive engineers and the IVC community. The working group considered various contributions to have a lasting value, particularly analytical models for information dissemination, approaches to control or to avoid congestion of the radio channel, building control applications on top of the unreliable wireless communication as well as a bunch of security approaches like broadcast authentication and misbehavior detection. In addition, the working group tried to check whether results from the previous Dagstuhl seminar on Inter-Vehicular Communication in October 2010 has led to new research directions and results. In the 2010 seminar, the participants proposed to put more focus on the applications and the assessment of their benefits, first ignoring too many technical details and then adding technological constraints successively. Several research results appeared to have followed the proposed roadmap, see for example [1, 2, 3].

The working group then did a 'gap analysis', touching the following two issues: a) to what extend should IVC research 'tailor' a specific technology and b) should the interaction with other research communities be strengthened? The working group identified fault tolerance, reliable consensus and cognition as computer science fields that should be more involved in IVC research. In addition, the engineering and deployment issues appear to deserve more attention, thus, an easy answer on how much 'tailoring' and how much 'general results' are needed could not be given.

As a result of the discussions, the following research topics showed great promise to the working group members:

- Group communication, application protocols and reliable consensus. While in the last decade the focus was on one-hop broadcast messages, with coordinated maneuvering and automated driving a group of vehicles needs to communicate reliably, with a specified application protocol, to achieve reliable consensus. As vehicular traffic is full of protocols, it is no big wonder that maneuvering requires application protocols. However, group formation and dealing with the unreliable wireless channel brings interesting research questions in.
- Cognition and safety. The cooperation with experts from cognitive vehicles and from automotive safety should be strengthened since application requirements come from detecting dangerous traffic situations (including pedestrians and bicyclists) as well as of safe driving strategies.

- Self-organizing systems. The promise made by the IVC community to design self-organizing networks is not enough for deployment or market entry, as many field operational tests clearly show: the radical new design of the network alone and the sheer scale of the system requires many innovations in the whole IT management chain. Here again, principles from self-organizing systems and the whole self-x movement might help while being complemented by existing IT management techniques.
- Flexible and adaptable communication architectures that can adjust to changing contexts, technologies and application mixes and that allows the system to evolve over time. This would also open a chance for building networks that go beyond IVC and would lead towards an Internet-of-Things approach.

With future cooperative automated vehicles, all the aspects mentioned above require and deserve further efforts in the field of inter-vehicular communication.

References

1 S. Joerer, M. Segata, B. Bloessl, R. Lo Cigno, C. Sommer, and F. Dressler, "To Crash or Not to Crash: Estimating its Likelihood and Potentials of Beacon-based IVC Systems," in *4th IEEE Vehicular Networking Conference (VNC 2012).* Seoul, Korea: IEEE, November 2012, pp. 25–32.

2 W. Klein Wolterink, G. Heijenk, and G. Karagiannis, "Constrained Geocast to Support Cooperative Adaptive Cruise Control (CACC) Merging," in *2nd IEEE Vehicular Networking Conference (VNC 2010).* Jersey City, NJ: IEEE, December 2010, pp. 41–48.

3 N. An, M. Maile, D. Jiang, J. Mittag, and H. Hartenstein, "Balancing the Requirements for a Zero False Positive/Negative Forward Collision Warnings," in *10th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2013).* Banff, Canada: IEEE, March 2013, pp. 191–195.

4.3 Best Practices for Field Operational Testing

David Eckhoff, Andreas Festag, Marco Gruteser, Florian Schimandl, Michele Segata, and Elisabeth Uhlemann

4.3.1 Introduction

The performance evaluation of vehicular network technology and applications is a non-trivial challenge. Field testing a system plays an important role in such evaluations and in advancing scientific knowledge. It is not only necessary to assess network performance in a real environment but also to discover previously unaccounted or unknown system properties. While some of these benefits can also be achieved with small-scale experimentation, only Field Operational Tests (FOTs) can evaluate systems at scale and cover a much wider range of scenarios.

Data collected in these trials can furthermore be used as input for the creation and validation of both analytical and simulation models, and therefore improve their quality and relevance. At the same time, conducting meaningful field operational tests is challenging. They often involve complex systems with proprietary technology components, which can make it difficult to interpret the results and to match them to analytical or simulation models.

As vehicular network research and development has moved into a stage of extensive field trials, this working group has discussed the potential impact on academic research and ways to improve collaboration between academia and the operators of field operational tests. We begin with a short overview of ongoing efforts and discuss why field testing can be a necessary and valuable asset for academia and the scientific field. From those discussions we distill recommendations for both academia and trial operators to further improve the value and benefit of future field trials.

4.3.2 Past and Current Efforts

Ongoing field trials in vehicular networks span evaluation topics ranging from driver acceptance of applications to network performance in highly congested environments.

In the United States, the Safety Pilot Model Deployment at the University of Michigan Ann Arbor hosts about 3,000 vehicles equipped with DSRC devices to test the effectiveness of the technology in real world conditions, to measure how drivers adapt to the technology, and to identify potential safety benefits. Results from this test are expected to influence a potential National Highway Traffic Safety Administration (NHTSA) rule making, which could make DSRC technology mandatory.

In addition to this more application oriented testing, the Crash Avoidance Metrics Partnership (CAMP) Vehicle Safety Communications 3 (VSC3) Consortium is conducting field trials under the connected vehicle technology research program of the USDOT. This activity studies scalability aspects of vehicle safety communications that will preserve the performance of vehicle safety applications in both congested as well as uncongested communication environments [1].

In Europe, the German simTD project [2] studied vehicle-to-vehicle and vehicle-to-infrastructure communication based on ad hoc and cellular networks. The trial addressed traffic efficiency applications (traffic monitoring, traffic information and navigation, traffic management) and safety applications (local danger alert, driving assistance) and included vehicles, road side units as well as traffic management centers. The tests were conducted with fleets of vehicles with professional, instructed drivers for scenario testing in a controlled environment and with free-flowing vehicles. The simTD project coincides with trials in other countries across Europe, for which the European project DRIVE C2X [3] enabled a common test methodology and technological basis. Objectives of the tests are to validate the vehicle communication technology and to collect data for impact assessment of the technology on safety and traffic efficiency.

4.3.3 Benefits and Challenges of Using FOT Data

The benefits that the academic community could gain from FOTs are manifold. Research groups studying Inter-Vehicle Communication (IVC) and Intelligent Transportation Systems (ITS) technologies in general, could use the data

ollected during FOTs even after the end of the project, investigating aspects that were not covered by the original FOT objectives. An important requirement for this to be possible is that all needed meta data is logged and documented.

Simulative evaluation of communication strategies and applications in vehicular networks heavily relies on data collected in field trials to further bridge the gap between simulation and reality and hence to increase the trustworthiness of simulation results. For example, the amount of work recently published on channel models for vehicular networks (including path-loss analysis, shadowing models for buildings and vehicles) requires real world data to be validated. The more data is available the better can these models be adjusted and therefore improved. But also MAC layer models would benefit from more extensive experimental validation. The results of network oriented FOTs (e.g., CAMP VSC3) but also more general ones (e.g, DRIVE C2X [3], simTD [2]) can therefore be extremely helpful to validate such models.

Not only can network models be improved with help of field trials but also can they help advance mobility related research. Vehicle traces collected during field tests, for example, could be used to derive behavioral models, which are becoming extremely important for the evaluation of safety applications. Further possible benefits include the tuning of psychological driver models (e.g., the following of recommendations made by the on-board unit) , the parameterization of car following models, or establishing a default mobility scenario to make simulations more comparable towards each other.

However, data access requested by institutions not directly involved in the FOTs requires some preconditions. First, there is a necessity for an in-depth documentation of the published dataset with not only the present goals of the FOT in mind, but also considering that the data will be used for other purposes. This requires a detailed and exact description of the experiments and the data format. Of course, making data publicly available requires specific solutions for data storage policies and locations, as data must be available to download to a potentially wide number of academic research groups, even after the FOT has long been completed.

4.3.4 Recommendations

Although the research community has a long history of analytical evaluations and simulations, the prior experience from FOTs is still rather limited. Since analytical results are used to validate simulators, and vice versa, the gain from having a third tool for performance evaluation is obvious. Some models of real world phenomena already exist in academic research and are used both in simulations as well as analytical evaluations. Examples include wireless channel models, modeling of shadowing and propagation based on different types of road environments, vehicular mobility models and data traffic patterns. The results of FOTs can be used to update and enhance these models, such that large scale simulations based on real vehicle traces are possible.

However, in order to fully benefit from FOTs, academic researchers need to become familiarized with the potentials, the limitations, the benefits and the drawbacks of this new tool. In addition, since the money and resources to conduct large scale field trials are often not available to academic researchers, they must rely on and collaborate with industry and governmental institutions. Unfortunately, the goals of FOTs outcomes are not necessarily the same for vehicular manufacturers, road operators, and academic researchers.

It is therefore of essence that we learn how to successfully convey the benefits of giving academic researchers access to FOT data. If we compile a list of possible use cases for that, it will facilitate a request to collect a specific set of data and record the relevant meta data needed to achieve a certain goal and to enable reproducible results. Further, there is a need to better understand the goals and the interests of the different stakeholders in FOT from the beginning, so that motivations to tightly restrict access to field test data can be identified and addressed.

Generated data and the respective scenarios, comprising the conditions under which the data was collected, should be documented in detail so that all stakeholders are able to work with the information easily. Naturally, this entails that resources should be allocated already in project planning processes for data documentation as well as archival, maintenance, and distribution after the project.

In-depth, general purpose documentation can not only improve the flow of information from the stakeholders to third parties in academia. Traceability can also improve the exchange of knowledge from one (completed) FOT to another, something that is oftentimes relying on stakeholders active in both FOTs.

Due to the complexity of many large scale tests, we recommend that validation activities (e.g., using simulation or analytical methods) are planned for and integrated even during the early testing stages of a field trial. Furthermore, small scale tests ("dress rehearsals") should be conducted (preferably already in an early project phase) in order to test processes and data collection deeply as well as pre-evaluate results. This also includes the allocation of time periods used analyze and revise the system and experiment design before conducting the final experiments.

4.3.5 Conclusion

FOTs represent an enormous resource for the entire vehicular networking community and are of utmost importance for the development of IVC technology. While FOTs are mainly conducted by the automotive industry, the outcomes of such trials can be also of huge value for academia. The successful collaboration with third parties, however, poses some challenges.

In particular, the academic community should try to be more involved during the trial design phase and communicate the exact requirements for the collected data. Non-involved parties from both academia and industry can also hugely benefit from publicly available data, if all the needed meta data is logged and a general purpose documentation is included. This does not only allow for the development of better, more realistic analytical and simulation models but can also help conduct future FOTs.

References

1 M. Lukuc, "V2V Interoperability Project," in *USDOT ITS Connected Vehicle Workshop*, Chicago, IL, September 2012.

2 H. Stübing, M. Bechler, D. Heussner, T. May, I. Radusch, H. Rechner, and P. Vogel, "simTD: A Car-to-X System Architecture for Field Operational Tests," *IEEE Communications Magazine*, vol. 48, no. 5, pp. 148–154, May 2010.

3 R. Stahlmann, A. Festag, A. Tomatis, I. Radusch, and F. Fischer, "Starting European Field Tests for Car-2-X Communication: The DRIVE C2X Framework," in *18th ITS World Congress and Exhibition*, Orlando, FL, October 2011.

4.4 IVC Applications

Natalya An, Wai Chen, Raphael Frank, Mario Gerla, Liviu Iftode, Stefan Joerer, Renato Lo Cigno, Florian Schimandl, Ozan Tonguz, and Peter Vortisch

As a working group, the Applications Working Group discussed some key emerging issues related to different applications of VANETs in the market place. These discussions included safety, efficiency, and entertainment applications. Below, we provide a summary of the key issues discussed by the Applications Working Group

4.4.1 Why DSRC applications are not yet on the market?

The group felt that VANET research, in general, is at crossroads since there are some rumors and speculations that FCC might take back the 75 MHz bandwidth it had allocated to safety applications at 5.9 GHz in the last decade or so. To this end, FCC is considering to open up this bandwidth to the use WiFi for commercial applications which could complicate the overall picture considerably. The main reason for this appears to be the reluctance of car manufacturers to install DSRC radios in their vehicles due to cost considerations.

On the other hand, US Department of Transportation (DoT) has allocated about 100 Million USD for field trials in 6 different locations of the USA to demonstrate the huge benefits of using DSRC-equipped vehicles to safety. The field trial in Detroit, Michigan, for instance, was initially designed as an 18 months experiment and has been continuing for the last one year or so. It involves about 3000 drivers selected from different age groups, professions, education levels, gender, etc. in an effort to collect significant empirical data for demonstrating how the use of DSRC radios could increase the safety on the road (in urban areas and highways) significantly. The main motivation behind these massive field trials and the investment made by the US DoT is to collect convincing data (in a statistical sense) to present to the Congress for passing legislation for mandating the use of DSRC radios. If this effort succeeds, within couple of years one can hope to see DSRC radios installed in every car sold in the USA as a safety feature (similar to seat belts and air bags).

Another interesting development is the fact that several auto manufacturers are considering solutions based on cellular communications. As an example, General Motors (GM) has recently announced an agreement with AT&T to use AT&T equipment in their vehicles for Internet access and other services. This entails the use of an LTE modem installed in GM cars and the use of LTE (or LTE-A) networks of AT&T for several services. It is known that Mercedes-Benz and other car manufacturers are also considering similar solutions for providing different services to their customers. This new development, however, does not seem to prioritize safety as the key application. So, it remains unclear and very doubtful whether safety can be supported at a significant level with cellular communications.

Based on these developments, two major outcomes seem plausible:

- Based on the aforementioned field trials, assuming the collected data provide convincing evidence about the benefits of DSRC radios in reducing accidents and enhancing safety of driving, DOT passes legislation and pushes the car manufactures to use DSRC.
- DSRC applications are gradually introduced into the market place and more and more drivers install DSRC radios in their vehicles as they see the benefit. This will involve after-market DSRC devices for legacy cars and perhaps the installation of DSRC radios into only new high-end cars.

In both cases, however, there has to be convincing evidence that safety can be improved substantially via the use of DSRC technology. In this sense, the 6 field trials in the USA (and other similar large field trials in other parts of the world) will carry a lot of weight in providing reliable and significant data to the Federal Government and to the public.

At this juncture, viable business models might also be important in convincing the stakeholders to go ahead and mandate the DSRC technology. There was a general consensus that the 'golden triangle' for mandating the DSRC technology might be the government-car manufacturers-insurance companies, as the key stakeholders. However, these stakeholders have different objectives: for example, the US government's main objective is to reduce the 35,000 fatalities every year due to car accidents while the car manufacturers and insurance companies see the introduction of DSRC radios as merely another business transaction and by incorporating this new technology they would like to increase their profit margins (e.g., insurance companies could reduce/increase the premiums they charge depending on whether or not a car is equipped with a DSRC radio). While the cooperation of these stakeholders will clearly expedite the process, the role of the government in serving as a catalyst cannot be underestimated.

4.4.2 What can be done in academia?

It was noted by our group that the networking and communications people in VANET research should have a closer collaboration with the traffic safety people in the transportation domain (most of the current planning activity is done by these people and does NOT involve V2V or V2I communications) as these are the key people who determine how traffic planning is currently done and what are the underlying safety concerns. By better understanding their current thinking, the ongoing VANET research at universities could be more focused and direct in addressing the current needs and shortcomings of the existing system.

In going forward, it will also be important to convince automakers and drivers about the safety benefits of using DSRC technology. A conscientious and orchestrated effort in this direction could certainly contribute to the adoption of DSRC technology. However, as mentioned before, since the motivation of all car manufacturers is to make money and increase their profit margins, perhaps safety should not be the first application that our research should offer to automakers. Instead, perhaps other applications that DSRC technology can enable (such as efficiency and entertainment) should come first and safety should be tagged to these applications which might have potential as a revenue stream.

Another trend that was discussed is the growing interest in autonomous driving (AD). After the advent of Google's autonomous driving in Las Vegas and Nevada, some of the car manufacturers (such as GM, Nissan, Volkswagen, etc.) are heavily invested in R&D for autonomous driving. It is clear, however, that the autonomous vehicles so far do NOT emphasize the use of inter-vehicle communications (IVC) but, rather, rely on the presence of a very large number of sensors and actuators to 'sense' their environment and navigate accordingly, hence the name 'autonomous'. It was noted that this might change in the coming years as IVC should and probably will become a major component in autonomous vehicles as well. This is because an autonomous vehicle is ultimately a mobile robot and in decision making as a mobile robot its most challenging task is to make correct decisions at an intersection (especially at intersections which are not regulated with traffic lights or other traffic signals). It is clear that the rotating cameras, radars, and lidars that exist on autonomous vehicles are essentially LOS devices and cannot always discern objects (and other vehicles) which are on orthogonal roads at an intersection and, therefore, might be N-LOS. Our group decided that we should capitalize on this new trend and try to convince Google and other parties involved in autonomous driving about the huge benefits that could be reaped by the use of DSRC technology and IVC. So, a conscientious effort on how to integrate IVC to autonomous driving will be very timely and very helpful.

4.4.3 Cooperative Autonomous Driving

Continuing along this promising direction, potential new applications where integration of IVC with autonomous driving can be easily achieved were also discussed.

One application where autonomous driving would benefit from the presence of DSRC technology and IVC was identified as lane merging. All collaborative applications that require cooperation could also benefit from cooperative autonomous driving.

An interesting observation that was made is the fact that autonomous driving by definition is currently a local concept whereas integrating it with IVC could lead to large-scale benefits as it makes the autonomous vehicles much more aware of the state of the network.

Autonomous vehicles will be coming to the market place very slowly (presumably by 2020). Even then, we will probably observe a slow penetration rate due to cost issues as well as other issues (liability etc.). It is clear that many more vehicles can be equipped with communications capability and for less money before 2020, so autonomous vehicles can profit from other non-autonomous vehicles, but those that can communicate with the autonomous vehicles (if the autonomous vehicles are also equipped with DSRC radios). This provides yet another motivation for the integration of IVC with autonomous driving.

It is no secret that certain capabilities that make autonomous vehicles truly 'autonomous' are the massive and sometimes expensive sensors (such as rotating cameras on the roof of the Google autonomous vehicle, radars, lidars, etc.). Using DSRC radios might obviate the use of some of these expensive sensors in autonomous vehicles, thus reducing the cost of autonomous vehicles substantially which, in turn, will accelerate their massive adoption and use.

4.4.4 Definition of an "Application"

The last issue discussed in the Applications Working Group was the concern about lack of a common agreement on the definition of 'an application' when we approach things in a top-down manner. Different stakeholders see applications differently which create some ambiguity and undesired outcomes. How to resolve this issue does not seem very clear. As an example, is it correct to see vehicles as a computer/smartphone where applications can be downloaded ?

It was agreed that defining some common denominator about the definition of certain applications and their requirements (e.g., a safety application) would be very helpful. For example, it seems very difficult to influence what car manufacturers would like to see as an application. To take this example further: if different car manufacturers do not agree on the definition of safety (consider, for instance, the need for having situation awareness in vehicles as a safety application), then it might be very difficult to achieve concrete results on safety applications.

In going further, it will be crucial for car manufacturers to agree at a minimum level on the definition of an application (and its requirements). If this can be achieved, then third-party vendors can build upon those minimum requirements and promote new applications of IVC. It seems clear that here also the DoTs and the Federal Governments will have a crucial role to play.

Participants

- Natalya An
KIT – Karlsruhe Institute of Technology, DE
- Claudio Casetti
Polytechnic Univ. of Torino, IT
- Wai Chen
China Mobile Research Institute – Beijing, CN
- Falko Dressler
Universität Innsbruck, AT
- David Eckhoff
Univ. Erlangen-Nürnberg, DE
- Andreas Festag
TU Dresden, DE
- Raphaël Frank
University of Luxembourg, LU
- Mario Gerla
University of California – Los Angeles, US
- Javier Manuel Gozalvez Sempere
University Miguel Hernandez – Elche, ES
- Marco Gruteser
Rutgers University – New Brunswick, US
- Jérôme Härri
EURECOM – Biot, FR
- Hannes Hartenstein
KIT – Karlsruhe Institute of Technology, DE
- Geert Heijenk
University of Twente, NL
- Liviu Iftode
Rutgers Univ. – Piscataway, US
- Stefan Jörer
Universität Innsbruck, AT
- Frank Kargl
Universität Ulm, DE
- Renato Lo Cigno
University of Trento – DISI, IT
- Giovanni Pau
University of California – Los Angeles, US
- Jonathan Petit
University of Twente, NL
- Björn Scheuermann
HU Berlin, DE
- Florian Schimandl
TU München, DE
- Michele Segata
University of Trento – DISI, IT
- Christoph Sommer
Universität Innsbruck, AT
- Tessa Tielert
KIT – Karlsruhe Institute of Technology, DE
- Ozan K. Tonguz
Carnegie Mellon University, US
- Elisabeth Uhlemann
Halmstad University, SE
- Peter Vortisch
KIT – Karlsruhe Institute of Technology, DE

Report from Dagstuhl Seminar 13401

Automatic Application Tuning for HPC Architectures

Edited by

Siegfried Benkner[1], Franz Franchetti[2], Hans Michael Gerndt[3], and Jeffrey K. Hollingsworth[4]

1 **Universität Wien, AT,** siegfried.benkner@univie.ac.at
2 **Carnegie Mellon University – Pittsburgh, US,** franzf@ece.cmu.edu
3 **TU München, DE,** gerndt@in.tum.de
4 **University of Maryland – College Park, US,** hollings@cs.umd.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13401 "Automatic Application Tuning for HPC Architectures". This workshop featured a series of talks and four breakout sessions on hot topics in the are of autotuning. The field of autotuning for HPC applications is of growing interest and many research groups around the world are currently involved. However, the field is still rapidly evolving with many different approaching being taken to autotuning. This workshop provided an opportunity to discuss these many approaches, and help to unify the terminology used by different groups.

Seminar 29. September to 4. October, 2014 – www.dagstuhl.de/13401
1998 ACM Subject Classification D.2 Software Engineering, D.2.1 Requirements/Specifications, D.2.8 Metrics, D.2.11 Software Architectures
Keywords and phrases Parallel Computing, Programming Tools, Performance Analysis and Tuning
Digital Object Identifier 10.4230/DagRep.3.9.214

1 Executive Summary

Siegfried Benkner
Franz Franchetti
Hans Michael Gerndt
Jeffrey K. Hollingsworth

Parallel computer systems especially for High Performance Computing are getting increasingly complex. The reasons are manyfold. HPC systems today with a peak performance of several petaflops have hundreds of thousands of cores that have to be able to work together efficiently. Those machines have a deep hierarchy, which has to be understood by the programmer to tune his program so that it profits from higher interconnection rates. In addition, to reduce the power consumption of those systems, advanced hard- and software techniques are applied, such as the usage of GPUs that are highly specialized for regular data parallel computations via simple compute cores and high bandwidth to the graphics memory. Another technique is to reduce the clock frequency of processors when appropriate, e.g. when the application or phases of the execution are memory bound. This transforms a homogeneous system into a heterogeneous system, which complicates programming tasks such as load balancing and efficient communication.

The complexity of today's parallel architectures has a significant impact on the performance of parallel applications. Due to the high amount of energy and money being lost because of the low processor utilization, application developers are now investing significant time to tune their codes for the current and emerging systems. This tuning is a cyclic process of gathering data, identifying code regions that can be improved, and tuning those code regions.

There are a growing number of autotuning researchers in Europe, the United States, and Asia. However, there are relatively few opportunities for these researchers to meet together. The unique format of a Dagstuhl seminar provides the opportunity to bring together researchers from around the world that are using different approaches to autotuning.

This workshop brought together those people working on autotuning with people working on performance analysis tools. While the analysis tools indicate performance problems, their combination with performance tuning might make those tools even more successful. The presentations of experts in both areas will increase the interest and the knowledge of the techniques applied in the other area. It will steer future collaborations and might also lead to concrete ideas for coupling performance analysis and performance tuning tools.

The workshop was driven by the European FP7 project AutoTune that started on October 15th, 2011. It is the goal of AutoTune to implement the Periscope Tuning Framework based on the automatic performance analysis tool Periscope. It will couple Periscope's performance analysis with performance and energy efficiency tuning in an online approach.

Performance Analysis. Performance analysis tools support the programmer in the first two tasks of the tuning cycle. Performance data are gathered during program execution by monitoring the application's execution. Performance data are both summarized and stored as profile data or all details are stored in so called trace files. In addition to application monitoring, performance analysis tools also provide means to analyze and interpret the provided performance data and thus to detect performance problems. The analysis is either supported by graphical display or by annotating the source code.

State of the art performance analysis tools fall into two major classes depending on their monitoring approach: profiling tools and tracing tools. Profiling tools summarize performance data for the overall execution and provide information such as the execution time for code regions, number of cache misses, time spent in MPI routines, and synchronization overhead for OpenMP synchronization constructs. Tracing tools provide information about individual events, generate typically huge trace files and provide means to visually analyze those data to identify bottlenecks in the execution.

Representatives for these two classes are gprof, OMPP and Vampir. Gprof is the GNU Profiler tool. It provides a flat profile and a callpath profile for the program's functions. The measurements are done by instrumenting the application. OmpP is a profiling tool for OpenMP developed at TUM and the University of Tennessee. It is based on instrumentation with Opari and determines certain overhead categories of parallel regions. In contrast to the previous two tools, Vampir is a commercial trace-based performance analysis tool from Technische Universität Dresden. It provides a powerful visualization of traces and scales to thousands of processors based on a parallel visualization server.

The major research challenges in the development of PA tools are to automate the analysis and to improve the scalability of the tools. Automation of the analysis is important to facilitate the application developer's task. Starting from the formalization of performance properties in the European-American working group APART (http://www.fz-juelich.de/apart), automatic performance analysis tools were developed. Paradyn from University of Wisconsin was the first automatic online analysis tool. Its performance consultant guided the search for performance bottlenecks while the application was executing. The most important representatives are SCALASCA and Periscope. SCALASCA is an automatic performance analysis tool developed at Forschungszentrum Jülich and the German Research School on Simulation Sciences. It is based on performance profiles as well as on traces. The automatic trace analysis determines MPI wait time via a parallel trace replay on the application's processors after the application execution terminated.

Periscope is an automatic performance analysis tool for highly parallel applications written in MPI and/or OpenMP currently under development at Technische Universität München. It is a representative for a class of automatic performance analysis tools automating the whole analysis procedure. Unique to Periscope is that it is an online tool and it works in a distributed fashion. This means that the analysis is done while the application is executing (online) and by a set of analysis agents, each searching for performance problems in a subset of the application's processes (distributed). The properties found by Periscope point to code regions that might benefit from further tuning.

Performance Autotuning. The central part of the tuning process is the search for the best combination of code transformations and parameter settings of the execution environment. This creates an enormous search space, which further complicates the whole tuning task. As a result, much research has been dedicated to the area of autotuning in the last years and many different ideas have been gathered. These can be grouped into four categories:

- self-tuning libraries for linear algebra and signal processing like ATLAS, FFTW, OSKI and SPIRAL;
- tools that automatically analyze alternative compiler optimizations and search for their optimal combination;
- autotuners that search a space of application-level parameters that are believed to impact the performance of an application;
- frameworks that try to combine ideas from all the other groups.

The first category contains special purpose libraries that are highly optimized for one specific area. The Automatically Tuned Linear Algebra Software (ATLAS) supports the developers in creating numerical programs. It automatically generates and optimizes the popular Basic Linear Algebra Subroutines (BLAS) kernels for the currently used architecture. Similarly, FFTW is a library for computing the discrete Fourier transform on different systems. Due to the FFTW design, an application using it will perform well on most architectures without modification.

However, the growing diversity of parallel application areas requires a more general autotuning strategy. Thus, substantial research has been done in a different application-independent approach of autotuning. This is based on the automatic search for the right compiler optimizations on the specific platform. Such tools can be separated into two groups according to their methodology: iterative search tools and those using machine learning techniques. There has been much work in the first category. All these tools share the idea of iteratively enabling certain optimizations. They run the compiled program and monitor its performance. Based on the outcome, they decide on the new tuning combination. Due to the huge size of the search space, these tools are relatively slow. There exists an algorithm called combined elimination (CE) that greatly improves the previous search-based methods.

The second branch of compiler-based autotuners applies a different strategy to look for the best optimization settings. They use knowledge about the program's behavior and machine learning techniques to select the optimal combination. This approach is based on an automatically built per-system model, which maps performance counters to good optimization options. This model can then be used with different applications to guide their tuning. Current research work is also targeting the creation of a self-optimizing compiler that automatically learns the best optimization heuristics based on the behavior of the underlying platform.

Among the tools in the third category is the Active Harmony system. It is a runtime parameter optimization tool that helps focus on the application-dependent parameters that are performance critical. The system tries to improve performance during a single execution based on the observed historical performance data. It can be used to tune parameters such as the size of a read-ahead buffer or what algorithm is being used (e.g., heap sort vs. quick sort). As compared with Active Harmony, the work from Nelson uses a different approach that interacts with the programmer to get high-level models of the impact of parameter values. These models are then used by the system to guide the search for optimization parameters. This approach is called model-guided empirical optimization where models and empirical techniques are used in a hybrid approach.

Popular examples for the last group of autotuning tools are the newly released Parallel Active Harmony, and the Autopilot framework. The Parallel Active Harmonyis a combination of the Harmony system and the CHiLL compiler framework. It is an autotuner for scientific codes that applies a search-based autotuning approach. While monitoring the program performance, the system investigates multiple dynamically generated versions of the detected hot loop nests. The performance of these code segments is then evaluated in parallel on the target architecture and the results are processed by a parallel search algorithm. The best candidate is integrated into the application. The second popular example in this group is the Autopilot. It is an integrated toolkit for performance monitoring and dynamical tuning of heterogeneous computational grids based on closed loop control. It uses distributed sensors to extract qualitative and quantitative performance data from the executing applications. This data is processed by distributed actuators and the preliminary performance benchmark is reported to the application developer.

Energy efficiency autotuning. Multi-Petascale supercomputers consist of more than one hundred thousand processing cores and will consume many MW of electrical power. Energy efficiency will be crucial for both cost and environmental reasons, and may soon become as important as pure peak performance. This is exemplified by the fact that since a few years the TOP500 list (http://www.top500.org/) also contains power consumption values. Current procurements for high-end supercomputers show that the cost for electricity and cooling is nearly as high as for the hardware, particularly in countries with high energy costs such as Germany. Power consumption is considered one of the greatest challenges on the road to exascale systems.

Dynamic frequency and voltage scaling provides a mechanism to operate modern processors across a broad range of clock frequencies and voltage levels, allowing to trade off performance vs. energy consumption. Overall frequency scaling ideas are based on Advanced Configuration and Power Interface (ACPI, http://www.acpi.info/) specification with Intel's SpeedStep implementation or Cool'n'Quiet by AMD, respectively. Processors like Intel's Sandy Bridge are fully compliant with ACPI. Sets of utilities to exploit these techniques are available, and ideas to use them for complete jobs in multi user HPC clusters have already been described.

Whereas dynamic frequency scaling is commonly used in laptops, the impact and usability in HPC is still quite challenging. For applications using several hundreds or thousands of cores, uncoordinated manipulation of the frequency by some background daemon would introduce a new source of OS jitter. Moreover, changing the processor frequency requires on the order of milliseconds and only yields a benefit if a major part of an application can be run in a given mode continuously. Typically, lowering the CPU frequency can yield a 10% decrease in power consumption while increasing the application runtime by less than 1%. However, the impact of lowering the frequency and voltage on the application performance depends on whether it is CPU, memory, cache or I/O bound. Code regions that are CPU or cache bound can take advantage of higher frequencies, whereas regions that are memory or I/O bound experience only minor performance impacts when reducing the frequency. Therefore it is essential to identify applications and those parts of them that are appropriate for running within a specific power envelope without sacrificing too much performance.

Different metrics for performance, cost, energy, power, cooling and thermal conditions may apply for different usage and optimization scenarios e.g.

- minimizing the energy consumption by reducing the performance of an application by a given percentage
- considering outside temperature conditions, i.e, if it is cold outside and free cooling is applied, an increased power consumption by the compute nodes might be tolerated
- optimizing the total cost of ownership (including baseline investment, power and cooling) for given throughput requirements.

It is quite cumbersome to investigate all these conditions and the various frequency settings manually. Therefore automatic tools are required to automatically identify suitable applications and particular code regions, and finally automatically tune the frequency and power settings to yield optimal results for the desired objectives.

Thematic Sessions

The seminar was organized as a series of thematic sessions. An initial session comprised two overview presentations about performance analysis and measurement tools as well as a general introduction to autotuning, setting the overall context for the seminar. A session on support tools covered code restructuring techniques, testing environments, and performance repositories for autotuning. Two sessions on infrastructures provided insights into frameworks and environments, language support for autotuning as well challenges and requirements in the context of very large-scale systems. A session on energy efficiency tuning gave insight into the challenges and recent developments in optimizing HPC systems and applications with respect to energy consumption. A session on accelerator tuning covered various issues in tuning for GPUs and accelerated parallel systems. A session on techniques covered various topics related to performance-guided tuning, modeling, and scalability. A session on tools covered recent developments in empirical autotuning, semantics support for performance tools and autotuners as well as synthesis of libraries. Various topics related to the tuning of message-passing applications and I/O-related autotuning were covered in a session on MPI and I/O tuning. The session on compiler transformations covered compiler transformations for multi-objective tuning, techniques for tuning irregular applications, as well as on language and compilation support for analysis of semantic graphs.

2 Table of Contents

Working Groups

3 Overview of Talks

3.1 Autotuning of Pipeline Patterns for Heterogeneous Manycore Architectures

Enes Bajrovic (Universität Wien, AT)

We have developed a high-level, component-based approach to programming heterogeneous manycore architectures with support for parallel patterns. Central to this approach are multi-architectural components, which encapsulate different implementation variants of application functionality tailored by expert programmers for different core types of a heterogeneous system. End users construct applications at a high level of abstraction using component interfaces and high-level coordination primitives, which are then mapped to a task-based runtime that selects and dynamically schedules the best component implementation variants for efficient parallel execution on a heterogeneous manycore architecture. In this talk we present our ongoing work on automatic performance tuning of pipeline patterns and the underlying pattern runtime for CPU/GPU based architectures as currently done within the European AutoTune project.

References

1 R. Miceli, G. Civario, A. Sikora, E. Cesar, M. Gerndt, H. Haitof, C. Navarrete, S. Benkner, M. Sandrieser, L. Morin, and F. Bodin. *AutoTune: A Plugin-Driven Approach to the Automatic Tuning of Parallel Applications*, In *Applied Parallel and Scientific Computing* (P. Manninen and P. Oester, eds.), vol. 7782 of *LNCS*, pp. 328–342, Springer, 2013.

2 S. Benedict, V. Petkov, and M. Gerndt. *PERISCOPE: An Online-Based Distributed Performance Analysis Tool.* In: Tools for High Performance Computing 2009 (M. S. Mueller, M. M. Resch, A. Schulz, and W. E. Nagel, eds.), pp. 1–16, Springer, 2010

3 E. Bajrovic, S. Benkner, J. Dokulil, M. Sandrieser. *Autotuning of Pattern Runtimes for Accelerated Parallel Systems.* In: PARCO 2013, September 2013, Munich, Germany, 10-13 Sept 2013, Munich, Germany (2013)

4 S. Benkner, E. Bajrovic, E. Marth, M. Sandrieser, R. Namyst, and S. Thibault. *High-Level Support for Pipeline Parallelism on Manycore Architectures.* In: Euro-Par 2012, vol. 7484, pp. 614–625, 2012.

5 S. Benkner, S. Pllana, J. L. Traeff, P. Tsigas, U. Dolinsky, C. Augonnet, B. Bachmayer, C. Kessler, D. Moloney, V. Osipov. *PEPPHER: Efficient and Productive Usage of Hybrid Computing Systems.* IEEE Micro, vol. 31, no. 5, pp. 28–41, Sep./Oct. 2011, DOI: 10.1109/MM.2011.67.

3.2 The Autotune Project

Siegfried Benkner (Universität Wien, AT)

The European AutoTune project develops the Periscope Tuning Framework (PTF) for performance and energy efficiency tuning of parallel applications on current and future multicore-based architectures. PTF extends Periscope, an automatic online and distributed performance analysis tool, with tuning plugins for automatic performance and energy efficiency tuning, closing the gap between performance analysis and tuning. The tuning plugins developed within the project include a GPU tuning plugin for HMPP/OpenCL codes, a plugin for tuning of pipeline patterns for heterogeneous manycore architectures, an energy efficiency tuning plugin, an MPI tuning plugin, and a compiler options exploration plugin. Wherever possible, the whole tuning process, consisting of automatic performance analysis and automatic tuning, will be executed online, i.e., during a single run of the application. This talk will present an overview of PTF, the underlying tuning model, and selected tuning plugins.

References

1 R. Miceli, G. Civario, A. Sikora, E. Cesar, M. Gerndt, H. Haitof, C. Navarrete, S. Benkner, M. Sandrieser, L. Morin, and F. Bodin. *AutoTune: A Plugin-Driven Approach to the Automatic Tuning of Parallel Applications*, In *Applied Parallel and Scientific Computing* (P. Manninen and P. Oester, eds.), vol. 7782 of *LNCS*, pp. 328–342, Springer, 2013.

2 S. Benedict, V. Petkov, and M. Gerndt. *PERISCOPE: An Online-Based Distributed Performance Analysis Tool.* In: Tools for High Performance Computing 2009 (M. S. Mueller, M. M. Resch, A. Schulz, and W. E. Nagel, eds.), pp. 1–16, Springer, 2010

3 E. Bajrovic, S. Benkner, J. Dokulil, M. Sandrieser. *Autotuning of Pattern Runtimes for Accelerated Parallel Systems.* In: PARCO 2013, September 2013, Munich, Germany, 10-13 Sept 2013, Munich, Germany (2013)

4 S. Benkner, E. Bajrovic, E. Marth, M. Sandrieser, R. Namyst, and S. Thibault. *High-Level Support for Pipeline Parallelism on Manycore Architectures.* In: Euro-Par 2012, vol. 7484, pp. 614–625, 2012.

3.3 High-Productivity and High-Performance Analysis of Filtered Semantic Graphs

Aydın Buluç (Lawrence Berkeley National Laboratory, US)

Joint work of Buluç, Aydın; Duriakova, Erika; Fox, Armando; Gilbert, John; Kamil, Shoaib; Lugowski, Adam; Oliker, Leonid; Williams, Samuel

Main reference A. Buluç, E. Duriakova, A. Fox, J. R. Gilbert, S. Kamil, A. Lugowski, L. Oliker, S. Williams, "High-Productivity and High-Performance Analysis of Filtered Semantic Graphs," in Proc. of the IEEE 27th Int'l Parallel and Distributed Processing Symp. (IPDPS'13), pp. 237–248, IEEE CS, 2013.

URL http://dx.doi.org/10.1109/IPDPS.2013.52

Graph theory is used to model large-scale complex systems in various scientific domains. Filtering on attributed semantic graphs enable a broad set of real applications that maintain important auxiliary information through attributes on individual edges and vertices. We achieve high performance in a high- productivity Python-based graph library (KDT) by utilizing a highly optimized sparse linear algebra based backend (Combinatorial BLAS), and automatically translating analytic queries via selective just-in-time embedded specialization (SEJITS).

References

1 Aydın Buluç, Erika Duriakova, Armando Fox, John Gilbert, Shoaib Kamil, Adam Lugowski, Leonid Oliker, and Samuel Williams. High-productivity and high-performance analysis of filtered semantic graphs. In *Proceedings of the IPDPS*. IEEE Computer Society, 2013.

3.4 HPCToolkit: Performance Tools for Tuning Applications on Heterogeneous Supercomputers

Milind Chabbi (Rice University, US)

Joint work of Chabbi, Milind; Murthy, Karthik; Fagan, Michael; Mellor-Crummey, John
Main reference M. Chabbi, K. Murthy, M. Fagan, J. Mellor-Crummey, "Effective Sampling-Driven Performance Tools for GPU-Accelerated Supercomputers," in Proc. of Int'l Conf. for High Performance Computing, Networking, Storage and Analysis (SC'13), pp. 43:1–43:12, ACM, 2013.
URL http://dx.doi.org/10.1145/2503210.2503299

Achieving higher performance under limited power is an open challenge for exascale computing. Modern supercomputers employ accelerators towards achieving this goal. Tuning applications employing CPUs and GPUs across an array of nodes demands a systemic performance perspective. In this talk, I introduce three key ideas developed by Rice University's HPCToolkit team to address these issues. First, we introduce CPU-GPU blame shifting – a technique to pinpoint and quantify idleness arising due to non overlapped CPU and GPU computations within a node. Second, we introduce a technique to identify and assess the impact of small set of sporadically poorly behaving nodes sabotaging the performance of entire application. Finally, we introduce a technique to model the performance impact of multiplexing GPUs among multiple processes.

3.5 Blue Gene Performance Data Repository

I-hsin Chung (IBM TJ Watson Research Center – Yorktown Heights, US)

As the high performance computer architectures become more complicated and larger in scale, it is a challenge to understand the utilization of system resources by applications. In this presentation, we introduce the Blue Gene Performance Data Repository, which automatically collects a wide range of performance information and stores it into a relational database for subsequent analysis. With minimal overhead to the users, the database collects performance data from different aspects of application execution including MPI, OpenMP, hardware counter, etc. With a standardized and uniform storage format, the database supports a wide range of queries and presentations. We will show examples of how this database can help users understand the application performance behavior and the system hardware usage. This information can also be used for software/hardware co-design for next generation systems. The Blue Gene Performance Data Repository is available to Blue Gene/Q users under an open source license.

3.6 Online Automatic Optimization of MPI Runtime Parameters with MPIT

Isaias Alberto Compres Urena (TU München, DE)

The point-to-point performance of MPI applications can be improved by adjusting the internal communication parameters of the MPI runtime. Point-to-point thresholds are set on good known values; however, the optimum value for a specific system may differ from the defaults. Performance can be further improved by selecting alternative internal implementations of collective operations. Default internal collective operations are set for specific process counts and buffer sizes and are also not always optimal. The space generated by all combinations of thresholds and internal algorithms is large; however, the search for optima for a particular application can be reduced based on its use of the MPI. The optimization process can be furder accelerated by exposing the relevant parameters through MPIT and performing the search online. Empirical data, collected on the SuperMUC, show that significant improvements in MPI performance can be achieved versus the default settings. These results were gathered with a modified version of the latest release of MPICH2.

3.7 Towards locality-based autotuning of irregular applications on HPC systems

Guojing Cong (IBM TJ Watson Research Center – Yorktown Heights, US)

As more analytics workloads emerge, tuning irregular applications for high performance becomes increasingly critical for the utilization of HPC systems. In this talk we show that locality-based tuning is an unified approach that can reduce communication overhead and improve cache performance simultaneously. We use multithreaded applications on cache-based NUMA architectures as our examples.

3.8 INSIEME: A compilation and runtime system for mulit-objective autotuning for parallel programs

Thomas Fahringer (Universität Innsbruck, AT)

Joint work of Jordan, Herbert; Thoman, Peter; Durillo, Juan; Gschwandtner, Philipp; Pellegrini, Simone; Fahringer, Thomas; Moritsch, Hans

Main reference H. Jordan, P. Thoman, J. J. Durillo, S. Pellegrini, P. Gschwandtner, T. Fahringer, H. Moritsch, "A multi-objective autotuning framework for parallel codes," in Proc. of the Int'l Conf. on High Performance Computing, Networking, Storage and Analysis (SC'12), pp. 10:1–10:12, IEEE CS, 2012.

URL http://dl.acm.org/citation.cfm?id=2389010

URL http://www.insieme-compiler.org

Efficient parallelization and optimization for modern parallel architectures is a time-consuming and error-prone task that requires numerous iterations of code transformations, tuning and performance analysis which in many cases have to be redone for every different target architecture.

We introduced an innovative autotuning compiler named INSIEME which consists of a compiler component featuring a multi-objective optimizer and a runtime system. It consists of a compiler component featuring a multi-objective optimizer and a runtime system. The multi-objective optimizer derives a set of non- dominated solutions, each of them expressing a trade-off among the different conflicting objectives such as execution time, energy consumption and efficiency. This set is commonly known as Pareto set in the field of multi-objective optimization research. Our search algorithm, which explores code transformations and their parameter settings, dynamic concurrency throttling (DCCT), and dynamic voltage and frequency scaling (DVFS) is based on Differential Evolution. Additionally, Rough sets are employed to reduce the search space, and thus the number of evaluations required during compilation. To make effective use of the resulting Pareto set of optimal solutions, each of them has to be made available at runtime. This is achieved by having the compiler generate a set of code versions per region, each corresponding to one specific solution. The runtime system then exploits the trade-off among the different objectives by selecting a specific solution (code version) for each region, based on context-specific criteria. We have implemented our techniques based on the Insieme Compiler and Runtime infrastructure. Our approach is generic and can be applied to arbitrary transformations and parameter settings. We demonstrate our approach by tuninng loop tiling, the nr. of threads, and clock frequency in cache sensitive parallel programs, optimizing for runtime, efficiency and energy.

The major contributions of this work include:

- the design of a novel autotuning architecture facilitating the consideration of multiple conflicting criteria simultaneously by interpreting the central task as a multi-objective optimization problem
- the combination of the search for optimal tile sizes, clock frequency settings, and the ideal number of threads for a parallel code section to minimize execution time and energy consumption, and maximize parallel efficiency into a single, multi-objective optimization problem
- the development of a multi-objective optimization algorithm capable of solving the combined problem using a reasonable number of iterative compilation steps

3.9 Towards Automating Black Belt Programming

Franz Franchetti (Carnegie Mellon University, US)

Main reference W. Yu, F. Franchetti, J. C. Hoe, T. Chen, "Highly Efficient Performance Portable Tracking of Evolving Surfaces," in Proc. of the 26th Int'l Parallel and Distributed Processing Symp. (IPDPS'12), pp. 296–307, IEEE CS, 2012.
URL http://dx.doi.org/10.1109/IPDPS.2012.36

Only a select few performance programmers in major processor vendors' software divisions, universities, national laboratories, and specialized companies have the skill set to achieve high efficiency on today's processors. Due to the labor-intensive nature of the work, the fast rate of newly arriving platforms, and the fact that performance tuning is more a black art than science, only the most important fundamental computation functions can be fully optimized. Today we are farther away than ever from John Backus' design criterion for the first Fortran compiler to automatically achieve close-to-human performance. Automatic performance tuning has bridged this gap for a few well-understood computational kernels like matrix multiplication, fast Fourier transform, and sparse matrix-vector multiplication, where systems like ATLAS, SPIRAL, FFTW, and OSKI showed that it is possible for automatic systems to compete with code that has been hand-tuned by "black belt programmers", which extracted the full performance potential from the target machines.

In this talk we investigate how black belt performance levels can be obtained automatically for irregular kernels with highly data-dependent control flow that operate on dynamic data structures, which makes the usual optimization methods impossible to apply. We focus on three illustrative examples: The first example is evaluating a logical equation in conjugate normal form, which is expressed as reduction across nested linked lists. The second example is the evolution of an interface surface inside a volume (e.g., a shock wave of an explosion), which translates into stencil operations on a contiguous sparse subset of pixels of a dense regular grid. The third example is a Monte Carlo simulation-based probabilistic power flow computation for distribution networks. In all cases we achieve a high fraction of machine peak, at the cost of applying aggressive optimization techniques that so far are beyond the capabilities of automatic tools. Autotuning and program generation played a major role in obtaining the final optimized implementation, as the necessary optimization techniques have a large parameter space and require extensive code specialization. In all cases the original code consisted of a few lines of C code, while the final code comprises hundreds to thousands of lines of architecture-specific SIMD intrinsic code and OpenMP or PThreads parallelization with custom synchronization. We extract lessons from the three examples on how to design future autotuning and program generation systems that will automate the optimization of such kernels.

References

1 W. Yu, F. Franchetti, J. C. Hoe, T. Chen. *Highly Efficient Performance Portable Tracking of Evolving Surfaces.* Proceedings of the 26th International Parallel and Distributed Processing Symposium (IPDPS), 296-307, 2012.
2 T. Cui and F. Franchetti. *Autotuning a Random Walk Boolean Satisfiability Solver.* roceedings of The Sixth International Workshop on Automatic Performance Tuning (iWAPT), 2011.

3.10 Crowdsourcing autotuning: challenges and possible solutions

Grigori Fursin (INRIA Saclay – Île-de-France – Orsay, FR)

Main reference G. Fursin, "Collective Mind: cleaning up the research and experimentation mess in computer engineering using crowdsourcing, big data and machine learning," INRIA technical report, arXiv:1308.2410v1 [cs.SE], 2013; also available on HAL (HAL-00850880).
URL http://arxiv.org/abs/1308.2410
URL http://hal.inria.fr/hal-00850880

Empirical program and architecture tuning combined with run-time adaptation and machine learning has been demonstrating good potential to improve performance, power consumption and other important metrics of computer systems for more than a decade. However, it is still far from the widespread production use due to unbearably long exploration and training times, dramatic growth in the amount of experimental data ("big data"), ever changing tools and their interfaces, lack of a common experimental methodology, and lack of unified mechanisms for knowledge building and exchange apart from publications where reproducibility of results is often not even considered.

We present our long-term holistic vision and a plugin-based Collective Mind infrastructure with unified web services to systematize characterization and optimization of computer systems through crowdtuning, extensible repositories of knowledge, and machine learning. In this cooperative approach, multi-objective program and architecture characterization and tuning is transparently distributed among many participants while consolidating and unifying existing techniques and tools, and utilizing any available mobile, cluster or cloud computer services for online learning and classification. Any unexpected behavior is analyzed using shared data mining and predictive modeling plugins or exposed to the community at cTuning.org for collaborative explanation. Gradually increasing optimization knowledge helps to continuously improve optimization heuristics of any compiler, predict optimizations for new programs or suggest efficient run-time adaptation strategies depending on end-user requirements. It also allows researchers to quickly reproduce and validate existing results, and focus their effort on novel approaches combined with data mining, classification and predictive modeling. At the same time, it allows conferences and journals to favor publications that can be collaboratively validated by the community.

We initially validated this concept in several academic projects including EU FP6 MILEPOST to build first publicly available machine learning based self-tuning compiler, and later in industry in collaboration with IBM, ARC (Synopsys), CAPS Entreprise, Intel/CEA Exascale Lab, STMicroelectronics and ARM. Since 2007, we started sharing all our past research artifacts including hundreds of codelets, numerical applications, data sets, models, universal experimental pipelines for adaptive optimization space exploration and autotuning, self-tuning machine learning based meta compiler, and unified statistical analysis, classification and predictive modeling plugins in a public Collective Mind repository (c-mind.org/repo). Therefore, we also present and discuss various encountered problems, pitfalls and possible long-term solutions when crowdsourcing autotuning.

3.11 Guiding Tuning with Semi-Analytic Performance Modeling

Torsten Hoefler (ETH Zürich, CH)

Any form of automated tuning often faces a parameter space that is too large to be searched completely. We propose to utilize manual and automated semi-analytic performance modeling techniques to guide the tuner through the search-space. We use those methods to guide an approximate search to limit the space that then has to be considered for a full search. Our techniques can be used for manual as well as automated tuning approaches. We remark that manual tuning is often more powerful since it can operate at the algorithm level. We show several approaches to the problem and how to integrate the modeling into the development and tuning workflow. Our techniques may lead to much more efficient and quickly converging automated tuning.

3.12 Getting More Auto into Autotuning

Jeffrey K. Hollingsworth (University of Maryland – College Park, US)

Joint work of Hollingsworth, Jeffrey K.; Chaen, Ray S.
Main reference J. K. Hollingsworth, "Towards Fully Automatic Auto-tuning: Leveraging language features of Chapel Ray Chen," Int'l Journal of High Performance Computing Applications, 27(4):394–402, Nov. 2013.
URL http://dx.doi.org/10.1177/1094342013493198

Nearly twenty five years ago I started working in performance tuning to make it easier to get programs to run well. Still tuning was tedious, so more than a decade ago I starting working on autotuning to make it easier to get programs to run well. Autotuning made it possible to mechanize the exploration of a space of tuning options. However, the process of identifying those options is still too manual and tedious. In this talk, I will argue that the next big issue in autotuning is making it truly automatic.

3.13 Performance Modeling for Performance Autotuning

Paul D. Hovland (Argonne National Laboratory, US)

We provide some thoughts on performance modeling in the context of automatic performance tuning. We consider analytic models constructed through source code analysis, semi-analytic models constructed with a combination of source code analysis and empirical measurement, and fully empirical models. We imagine several uses for performance models in conjunction with autotuning, including surrogate construction for algorithm evaluation, surrogate-based search, and bounds analysis for search space truncation.

3.14 Automatic Tuning for GPU BLAS kernels

Toshiyuki Imamura (RIKEN – Kobe, JP)

Development of a GPGPU numerical kernel is a typical example of performance autotuning. In this talk, the author will present the development of CUDA BLAS kernels, especially Level 2 kernels, which are bounded by memory bandwidth and have numerous parameter space to be searched. Sieving the number of parameter set and selecting appropriate faster kernel code is significant. Based on a two-stage sieving scheme, we eventually obtained the best-tuned kenrnels, which outperform the state-of-the-art CUDABLAS libraries like CUBLAS and MAGMABLAS. We conclude that better optimization has been carried out by automatic tuning.

3.15 ppOpen-AT: Yet Another Directive-base AT Language

Takahiro Katagiri (University of Tokyo, JP)

Main reference T. Katagiri, S. Ito, S. Ohshima, "Early Experiences for Adaptation of Auto-tuning by ppOpen-AT to an Explicit Method," in Proc. of the IEEE 7th Int'l Symp. on Embedded Multicore SoCs (MCSoC'13), pp. 153–158, IEEE CS, 2013.
URL http://dx.doi.org/10.1109/MCSoC.2013.15
URL http://ppopenhpc.cc.u-tokyo.ac.jp/

Although several Autotuning (AT) languages have been proposed and studied for the last decade, we do not still have crucial languages and tools for AT for supercomputers in operation. In this presentation, we present another AT language based on directive-base language, which is ppOpen-AT[1]. ppOpen-AT is designed with based on basic functions of ABCLibScript[2]. Several new AT functions are developed with taking into account kernels of application software. With respect to supercomputer environments, code generator and AT framework of ppOpen-AT do not require any daemons and script languages for code generation and AT cycle phases.The new developed AT function of ppOpen-AT is loop transformation, such as loop split and loop fusion. This is not new function for loop transformation. However, the loop split function we focus on in ppOpen-AT breaks data flow-dependencies in original loop. In particular, the split requires increase of computations when the loop is split in some examples. Hence, vender compilers cannot supply the function to perform the loop split.To establish the AT with loop split function, we use developers knowledge via directives of ppOpen-AT. As a scenario of AT, we adapt the AT function to a real application based on finite difference method (FDM). By using the Fujitsu PRIMEHPC FX10, that is supercomputer in operation in Information Technology Center, the University of Tokyo, we obtain essential performance improvement by adapting the AT function. We also present preliminary results for the Fujitsu FX10, the Sandy Bridge, and the Xeon Phi.

References

1 T. Katagiri, S. Ito, S. Ohshima. *Early Experiences for Adaptation of Autotuning by ppOpen-AT to an Explicit Method*. Special Session: Auto-Tuning for Multicore and GPU (ATMG) (In Conjunction with the IEEE MCSoC-13), Proceedings of MCSoC-13, 2013.
2 T. Katagiri, K. Kise, H. Honda, T. Yuba. *ABCLibScript: A Directive to Support Specification of an Autotuning Facility for Numerical Software*. Parallel Computing, Vol. 32, Issue 1, 92-112, 2006.

3.16 Energy-Aware HPC – A Tools Perspective

Michael Knobloch (Jülich Supercomputing Centre, DE)

The power consumption and energy-efficiency of supercomputers have been a major topic in High-Performance Computing (HPC) in recent years [1], with the target of 20 MW for an Exascale system. This goal still requires power and energy-efficiency improvements of two orders of magnitude and can only be reached by hardware and software working closely together. Additionally, tuning for power and tuning for energy consumption might require different tools and analyzes.

In this talk I present the past, present and future of the work done at the Jülich Supercomputing Centre (JSC), one of the largest supercomputing centres in Europe, in terms of tools to measure and optimize power and energy consumption of HPC applications. JSC is involved in energy-efficiency related projects since 2009, when awareness for energy-aware HPC started in Germany and the first projects were funded and continues to research ways to reduce the energy-efficiency of its systems.

The eeClust (Energy-Efficient Cluster Computing) project [2, 3], the first project I present, had the goal to analyze resource usage of HPC applications and turn unused components to lower power states when not used [4]. An integral part of this work was extending the existing performance analysis tool-sets Vampir [5] and Scalasca [6] to collect and display power and energy consumption data and determine energy-saving potential within applications [7].

The second project I present is the Exascale Innovation Centre (EIC), a joint collaboration of JSC and IBM research Germany with the goal of co-design the next generation of supercomputers, which has one work package dealing with energy efficiency. Here we investigated the power consumption of IBM machines like Blue Gene/P [8] and POWER7 [9]. We further used these results to generate a model for the CPU power consumption of the POWER7 [10].

Third, I give an outlook to Score-E, a project which will start in October 2013. One major goal of this project is to extend Score-P [11], the new community-driven measurement system which is used by Vampir and Scalasca, to capture and store events related to power and energy consumption. Further, it's planned to enhance the power and energy consumption modelling and prediction as well as to develop new ways of visualization of performance and power data.

I finally conclude with an overview of the lessons we learned during these years and outline the requirements tools pose to future hardware generations in order to be most useful and productive.

References

1 Knobloch, M. *Energy-Aware High Performance Computing – A Survey*. Green and Sustainable Computing: Part II, Academic Press, 2013

2 Minartz, T. and Molka, D. and Knobloch, M. and Krempel, S. and Ludwig, T. and Nagel, W.E. and Mohr, B. and Falter, H. *eeClust: Energy-Efficient Cluster Computing*. Competence in High Performance Computing 2010, (CiHPC-2010), Proceedings of an International Conference on Competence in High Performance Computing, June 2010

3 Knobloch, M. and Minartz, T. and Molka, D. and Krempel, S. and Ludwig, T. and Mohr, B. *eeClust – Energy-efficient Cluster Computing*. Proceedings of the SC11, 2011

4 Minartz, T. and Ludwig, T. and Knobloch, M. and Mohr, B. *Managing hardware power saving modes for high performance computing*. Green Computing Conference and Workshops (IGCC), 2011

5 Knüpfer, A. and Brunst, H. and Doleschal, J. and Jurenz, M. and Lieber, M. and Mickler, H. and Müller, M.. and Nagel, W.E. *The Vampir Performance Analysis Tool-Set*. Tools for High Performance Computing, Springer, 2008

6 Geimer, M. and Wolf, F. and Wylie, B.J.N. and Ábrahám, E. and Becker, D. and Mohr, B. *The Scalasca performance toolset architecture*. Concurrency and Computation: Practice and Experience, Wiley, 2011

7 Knobloch, M. and Mohr, B. and Minartz, T. *Determine energy-saving potential in wait-states of large-scale parallel programs*. Computer science – research and development, Springer, 2012

8 Hennecke, M. and Frings, W. and Homberg, W. and Zitz, A. and Knobloch, M. and Böttiger, H. *Measuring power consumption on IBM Blue Gene/P*. Computer science – research and development, Springer, 2012

9 Knobloch, M. and Foszczynski, M. and Homberg, W. and Pleiter, D. and Böttiger, H. *Mapping fine-grained power measurements to HPC application runtime characteristics on IBM POWER7*. Computer science – research and development, Springer, 2013

10 Gschwandtner, P. and Knobloch, M. and Mohr, B. and Pleiter, D. and Fahringer, T. *Modeling CPU Energy Consumption of HPC Applications on the IBM POWER7*. to appear

11 Knüpfer, A. and Rössel, C. and an Mey, D. and Biersdorff, S. and Diethelm, K. and Eschweiler, D. and Geimer, M. and Gerndt, M. and Lorenz, D. and Malony, A.D. and Nagel, W.E. and Oleynik, Y. and Philippen, P. and Saviankou, P. and Schmidl, D. and Shende, S.S. and Tschüter, R. and Wagner, M. and Wesarg, B. and Wolf, F. *Score-P – A Joint Performance Measurement Run-Time Infrastructure for Periscope, Scalasca, TAU, and Vampir*. Proc. of 5th Parallel Tools Workshop, Springer, 2012

3.17 Potentials and Limitations for Energy Efficiency Auto-Tuning

Andreas Knuepfer (TU Dresden, DE)

With an objective target of 20 MW for an Exascale system, the power budget for future super computers has a severe restriction. Fortunately, current hardware and runtime environments feature various adjustable parameters that can be changed at runtime in order to confine the power usage. We present an overview of the potentials and limitations of several power saving mechanisms and describe the integration of energy efficiency optimizations in existing performance analysis tools. We also introduce an energy efficiency tuning cycle and exemplify its usage.

3.18 Bench-testing Environment for Automated Software Tuning (BEAST)

Jakub Kurzak (University of Tennessee, US)

The goal of BEAST is to create a framework for exploring and optimizing the performance of computational kernels on hybrid processors that 1) applies to a diverse range of computational kernels, 2) (semi)automatically generates better performing implementations on various hybrid processor architectures, and 3) increases developer insight into why given kernel/processor combinations have the performance profiles they do. We call this form of optimization "bench-tuning" because it builds on the model used for traditional benchmarking by combining an abstract kernel specification and corresponding verification test with automated testing and data analysis tools to achieve this threefold goal.

3.19 Knowledge-based Performance Engineering and Empirical Autotuning

Allen D. Malony (University of Oregon, US)

In the last 20 years, the parallel performance community has delivered powerful techniques and tools for measurement and analysis of parallel systems and applications. While these research and development advances have been important for observing how the evolving features of parallel machines affect application performance, there has been less done by the parallel computing community as a whole to carry forward performance knowledge from technology generation to generation.

Parallel computing is now at a point where it is imperative to incorporate knowledge at all levels in order to engineer optimized applications targeted to the essential features of the environment in which they will run. This talk will discuss the role of empirical autotuning as a process for systematic experimentation to create a knowledge base of performance information that can be applied in support of automatic performance engineering and tuning objectives. Results will be presented on the developement of an empirical autotuning framework combining a parallel performance analysis system with a code transformation tool and autotuner environment.

3.20 Dynamic Tuning for Large-Scale Computing using ELASTIC

Andrea Martinez (Autonomus University of Barcelona, ES)

Dynamic tuning is the most viable tactic to improve the performance of parallel applications which present long runnings times or behavioural patterns that change depending on the input data set or according to data evolution. In order to bring this strategy to large scale computers, we propose a model that enables scalable dynamic tuning. This model is based on the application decomposition combined with an abstraction mechanism to solve local and global problems. The proposed model has been implemented in the form of ELASTIC, a tool for large- scale dynamic tuning. Using ELASTIC, an experimental evaluation has been performed over a synthetic parallel application (up to 16384 tasks) and an agent- based real parallel application (up to 2048 tasks). The results demonstrate that the proposed model, embodied in ELASTIC, is able to manage the increasing number of processes, allowing for the effective tuning of large-scale parallel applications.

References

1 Andrea Martínez and Anna Sikora and Eduardo César and Joan Sorribes. *How to Scale Dynamic Tuning to Large-Scale Applications*. Proceedings of International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS) – IPDPS, 2013.

2 Andrea Martínez and Anna Sikora and Eduardo César and Joan Sorribes. *How to Determine the Topology of Hierarchical Tuning Networks for Dynamic Auto- Tuning in Large-Scale Systems*. Proceedings of International Workshop on Automatic Performance Tuning (IWAPT) – ICCS, 2013.

3.21 Opportunities and Strategies for I/O Auto-Tuning

Renato Miceli Costa Ribeiro (ICHEC – Galway, IR)

In the HPC community, I/O issues are both one of the most common bottleneck and scalability limiting factors in codes, and one scientists and developers alike are the least aware of. For this, automating the I/O tuning is likely to lead to significant performance improvements, especially since the corresponding manual tuning is a complex task often out of reach of users and code developers. In this talk we will discuss the opportunities and possible approaches to automatically tuning the I/O of HPC software. We will introduce some of the most common I/O issues one can encounter while developing or using a HPC code, examine their relationship to the machine hardware, OS and filesystem settings, and explore prospective automatic tuning strategies adapted to address each one of these issues.

3.22 Parallel Performance Measurement and Analysis on Extreme-Scale Computer Systems

Bernd Mohr (Jülich Supercomputing Centre, DE)

The number of processor cores available in high-performance computing systems is steadily increasing. In the June 2013 list of the TOP500 supercomputers, only three systems have less than 4,096 processor cores and the average is almost 39,000 cores, which is an increase of 9,000 in just one year. Even the median system size is already over 16,000 cores. While these machines promise ever more compute power and memory capacity to tackle today's complex simulation problems, they force application developers to greatly enhance the scalability of their codes to be able to exploit it. To better support them in their porting and tuning process, many parallel tools research groups have already started to work on scaling their methods, techniques and tools to extreme processor counts. In this talk, we survey existing performance analysis and optimization tools covering both profiling and tracing techniques, report on our experience in using them in extreme scaling environments, review existing working and promising new methods and techniques, and discuss strategies for addressing unsolved issues and problems. Important performance tool sets covered include TAU [1], HPCToolkit [2], Paraver [3], Vampir [4], Scalasca [5], Open|SpeedShop [6], and Periscope [7].

References

1 TAU, University of Oregon, US, http://tau.uoregon.edu.
2 HPCToolkit, Rice University, US, http://hpctoolkit.org.
3 Extrae/Paraver, BSC, Spain, http://www.bsc.es/paraver.
4 Vampir, TU Dresden, Germany, http://www.vampir.eu.
5 Scalasca, JSC, Germany, http://www.scalasca.org.
6 Open|SpeedShop, Krell Institute, US, http://www.openspeedshop.org.
7 Periscope, TU Munich, Germany, http://www.lrr.in.tum.de/~periscop/.

3.23 Restructuring Legacy Codes to Enable Autotuning

Shirley V. Moore (University of Texas – El Paso, US)

Although autotuning has been applied successfully in the domain of linear algebra software, its use with general application codes, especially parallel codes, has had limited success. In many cases, the parallel implementation obscures the inherent parallelism and introduces false data dependencies and unnecessary synchronization that can be hard to detect and eliminate. In this talk, we discuss ways of restructuring legacy codes to increase opportunities for autotuning and give examples from the domain of computational chemistry.

3.24 Autotuning the Energy Consumption

Carmen Navarrete (Leibniz Rechenzentrum – München, DE)

Joint work of Navarrete, Carmen; Guillen, Carla; Hesse, Wolfram; Brehm, Matthias

Saving energy in high performance systems is becoming increasingly important as systems become larger and energy costs rise. Although modern processors with Dynamic Voltage and Frequency Scaling (DVFS) already automatically scale the frequency depending on the load of the processor, they do not detect the effects of this scaling in a parallel application as a whole. Manual efforts to tune the processor frequency of a parallel application to obtain the best energy to solution can be extremely time consuming. Thus it is important to have analysis tools which help measure, analyse and propose solutions to the user. The Periscope Tuning Framework, henceforth called PTF, has the capabilities to automatically search for optimizations in a code. This tool can be used to define a structured analysis mechanism which will automatically search for an optimal solution. This mechanism can be coded as a plugin and can be easily integrated into the PTF. We present a plugin for the PTF which will enable the detailed analysis of a parallel code, i.e. per code region and task. The tool will automatically propose processor frequencies at the level of regions and tasks which optimize the energy to solution of a parallel application as a whole.

References

1 Miceli et al. *AutoTune: A Plugin-Driven Approach to the Automatic Tuning of Parallel Applications*. In Applied Parallel and Scientific Computing, Vol. 7782 (2013), pp. 328–342, Springer Berlin Heidelberg, 2013.
2 Tiwari et al. *Auto-tuning for energy usage in scientific applications*. Proceedings of the 2011 international conference on Parallel Processing – Euro-Par'11, Volume 2, pp. 178–187, Springer-Verlag, Berlin, Heidelberg, Germany (2012)
3 Rountree et al. *Beyond DVFS: A First Look at Performance under a Hardware-Enforced Power Bound*, IPDPS Workshops, pp.947-953, IEEE Computer Society, (2012)
4 Wittman et al. *An Analysis of energy-optmized lattice-Boltzmann CFD simulations from the chip to the highly parallel level*, CoRR, abs/1304.7664 (2013)
5 Weaver V, Johnson M, Kasichayanula K, Ralph J, Luszczek P, Terpstra D, Moore S. *Measuring Energy and Power with PAPI* (2012), doi: 10.1109/ICPPW.2012.39

6 Browne S, Deane C, Ho G, Mucci P. In Proceedings of the Department of Defense HPCMP Users Group Conference, *PAPI: A Portable Interface to Hardware Performance Counters* (1999), pp. 7–10.
7 David H, Gorbato E, Hanebutte U, Khanna R, Le C. Low-Power Electronics and Design (ISLPED), *RAPL: memory power estimation and capping* (2010), pp. 189–194.
8 Intel Corp. 2012. *Intel 64 and IA-32 Architectures Software Developer Manual.*
9 *CPU Frequency Scaling.* Webpage: https://wiki.archlinux.org/index.php/CPU_Frequency_Scaling.
10 SuperMUC https://www.lrz.de/services/compute/supermuc
11 Intel Corp. 2012. Intel Xeon Processor. http://www.intel.com/xeon
12 Treibig J, Hager G, Wellein G. *LIKWID: A lightweight performance-oriented tool suite for x86 multicore environments*, Proceedings of the 39th International Conference on Parallel Processing Workshops (2010), pp. 207–216, IEEE Computer Society, Washington, DC, USA (2010)
13 Treibig J, Hager G, Wellein G. *Best practices for HPM-assisted performance engineering on modern multicore processors*, Proceedings of the CoRR (2012)
14 Strohmaier E, Shan H. *Architecture independent performance characterization and benchmarking for scientific applications.* Proceedings of the The IEEE Computer Society's 12th Annual International Symposium on Modeling, (2004), pp.467–474, IEEE Computer Society, Washington, DC, USA. (2004)
15 Strohmaier E, Shan H. *Apex-Map: A global data access benchmark to analyze HPC systems and parallel programming paradigms*, Proceedings of the SC2005 (2005). pp. 12–18
16 HWMON https://www.kernel.org/doc/Documentation/hwmon/sysfs-interfaces

3.25 High performance program generators: systematic construction with language support

Georg Ofenbeck (ETH Zürich, CH)

Main reference G. Ofenbeck, T. Rompf, A. Stojanov, M.n Odersky, M. Püschel, "Spiral in Scala: Towards the Systematic Construction of Generators for Performance Libraries," in Proc. of the Int'l Conf. on Generative Programming: Concepts and Experiences (GPCE'13), pp. 125–134, ACM, 2013.
URL http://dx.doi.org/10.1145/2517208.2517228

Program generators for high performance libraries are an appealing solution to the recurring problem of porting and optimizing code with every new processor generation, but only few such generators exist to date. This is due to not only the difficulty of the design, but also of the actual implementation, which often results in an ad-hoc collection of standalone programs and scripts that are hard to extend, maintain, or reuse. In our recent work we asked the question which programming language concepts and features are needed to enable a more systematic construction of such generators.

This talk will provide a teaser on some of the solutions we investigated.

3.26 Methodology for MPI Applications Autotuning

Antonio Pimenta (Autonomus University of Barcelona, SP)

We present a methodology designed to tackle the most common problems of MPI parallel programs. By developing a methodology that applies simple steps in a systematic way, we expect to obtain the basis for a successful autotuning approach of MPI applications based on measurements taken from their own execution. As part of the AutoTune project, our work is ultimately aimed at extending Periscope to apply automatic tuning to parallel applications and thus provide a straightforward way of tuning MPI parallel codes. Experimental tests demonstrate that this methodology could lead to significant performance improvements.

3.27 Exploiting processor inhomogeneity for performance optimization under a power bound

Barry Roundtree (LLNL – Livermore, US)

Cluster heterogeneity has traditionally referred to the presence of multiple different kinds of processors. However, significant variation in efficiency exists within individual processors families, with 10% variation observed in current processors and up to 25% expected in upcoming generations. While it is possible to mask these differences with intelligent power scheduling, the use of autotuning for performance optimization under a power bound will require exploiting these differences. In order to do so, however, we will have to rethink how jobs are scheduled, how MPI ranks (or equivalent) are mapped to processors, runtime critical path discovery as well as how supercomputers are designed and built.

3.28 Tuning support for a Xeon Phi template library

Martin Sandrieser (Universität Wien, AT)

Joint work of Bajrovic, Enes; Benkner, Siegfried; Dokulil, Jiri; Sandrieser, Martin

We developed a template library that provides TBB-like parallel patterns that are executed in a hybrid fashion on the host and one or more Xeon Phi coprocessors simultaneously. The performance achievable with this library is profoundly influenced by the distribution of work among host processors and the coprocessors. Work distribution is performed automatically at runtime depending on different tunable parameters. These parameters can be reconfigured during runtime. In this talk we outline the runtime support implemented for our library, the initial support for autotuning, and discuss possible integration with existing autotuning frameworks.

References

1 J. Dokulil, E. Bajrovic, S. Benkner, M. Sandrieser, and B. Bachmayer. *HyPHI – task based hybrid execution C++ library for the Intel Xeon Phi coprocessor.* In 42nd International Conference on Parallel Processing (ICPP-2013), Lyon, France, 2013.

3.29 Performance Analysis, Energy Efficiency Optimizations and Auto-Tuning

Robert Schoene (TU Dresden, DE)

With an objective target of 20 MW for an Exascale system, the power budget for future super computers has a severe restriction. Fortunately, current hardware and runtime environments feature various adjustable parameters that can be changed at runtime in order to confine the power usage. We present an overview of the potentials and limitations of several power saving mechanisms and describe the integration of energy efficiency optimizations in existing performance analysis tools. We also introduce an energy efficiency tuning cycle and exemplify its usage.

3.30 Providing the Necessary Semantic Context for Performance Tools and Autotuners

Martin Schulz (LLNL – Livermore, US)

Performance tools and auto tuning systems alike require or can benefit from semantic context information as this enables a direct correlation and attribution of performance observations to application properties. In this talk I will show a few case studies in which we used such information to aid in the analysis of performance data. Further, I will introduce a system that can be used to create the necessary APIs for application programers to safely express such semantic information.

3.31 Smarter Libraries with Synthesis

Armando Solar-Lezama (MIT, US)

Performance tools and auto tuning systems alike require or can benefit from semantic context information as this enables a direct correlation and attribution of performance observations to application properties. In this talk I will show a few case studies in which we used such information to aid in the analysis of performance data. Further, I will introduce a system that can be used to create the necessary APIs for application programers to safely express such semantic information.

3.32 Application-independent Autotuning for GPUs

Walter F. Tichy (KIT – Karlsruhe Institute of Technology, DE)

We investigate the potential of online autotuning for general purpose computation on GPUs. Our application-independent autotuner AtuneRT optimizes GPU-specific parameters such as block size and loop-unrolling degree. We also discuss the peculiarities of auto- tuning on GPUs. We demonstrate tuning potential using CUDA and by instrumenting the parallel algorithms library Thrust. We evaluate our online autotuning approach on three GPUs with four applications.

3.33 Generalized roofline analysis?

Richard Vuduc (Georgia Institute of Technology – Atlanta, US)

Joint work of Choi, Jee; Vuduc, Richard
Main reference J. W. Choi, D. Bedard, R. Fowler, R. Vuduc. "A roofline model of energy," in Proc. of the IEEE 27th Int'l Symp. on Parallel & Distributed Processing (IPDPS'13), pp. 661-672, IEEE CS, 2013.
URL http://dx.doi.org/10.1109/IPDPS.2013.77

The roofline model of Williams, Waterman, and Patterson (2009) is an analytic tool for gaining insights into how performance changes as intrinsic properties of a computation interact with features of a hardware system. We are generalizing the original roofline to develop a first-principles understanding the relationships among computational time, energy, and power. Our initial suggests how we might use these to explore the landscape of systems and algorithms that might be possible in the future.

References

1 Jee Whan Choi and Daniel Bedard and Robert Fowler and Richard Vuduc. "A roofline model of energy." In *Proceedings of the IEEE Parallel and Distributed Processing Symposium* (IPDPS), Boston, MA, USA, May 2013. DOI: 10.1109/IPDPS.2013.77
2 Kenneth Czechowski and Richard Vuduc. "A theoretical framework for algorithm-architecture co-design." In *Proceedings of the IEEE Parallel and Distributed Processing Symposium* (IPDPS), Boston, MA, USA, May 2013. DOI: 10.1109/IPDPS.2013.99

3.34 Autotuning for Scale

Felix Wolf (German Resarch School for Simulation Sciences, DE)

Joint work of Calotoiu, Alexandru; Hoefler, Torsten; Poke, Marius; Felix Wolf
Main reference A. Calotoiu, T. Hoefler, M. Poke, F. Wolf, "Using Automated Performance Modeling to Find Scalability Bugs in Complex Codes," in Proc. of the Int'l Conf. for High Performance Computing, Networking, Storage and Analysis (SC'13), pp. 45:1–45:12, ACM, 2013.
URL http://dx.doi.org/10.1145/2503210.2503277

In this talk, we present a method designed to detect scalability bugs based on automatically generated performance models and discuss how it can be used to autotune for scale. Applications include the comparison of different algorithmic choices or execution configurations with respect to scalability. Since our method requires only small-scale experiments to make predictions for larger scales, the cost of this approach is relatively low even if the number of variants to be compared is high.

4 Working Groups

4.1 Energy and power autotuning

This breakout session investigated goals and techniques for energy and power optimization. While energy consumption reduction was only shortly discussed in the group of about 12 people, the main focus was on power issues in future exascale computers.

LLNL estimates that their first exascale system will be able to run with 60 MW but only 20 MW will be available to the machine. A first power budget limited machine is already under procurement at LLNL.

The contract with the energy company will be optimized for a steady use of the power. While it will have an upper limit of 20 MW, lower energy consumption will lead to higher costs as well. The motivation behind this is to enable the power company to optimize for a steady flow of energy. In addition, sudden drops in power usage will also not be possible to protect the power supply infrastructure from damage. These constraints require careful management of the power budget.

While a limitation of the power used by an application can be achieved by extended methods for power capping going beyond the processor's RAPL support. The second issue to not suddenly drop the machine's power consumption requires some additional mechanism, such as careful management by the job scheduler. In addition scenarios like trading power among the processes of an application to speedup those on the critical path as well as between different applications might gain importance on such machines.

The discussion on energy reduction concentrated on collecting the already applied techniques. Many projects explored the use of DVFS to reduce the energy by allowing to slow down the application for a limited percentage of the execution time. Instead of influencing the frequency, the same can be achieved by using the processor's power capping mechanism and letting the processor figure out how far to reduce the frequency.

Another approach is to trade performance for energy savings is to apply concurrency throttling. While selecting more efficient parallel execution with less threads or processes an energy reduction can be achieved.

A third approach is to slow down fast processors in case of load imbalances. Instead of a race to idle, these processors will arrive just in time while consuming less energy.

Besides reduction of the frequency and the voltage, switching off components might give significant power reduction. For example, the vector units of the processors are quite power hungry and should be switched off if not used efficiently.

In general, the group agreed that any performance tuning techniques are good for reduction of the energy, since faster and more efficient computation most frequently reduces the energy due to the constant static power used in the processors.

4.2 Language support for Autotuning

This breakout session discussed the issues of language support for autotuning. There was quite a bit of discussion about what it means to have language support for autotuning. Eventually we agreed there are two rather different approaches to languages for autotuning. One approach is to provide extensions to current languages to express tuning options in the form of parameters and algorithmic choices. The second approach is to develop new high level languages that support expressing the program to be performed in more abstract ways to allow an autotuner to generate many different program variants. The relative advantages of the two approaches are:

Extensions: the extension approach allows simpler integration with existing languages and would permit software to be evolved to be autotuned over time. These extensions could be developed in the form of pragmas that would be automatically ignored by compilers that didn't support autotuning or the specific pragma. Support could also be included in the extensions to describe hints about how the various parameters and pragmas interact. This could include both constraints such as one parameter must be less than the another to hints about the relative importance of different options.

New Languages: the new languages approach would allow for more aggressive autotuning. Programs would be written in a very high level language more like Matlab, and then compiled and autotuned to machine code. Machine specific descriptions of the transformations and the autotuning options could be written by people other than the application developers and potential reused for multiple programs.

4.3 Infrastructures

The breakout session about infrastructures consisted mainly of discussions related to what performance analysis and measurement tools can provide to automatic tuners. The consensus was that automatic tuners are stand-alone components that exist outside of the tools (with the possible exception of Periscope). What can be provided by tools (as well as applications) are operations that split logically into information and control related.

Information interfaces are already offered by several tools. In many cases, runtime environments also offer information interfaces. With the release of the MPI-3 standard, many MPI implementations give information about communication performance, message queue states, protocol thresholds, among others. Most tool developers seem to agree that exposing information that can be useful for automatic tuners is not only desirable, but also does not require a lot of development time.

In contrast with their posture regarding information interfaces, not all groups agreed on the feasibility or usefulness of control interfaces. The debate centered around the required scope of the automatic tuners. For example, some argued that the tuner should not touch internal parameters of communication libraries, such as MPI, while others saw the potential performance benefit achievable as worth pursuing. MPI does allow implementors to allow control access to internal parameters through MPIT, if they so desire.

A common issue that was discussed is that, because tools work at different abstraction levels and target hardware depending on what they measure or analyze, developing a common set of APIs for information and control will be difficult. Having a common set of APIs would allow an automatic tuner to interoperate with multiple tools in a modular manner.

Experienced members in the discussion shared that the tools community has not managed to come up with standard APIs and data structures even after many years of cooperation.

It was generally agreed that online access APIs to information and control APIs can greatly benefit automatic tuners. The need for restarts increases the time required for tuning significantly. Together with online approaches, modeling can be used to reduce the times required while optimizing. It was also agreed that, depending on the frequency of samples and scale of applications, the balance between flexibility and processing time required for these online interfaces may differ.

4.4 Black/Whitebox Autotuning

In this breakout session we discussed various approaches to autotuning that we classified as black box and white box.

Black Box: In black box approaches the autotuning search machinery does not have knowledge of the search space and thus cannot employ any knowledge about the underlying problem, except for knowledge obtained through evaluating instance. Notable black box successes include ActiveHarmony+Chill and profile guided optimization in compilers like the Intel compiler and the IBM XL compiler.

White Box: In white box approaches the search machinery has some understanding of the underlying problem structure and thus the search can be guided. Notable successes in white box autotuning include ATLAS and Spiral.

Generally, black box approaches are wider applicable as they do not need to understand the underlying problem structure, while white box approaches are limited to well-understood kernels or libraries that are easier to model.

Other aspects that separate the approaches are the cost to evaluate search points (recompile versus relaunch), model quality and constraints in guided search. Model types and outputs are covering a wide range of possibilities and models may not be very accurate by nature. Types of models may vary, they may include target architecture, operating systems, and data sets or even the semantics of the tuned program. A model may be predictive or just provide relative ordering of instances. The working group felt that a "trust but verify" approach to models is needed in autotuning.

4.5 Search Algorithms

During this breakout session the group brought together an number of different search algorithms that are used in autotuning systems. Autotuning starts from a search space that is constructed from valid setting for the tuning parameters. The following search algorithms to walk the search space are available:

Exhaustive: The full search space is explored and all the combinations of the tuning parameters are evaluated.

Random: This algorithm selects valid points in the search space randomly. Random and exhaustive are good for autotuning if nothing is known about the structure of the objective function.

Meta-heuristics: This class covers genetic algorithms and simulated annealing.

Derivative-free algorithms: To this class belong algorithms such as parallel rank order, nelder mead, hill climbing, and tabu search.

Machine learning: Several techniques from machine learning can be applied to improve autotuning, such as neural networks, decision trees, and support vector machines. All these techniques try to learn a correlation between an application signature and the tuning parameters or the application performance. They can be applied to predict the application performance for certain parameter settings or as an oracle to predict the best or a best set of parameter settings. An important program with machine learning techniques is to find the right set of application properties that discriminate the applications.

Rule-based: Another way to shrink the search space is to apply rules that let the autotuner deduce a certain subspace.

Besides the collection of search algorithms, the group discussed the aspect of generating valid points in the search space when a number of constraints are given. Several tools have used Presburger Arithmetic to solve this problem.

In important aspect in autotuning is also the tradeoff between quality of the results and the effort spent in searching for the best solution. Some existing tools provide a search time limit and return the found best solution when the limit is reached. This is especially used for meta-heuristics and random search.

Besides using just one search algorithms, multiple algorithms can be combined to solve the problem. A good example is to use machine learning to get a good estimate and perform a local search afterwards. Also a random selection of starting points with a local search might be useful.

At the end, the group also discussed how the quality of search algorithms can be evaluated. A comparison with exhaustive search is typically due to the size of the search space impossible. One approach is to replace exhaustive with random or to compare different algorithms by selecting the same number of tries for the algorithms.

Participants

- Enes Bajrovic
Universität Wien, AT
- Shajulin Benedict
St.Xavier's Catholic College of Engineering, IN
- Siegfried Benkner
Universität Wien, AT
- Aydın Buluç
Lawrence Berkeley National Laboratory, US
- Milind Chabbi
Rice University, US
- I-hsin Chung
IBM TJ Watson Res. Center – Yorktown Heights, US
- Isaias Alberto Compres Urena
TU München, DE
- Guojing Cong
IBM TJ Watson Res. Center – Yorktown Heights, US
- Thomas Fahringer
Universität Innsbruck, AT
- Franz Franchetti
Carnegie Mellon University, US
- Grigori Fursin
INRIA Saclay – Île-de-France – Orsay, FR
- Hans Michael Gerndt
TU München, DE
- Carla Guillen
Leibniz Rechenzentrum – München, DE
- Torsten Höfler
ETH Zürich, CH
- Jeffrey K. Hollingsworth
University of Maryland – College Park, US
- Paul D. Hovland
Argonne National Laboratory, US
- Toshiyuki Imamura
RIKEN – Kobe, JP
- Thomas Karcher
KIT – Karlsruhe Institute of Technology, DE
- Takahiro Katagiri
University of Tokyo, JP
- Michael Knobloch
Jülich Supercomputing Centre, DE
- Andreas Knüpfer
TU Dresden, DE
- Jakub Kurzak
University of Tennessee, US
- Allen D. Malony
University of Oregon, US
- Andrea Martinez
Autonomus University of Barcelona, ES
- Renato Miceli Costa Ribeiro
ICHEC – Galway, IE
- Robert Mijakovic
TU München, DE
- Bernd Mohr
Jülich Supercomputing Centre, DE
- Shirley V. Moore
University of Texas – El Paso, US
- Carmen Novarrete
Leibniz Rechenzentrum – München, DE
- Georg Ofenbeck
ETH Zürich, CH
- Antonio Pimenta
Autonomus University of Barcelona, ES
- Barry Rountree
LLNL – Livermore, US
- Martin Sandrieser
Universität Wien, AT
- Robert Schoene
TU Dresden, DE
- Martin Schulz
LLNL – Livermore, US
- Armando Solar-Lezama
MIT, US
- Walter F. Tichy
KIT – Karlsruhe Institute of Technology, DE
- Jesper Larsson Träff
TU Wien, AT
- Richard M. Veras
Carnegie Mellon University, US
- Richard Vuduc
Georgia Institute of Technology – Atlanta, US
- Felix Wolf
German Resarch School for Simulation Sciences, DE

Report from Dagstuhl Seminar 13402

Physical-Cyber-Social Computing

Edited by

Amit P. Sheth[1], Payam Barnaghi[2], Markus Strohmaier[3], Ramesh Jain[4], and Steffen Staab[5]

1 **Wright State University – Dayton, US,** amit@knoesis.org
2 **University of Surrey – Guildford, GB,** p.barnaghi@surrey.ac.uk
3 **Universität Koblenz-Landau, DE,** strohmaier@uni-koblenz.de
4 **Univ. California – Irvine, US,** jain@ics.uci.edu
5 **Universität Koblenz-Landau, DE,** staab@uni-koblenz.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13402 “Physical-Cyber-Social Computing”.

Seminar 29. September – 04. October, 2013 – www.dagstuhl.de/13402

1998 ACM Subject Classification C.2 Computer-communication Networks, K.4 Computers and Society, B. Hardware

Keywords and phrases Semantic sensor networks, Web of Things, Internet of Things, Semantic interoperability and context, Social computing, Collective intelligence, Situation-awareness, Human and machine perception, Computing for human experience, Social life networks

Digital Object Identifier 10.4230/DagRep.3.9.245

Edited in cooperation with Rajendra Akerkar (Vestlandsforsking, NO), Josiane Xavier Parreira (National University of Ireland, IE), Pramod Anantharam (Wright State University, US)

1 Executive Summary

Amit P. Sheth
Payam Barnaghi
Markus Strohmaier
Ramesh Jain
Steffen Staab

Miniaturisation, progress with energy issues and cost reductions have resulted in rapid growth in deployment of networked devices and sensing, tightly connecting the physical word with the cyber-world as well as interconnected humans bringing along them virtual social interactions.. The number of devices connected to the Internet already exceeds the number of people on earth and is estimated to grow to 50 billion devices by 2020. The resulting system called Internet of Things (IoT) incorporates a number of technologies including wireless sensor networks, pervasive computing, ambient intelligence, distributed systems and context-aware computing. With growing adoption of smart-phones and social media, citizens or human-in-the-loop sensing and resulting user generated data and data generated by user carried devices have also become key sources of data and information about the physical world and corresponding events. Data from all these sources will result in tremendous volume, large variety and rapid changes (velocity). The combination of cyber-physical and social data can help us to understand events and changes in our surrounding environments better, monitor and control buildings, homes and city infrastructures, provide better healthcare and elderly care services among many other applications. To make efficient use of the physical-cyber-social data, integration and processing of data from various heterogeneous sources is necessary. Providing interoperable information representation and extracting actionable knowledge from deluge of human and machine sensory data are the key issues. We refer to the new computing capabilities needed to exploit all these types of data to enable advanced applications as physical-cyber-social computing.

2 Table of Contents

3 Overview of Talks

3.1 From smart meters to smart behaviour

Harith Alani (The Open University – Milton Keynes, GB)

It is becoming clear to policy makers and environment agencies that changing the behaviour of energy consumer is vital for battling climate change. Much work has being done recently to support behaviour change in this domain, for example with smart meters, refined energy consumption reports, and information campaigns. However, many questions remain unanswered, such as which combination of incentives work best, and for which scenarios and demographic areas, which behaviour change strategies are short/long term, and how much behavioural change can be expect from which strategy. Answering these important and complex questions requires the close collaboration of computer scientists, sociologists, energy meter producers, and environment campaigners, to experiment with, and evaluate, the impact of different approaches in changing the behaviour of energy consumers in households and businesses.

3.2 Beyond Factual Question Answering

Pramod Anantharam (Wright State University – Dayton, US)

Question Answering (QA) systems have been around for decades but the most notable breakthrough in QA systems was achieved by IBM Watson, the machine that defeated Ken Jennings, a champion who had won Jeopardy! 72 times in a row. The Jeopardy! game was broadcasted on national television in February 2012. Such QA systems will alleviate the challenge of decision making in data intensive environments. This type of question answering is called factual question answering as the answers are synthesized by facts in textual documents. While this has been the state-of-the-art in QA systems, Physical-Cyber-Social systems offer a unique set of challenges and this short presentation proposes a provocative idea of building a cognitive system that goes beyond factual QA. By presenting an asthma example, the challenge of combining observations from multimodal and multi-sensory sources is demonstrated.

3.3 Social networks across the digital-physical boundary

Ciro Cattuto (ISI Foundation, IT)

The advances in mobile technologies and wearable sensors allow to quantify human behavior at unprecedented levels of scale and detail. Wearable sensors, in particular, are opening up a new window on social behavior at the finest resolution of individual interactions, impacting diverse research areas such as social network analysis, organizational science and infectious disease dynamics. I will summarize recent efforts on measuring and analyzing social networks from spatial behavior, and highlight important structural properties of the empirical data collected in real-world environments. I will discuss specific challenges in data cleaning, curation and integration of social network data from sensors. Finally, I will reflect on a few outstanding challenges in using these data sources to achieve impact in specific domains such as infectious disease dynamics.

3.4 A few thoughts on engineering social-computational systems

Markus Strohmaier (Universität Koblenz-Landau, DE)

Social computational systems are integrated systems of people and computers. What distinguishes social computational systems from other types of software systems – such as software for cars or air planes – is the unprecedented involvement of data about user behavior, -goals and -motivations into the software system's structure. In social computational systems, the interaction between a user and the system is mediated by the aggregation of explicit or implicit data from other users. This is the case with systems where, for example, user data is used to suggest search terms (e.g. Google Autosuggest), to recommend products (e.g. Amazon recommendations), to aid navigation (e.g. tag-based navigation) or to filter content (e.g. Digg.com). This makes social computational systems a novel class of software systems (as opposed to for example safety-related software that is being used in cars) and unique in a sense that potentially essential system properties and functions – such as navigability – are dynamically in uenced by aggregate user behavior. Such properties can not be satisfi ed through the implementation of requirements alone, what is needed is regulation, i.e. a dynamic integration of users' goals and behavior into the continuous process of engineering.

3.5 Citizen Actuation For Lightweight Energy Management

Edward Curry (National University of Ireland – Galway, IE)

Cyber-Physical Energy Systems (CPES) exploit the potential of information technology to boost energy efficiency while minimizing environmental impacts. CPES can help manage energy more efficiently by providing a functional view of the entire energy system so that energy activities can be understood, changed, and reinvented to better support sustainable practices. CPES can be applied at different scales from Smart Grids and Smart Cities to Smart Enterprises and Smart Buildings. Significant technical challenges exist in terms of information management, leveraging real-time sensor data, coordination of the various stakeholders to optimize energy usage.

In this talk I describe an approach to overcome these challenges by re-using the Web standards to quickly connect the required systems within a CPES. The resulting lightweight architecture leverages Web technologies including Linked Data, the Web of Things, and Social Media. The talk describes the fundamentals of the approach and demonstrates it within an Enterprise Energy Management scenario smart building.

3.6 Principles of Elastic Systems Towards building Cyber-Physical-Social Systems

Schahram Dustdar (TU Wien, AT)

In this talk I present the fundamental models, algorithms and engineering methods and programming abstractions which enable software engineers to model, design, and execute a novel class of software systems: Elastic Systems on a Cloud computing infrastructure. These systems are compositions of the Internet of Things, People, and Software services (including compute, storage, network units).

3.7 Human-Machine Cooperation in Research

Michael Granitzer (Universität Passau, DE)

While Machines process enormous processing capabilities, the lack Humans creativity, intuition and common sense background knowledge. A circumstance unlikely to be changed within the next few years, if ever. In this talk i aim to briefly highlight recent developments in machine learning and outline the need for a tighter integration of machines and humans for upcoming data challenges. Particular emphasis will be placed on data challenges in research and the Giant Global Graph as all encompassing database.

3.8 Understanding and shaping human behavior

Vivek K. Singh (MIT Media Lab, US)

With the growth trends in sensing and information sharing, we can soon expect personal behavioral data (e.g. calls, Bluetooth, heart rate, other quantified-self sensing) to become readily accessible via the ubiquitous Internet-of-Everything. This will allow computational systems to go beyond cyber trails/partial reports of people's actions (e.g. tweets, surveys, or yearly medical checkups) and actually work with real-world signals, coming in real-time, from the real world. Analyzing and utilizing this data is important because: 1) it can answer epistemological questions on human behavior in a data-driven manner, and 2) provide actionable guidelines on how to persuade humans in real world social scenarios. This talk gives a short review of the recent approaches at understanding and shaping human behavior using such data.

3.9 Big Money and NSA^2 – The Future for Physical-Cyber-Social Computing?

Manfred Hauswirth (National University of Ireland – Galway, IE)

It is an established fact that we produce enormous amounts of static and dynamic information. This information is exploited to a certain extent already. Research focused on making this information accessible in a simple fashion (infrastructures), ways of scalable data integration in open environments (Linked Data, ontologies) and putting data to use (analytics, smart cities, etc.). A lot of business opportunities are predicted in this area. However, the systems are not mature enough yet and a lot of research is still required. Additionally, if successful, the flipside of this success will be that we may completely lose any privacy as we can then be monitored comprehensively in the real and in the online worlds. The question is how we can do good research while not making the job of the NSA even easier as it is already. Everyone talks about privacy and actually offer and apply it. In this talk, I will quickly review the existing state of the art, some of the driving requirements (in my opinion) and issues the research community must turn their attention to (again my personal opinion).

3.10 Social and Sensor Information – Two Views on the World

Andreas Hotho (Universität Würzburg, DE)

In the last decade the social web emerged and had a strong influence on everyone's daily live. Today, most of the newly bought mobile phone are smartphones which have a bunch of additional sensors on board. Using this novel combination of sensor information and opinions of users uttered in the social web will lead to a new level of information quality. This talk will discuss this emergent new area along examples from the EveryAware project. We will use the results to illustrate and explain the future changes and challenges.

3.11 Weaving the Social Web into User Modeling and Adaptation

Geert-Jan Houben (TU Delft, NL)

The social web is having a clear impact in the field of user modeling and adaptation. On the social web a large source of data s generated by users themselves, often for different purposes, and that provides an unprecedented potential for systems to understand their users and to adapt based on that understanding. As we can see from researchers and projects in a number of relevant fields, data on various manifestations of what users do socially on the web brings new opportunities. Exciting ideas are generated and first explorations show promising results. In this talk we aim to understand the impact on methods and techniques for user modeling and adaptation. We also look forward by identifying challenges that can drive our research.

3.12 Smart Social Systems

Ramesh Jain (University of California – Irvine, US)

Availability of enormous volumes of heterogeneous Cyber-Physical-Social (CPS) data streams may allow design and implementation of networks to connect various data sources to detect situations with little latency. In fact, in many cases it may even be possible to predict situations well in advance. This opens up new opportunities in designing smart social systems for specific tasks. Such systems may be very useful for many important problems at local as well as regional and even global level. We believe that such systems offer many novel challenges to researchers in multimedia, particularly in social and cross-modal media systems. We will present our ideas and challenges derived from our early experience towards building smart social systems.

3.13 Future of End-User Configuration of IoT : "Do It Yourself" or (only) "Choose It Yourself"?

Artem Katasonov (VTT – Espoo, FI)

This talk aims at initiating a discussion about what the future holds with respect to end-user configuration of IoT environments, such as a smart home. Will users "program" their homes using services like IFTTT.com, or they will prefer to browse through thousands of applications published in an online app store and try some of them out? Both approaches pose a number of research issues, and it is argued that the semantic technology can help to approach most of them. In addition, the talk briefly describes three related research efforts at VTT: Smart Modeller (for do-it-yourself future) and Semantic Smart Gateway Framework and Semantic Agent Programming Language (for app store future).

3.14 Using Insights from Social Computing to Augment Automotive Sensory Data

Claudia Müller-Birn (FU Berlin, DE)

Social computing systems provide added value by processing user information created by social interactions. Based on a couple of examples we show how these added values are realized in the Web and what challenges exist to tape the whole potential of these systems. Nowadays, ideas from social computing are entering completely new areas such as in the automotive industry. The car manufacturer Ford, for example, provides freely and at no cost its Sync AppLink to any automaker. The idea is to have a shared platform for developers to create novel apps based on sensory data. We enter this emerging area by presenting first results from a project on car data we have carried out recently and show, how car data can provide benefit to humans or web services. We conclude the talk by highlighting existing drawbacks and discussing implications for future research.

3.15 Towards Linked Closed Data?

Axel Polleres (Institute for Information Business of WU Wien, AT)

In the current trend for open Data, a lot of optimism is join into the belief that efforts like Linked Open Data from public sources will enrich and enable the usage of closed sensing data from all kinds of sources, and that aggregated dynamic sensing data will again be potentially published openly. However, various variables are unbound in this equation: How private can data in physical-cyber-social computing be? Can linked open data be trusted? How can physical-cyber-social-data be protected? How can data be charged and what's the value of aggregated data? I don't have answers to these questions but I'd like to discuss these issues in the workshop along with a roadmap and strategies on enabling technologies to answer them.

3.16 Identity: Physical, Cyber, Future

Matthew Rowe (Lancaster University, GB)

Social web systems offering communication functionality allow users to form groups, make connections and shape their identity over time. The development of identity, and the theoretical underpinnings that currently explain such developments, are based on psychoanalysis grounded by real-world, physical experiences. In this talk I will explain how such theories transcend physical-cyber boundaries and that users also exhibit identity crises in social web systems when interacting in a cyber environment. Such a transcendent phenomena leads to questions such as: how can identity be defined in the future? Does behaviour diffusion occur between the cyber and physical worlds? And how can we pre-empt physical decisions through cyber-based analyses?

3.17 Approximate Services in the Internet of Things

RangaRao Venkatesha Prasad (TU Delft , NL)

With the advent of newer technologies and highly miniaturized and computationally capable communicating devices, many possibilities of service provisioning is opening-up. The ICT (Information and Communication Technology) devices are now getting into our everyday life without our notice. Now, with the advent of Internet of Things (IoT), many newer possibilities of service provisioning are opening up. Since the needs of a person are different from another and so many different situations have to be dealt with, an exact service could not always be offered. To deal with this, approximate services paradigm is proposed. The idea is to find services that are close to the required with whatever the surrounding devices could provide opportunistically. We provided an example to motivate towards such a paradigm.

We identify some structural components of such a service. We expect a truly Google like searching for objects in our daily life is not ruled out in near future. We proposed many aspects of approximate service provisioning. We discussed briefly the notion and concept of approximate services and also the ideas of achieving such a paradigm. Further, since we know that there could be enormous number of devices in the future and in particular in IoT, a centralized solution seems impractical. Thus the use of distributed approximate services is the target. Here is a list of articles that deal with approximate services.

3.18 Physical-Cyber-Social Systems, Challenges and Opportunities

Amit P. Sheth (Wright State University – Dayton, US)

Main reference A. P. Sheth, P. Anantharam, C. Henson, "Physical-Cyber-Social Computing: An Early 21st Century Approach," IEEE Intelligent Systems, Vol. 28, No. 1, pp. 78–82, 2013.
URL http://dx.doi.org/10.1109/MIS.2013.20

The proper role of technology to improve human experience has been discussed by visionaries and scientists from the early days of computing and electronic communication. Technology now plays an increasingly important role in facilitating and improving personal and social activities and engagements, decision making, interaction with physical and social worlds, generating insights, and just about anything that an intelligent human seeks to do. Increasing number of exciting and important applications that technology have started to enable now include three interacting components:

- physical component: data about the physical world as captured by the sensors/devices and Internet of Things (IoT),
- social component: social interactions as enabled by social networks as well as the use of sensors to capture human interactions, and
- cyber component: by the use of massive amount of background knowledge often created through collective processes (e.g., Wikipedia), and other factual data (such as those becoming part of the Linked Open Data).

Given that the cyber component also provides computing needed to bridge physical world to the social world made up of humans, and whom any technology ultimately seeks to serve, we call this emerging computing paradigm as Physical-Cyber-Social Computing.

Related papers and talks/keynote on this topic appear at:
http://wiki.knoesis.org/index.php/PCS

3.19 Physical-Cyber-Social Agriculture

Kerry Taylor (CSIRO – Canberra, AU)

Many of the major challenges facing the world in the decade ahead are focused on food production: feeding the growing population; preserving biodiversity; mitigating and responding to climate change. As scientists, we dedicate ourselves to reducing uncertainty, whereas practicing farmers are daily experts in decision-making under uncertainty. How can we improve the precision and reliability of information for farm management? How can we make that information more directly actionable in the farmer's knowledge-intensive world? And are the future technology developments a threat to the best of traditional rural lifestyle and culture?

4 Working Groups

Ramesh Jain, Amit P. Sheth, Steffen Staab, Markus Strohmaier, Payam Barnaghi, Rajendra Akerkar, Pramod Anantharam, Josiane Xavier Parreira, Harith Alani, Ciro Cattuto, Benoit Christophe, Edward Curry, Emanuele Della Valle, Schahram Dustdar, Frieder Ganz, Michael Granitzer, Manfred Hauswirth, Laura Hollink, Andreas Hotho, Geert-Jan Houben, Mohan S. Kankanhalli, Artem Katasonov, Claudia Müller-Birn, Axel Polleres, RangaRao Venkatesha Prasad, Matthew Rowe, Vivek K. Singh, Kerry Taylor, and Koji Zettsu

4.1 Physical-Cyber-Social Computing

The discussions were followed in three main working groups focused on data, semantic and social aspects of Physical-Cyber-Social. Several joint discussions and talks, that are included in this report, also addressed inter-relations between these topics. The following sections summarise some of the key issues, challenges and solutions that were discussed in the working groups.

4.2 Working Group on Data

Physical-Cyber-Social (PCS) applications are characterised by the fact that information is represented through different types of data such as text (alphanumeric), images, audio, video and various data types such as like temperature, sound and light. The abundance of physically different data types or data heterogeneity is driven both by the increasing capabilities of computational systems as well as the increasing sophistication and ease of use of digital sensor technologies.

Some important issues with regards to the data in such systems include:

- Volume of data is growing by orders of magnitude every year.
- Different data sources provide facets of information which have to be combined to form a complete picture.
- There is a need to describe the dynamic nature of data and its refinement over time
- In several applications, real-time data processing is crucial
- Data assimilation in Physical-Cyber-Social system increasingly requires to consider the spatio-temporal characteristics of the data.

The above issues result in some challenges that are listed below:

1. How to trust the quality of data?
2. How to find useful data sets out of available data?

3. We need a mechanism to reason what the data pipelines do.
4. How to wrap up data, coming from multiple users and heterogeneous devices, into a common format and make accessible to the system?
5. How to handle multiple issues of scale, real-time processing and indexing of the physical organisation of data?
6. How to understand the power of data and a data pipeline? Determine the expressiveness of operators.
7. How to translate data from localised sensor/human input to higher level situational abstractions?

The Data Working Group followed several discussions with regards to the above issues and challenges. The following provides a summary of the discussions.

There are a lot of open issues which need to be solved to access wide variety of data with Physical-Cyber-Social computing. In order to make the data available for various tasks in Physical-Cyber-Social computing we need to describe the data. Web3.0 approaches provide appropriate solutions to describe sensor data in an abstract way. A more abstract and machine readable description of data allows for a better use of data. However, to make this vision a reality, we have to solve problems resulting from limited resources, limited bandwidth and/or missing and contradicting data. To sense and report the real world observations and measurements, one facet is using citizen sensing. Citizen sensing has many advantages to machine sensing. Machines are good at symbolic processing but are limited with respect to perception, which is the act of transforming sensory information into symbols/words that are expressive to humans. Furthermore, humans are better at contextualising data, filtering across multiple modalities, and capturing the resulting observations for future symbolic processing by machines. The Physical-Cyber-Social data can be combined with other data to create different abstractions of the environment, or it can be integrated into the data processing chain in an existing application to support context and situation awareness. It is necessary that heterogeneous data can be effectively integrated or one type of data can be combined with other physical, cyber, or social world data. Here, enhancing data interoperability (semantic interoperability), between different sources, by means of standardisation (common models) and benchmarks (describing the best scientific approach to describe the data) is required to facilitate the PCS data integration with other existing domain knowledge.

The cyber data can help to interpret/enrich the physical world data. There are many open issues. There is a need for a coherent implementation-independent description of the real world data by including rich semantics. There is also a need to describe the dynamic nature of data and its refinement over time. Finally, there is a need to provide patterns and best practices that describe how to implement such data descriptors in resource constrained environments.

4.3 Working Group on Semantics

The discussions in this working group focused on how the semantic Web standards should be extended and/or adapted to make them more suitable for Physical-Cyber-Social computing. One of the main questions was if we need upper ontologies for sensors. The participants argued that probably not; there are existing (partial) ontologies that cover important aspects of the requirements for the current use-cases. The group discussed whether concrete extensions and best practices of standard in terms of how to model “context” would be necessary? For example the W3C SSN ontology [SSN1], [SSN2] can be extended with concrete (recommended?) ontologies for modelling temperature, units, etc.

The following summarises the discussions on various concepts regarding the role and issues of using semantics in Physical-Cyber-Social systems that were discussed.

Provenance. Provenance is a key element in understanding where the data comes from the physical world (e.g. in what conditions the measurements were recorded, at which location and at what time). This is particularly important to derive and understand the different contexts in cases that the data is used in different distributed systems and applications and also when the data is aggregated (e.g. certain averages might be location or time dependent).

The W3C SSN ontology already provides a set of properties that allow recording provenance information. What is left to be done is to analyse how existing sensor ontologies need to be extended and how they align with other provenance ontologies so that the provenance for sensor data can be defined. This is an opportunity that needs to be explored.

Data linking. There are several existing efforts on integrating sensor data with static data at the metadata level. There might be an opportunity to investigate how to link dynamic data to static data via semantics. This will require to analyse use-cases and specific challenges in order to provide dynamic links and update the associations between the linked resources.

Data annotation / Standards for data representation and processing. The key question related to annotation is whether having standards are always the best choice. An alternative approach to common standardisation can be providing data structures that are better suited to certain applications, e.g. annotations for sensor data, as opposed to using properties. This also applies to the query languages. Named graphs are not suitable for such annotations. This is particularly true for annotations that are associated with certain "logics".

An important concept to note is that the linking is not an "either or" decision; time and other attributes might require to be stored in annotations for faster processing, whether other types of provenance (e.g. description of a procedure) can take advantage of the provenance ontologies. Existing query processing solutions would need to be revised to account for the annotations. Another opportunity is using the temporal logic for reasoning data in Physical-Cyber-Social computing.

Emerging semantics. There are existing solutions that describe containers (e.g. representation models to express Resource A is a sensor), but the same is not true for the content (e.g. is the weather hot or cold). This applies to both physical and citizen (social) sensing. However, it is even more challenging when it comes to social sensors, as their content often do not have a defined domain (e.g. what does a Tweet mean? Is it an event an observation or a feeling?)

One of the challenges is how to model social sensors. Extracting semantics from text only works in static environments. In social sensors the semantics change. This is different from physical sensors because on its own an individual sensor observation is rather meaningless. How should this be attacked? How can such social data be correlated with physical data?

Another question is related to tools and whether the existing tools are adequate to extract information from social sensors. The semantics might influence the data extraction procedure. The interpretation of social data may require a "longer pipeline" compared to physical data. Context description becomes even more challenging in social semantics, where the meaning of a particular post or tweet, and how this relates to other entities can be very subjective.

4.4 Working group on Social Systems

Social science research for decades has been mostly studying social interactions in the physical world by formulating social science theories and conducting scientific experiments. These experiments are focused on the social interactions (interpersonal) in the physical world. Social networks on the cyber world have lead to unprecedented connectivity for information sharing and social interactions extending beyond the social interactions in the physical world. Citizens behave as sensors outside the context of social interactions and this provides a valuable complementary source of information in a Physical-Cyber-Social system.

With Physical-Cyber-Social computing there is an opportunity to understand two fundamental questions:

Q1. How does the emergence of physical-cyber-social systems enable new forms of social science research? Social scientists have access to the social networks on the cyber world which serves as a valuable source of information to understand social interactions. Social networks consist of social connections and interactions in a dynamic setting between people in the network. Social scientists are no longer limited to social interactions in the physical world. They can study interactions in the social networks to validate existing theories and to propose new theories. The validation of social theories is conventionally done by conducting surveys and by employing people to participate in a study. Social scientists have now access to the data from the real-world social interactions in the cyber world. These interactions provide a sample (incomplete) picture of the real-world social interactions but on a massive scale. The social science theories need to be adopted to deal with this incompleteness and data biases in a physical-cyber-social system.

Q2. How can we use social theories to inform the design of novel Physical-Cyber-Social systems? Understanding social theories is the first step toward building systems that interact smoothly with people. These theories are crucial in the design and development of physical-cyber-social systems. For example, a system that interprets various social interactions can be used to capture images/videos. One of the key challenges in dealing with social systems is how to maintain the privacy of participants. Privacy becomes an essential component and it is crucial for wide adoption of physical-cyber-social systems. This is partly due to the fine-grained information collected from sensors and its correlation with behavior patterns that would reveal personal information which may be misused. For example, a smart-meter installation may result in revealing the occupancy of a house.

Some of the key challenges when using social data, or in general any data processed by Physical-Cyber-Social computing are listed below.

- Social bots: there are attempts to simulate and flood the social data generated automatically by programs that try to emulate human behaviour. Such a source of information should be used carefully and separated from rest of the social data.
- 'Twitter' data is not always reliable (i.e. requires careful consideration): while social data is available in massive scale (e.g., around 500 million tweets a day), the data is often very noisy, informal, and unevenly distributed.
- Assessing the relevance of Twitter to a problem: not all studies can be done on Twitter data since the nature of data and the social behaviours have a great variance on Twitter.
- Sometimes there is an assumption that data is available at all times: theoretically, there is data available related to various events. However in reality, it may be very hard to find sensors and their observations on Twitter and the Web in general. Choosing the

appropriate data source is an important challenge in the context of Physical-Cyber-Social systems.

- Understanding the feedback mechanism: social scientists need to understand the feedback mechanism that exists between the physical world and the social world interactions. This is a challenging and important task to gain insights into systems that involve the social component.
- Data biases are crucial: social scientists should consider data biases carefully with the availability of massive data from social networks such as Twitter.
  - Tweeter vs. non-Tweeter: for example, not all the social interactions are present on Twitter and only a group of people may use Twitter.
  - people who can read vs. cannot read: such biases have been around even in the past when social scientists has to deal with the physical world for conducting their studies.
- Combining reactive vs. non-reactive data: reactive data are those collected by social scientists through surveys and questions. Non-reactive data are those collected by sensors on a continuous basis.
  - collecting sensor data: this is a type of non-reactive data and it is available continuously without active involvement of people.
  - how do we combine survey data and log data?: this is a challenge as the granularity of both of them are different. Also, combining textual data (e.g., emotion) with sensor data (e.g., high heart rate) may be particularly difficult.
  - understanding the differences between reactive vs. non-reactive data enable us to design methodologies for dealing with them in a single framework.
  - Can the theories from social science be translated into graph formalisms and whether this will help expose social theories?

5 Open Problems

As the Web provided useful mechanisms to access and use new types of resources, techniques increasingly moved from syntactic and structural to semantic representations. There is also a recent resurgence of research towards Computing for Human Experience [Sheth10]. This line of work has a long lineage, starting in part with Vannevar Bush's Memex through Mark Weiser's "Computing in the 21st Century" and others. But the essence of the vision incorporates technology that serves human needs without explicit human effort. This multidisciplinary seminar seeks to develop a vision of a new class of 21st century systems involving machines (physical), computing and communications (cyber) and human-centric and social systems (social). Several challenges arise in this context, which can be categorized into the following major topics that were discussed at the seminar:

1. emerging manifestations of physical-cyber-social systems, including applications in industry markets such as health and transportation. Specifically investigation of new capabilities in terms of improving human experience and serving societal causes in ways that have not been possible earlier.
2. How are emerging physical-cyber-social systems different than integration of cyber-physical systems with social systems as conceived today?
3. How do physical-cyber-social systems utilise or benefit from Web of Things/ Internet of Things, Semantic Web, Crowd sourcing and Semantic Social Web, Semantic Sensor Web, Intelligent/natural Interfaces, Ambient Intelligence and other technologies/ advances during the last decade?

4. What novel disruptive applications are likely to result due to emerging technology in Physical-Cyber-Social systems after the initial obvious applications are addressed?
5. What is the role of semantics in physical-cyber-social systems, and how do existing semantic technologies accelerate or constrain the emergence of Physical-Cyber-Social computing?
6. How do physical-cyber-social systems transform traditional perceptions of physical objects, online engagement and social interactions?
7. What implications will the confluence of physical-cyber-social systems have on societies, including aspects such as citizen participation, democracy, open government, open government data and others?

6 Panel Discussions

Soon there will be tremendous volumes of data collected from the physical world in addition to the growing data collected from the cyber and social world. The combination of this cyber-physical and social data can help us to understand events and changes in our surrounding environments better, monitor and control buildings, homes and city infrastructures, provide better healthcare and elderly care services among many other applications. To make efficient use of the physical-cyber-social data, integration and processing of data from various heterogeneous sources is necessary. Providing interoperable information representation and extracting actionable knowledge from deluge of human and machine sensory data are the key issues.

The role of semantics for interoperability, integration, and improved querying has been investigated for over four decades. The 'Semantic Web' movement brought focus to using semantics and metadata initially to the Web documents. As the Web provided useful mechanisms to access and use new types of resources–richly represented data, services, user generated content and other social data, sensor and devices (Web of Things – WoT) data–techniques increasingly moved from syntactic and structural to semantic ones. Compared to the semantic systems built using Semantic Web languages, standards and mainstream Semantic Web technologies that employ formal representation of semantics, however, more systems are being built using informal and implicit forms of semantics. One reason is that the role of the Web is increasingly becoming diffused and incidental (e.g., more people access content through applications compared to the Web browsers). The second reason is that lighter-weight approaches have led to better developer and user engagements, and have become a lot more scalable. Apple Siri, IBM Watson, and Google Knowledge Graph are examples of using semantics at scale, but where the formal form of semantic representation or RDF/SPARQL have not found a place. All these lead us to think that 10 years from now, Semantic Web would be thought of as something that popularized the core value proposition of semantics – better search, interoperability/integration and analysis – to deal with and exploit a vast variety of things that the Web (and its on going transformations) interconnects. An analogy that comes to mind is that of Object Oriented Databases which generated huge excitement in the 1980s, and indeed had a number of secondary impacts, it only remained a niche technology, product class and market. Simultaneously, Semantic Web is increasingly merging with other powerful technologies that support semantics, including Machine Learning, NLP, and Knowledge-based systems where background knowledge is applied. Consequently, what we think of as rather distinct Computer Science areas today will not retain strong distinctions, but will broadly incorporate semantics.

References

1 Amit Sheth, "Computing for Human Experience: Semantics-Empowered Sensors, Services, and Social Computing on the Ubiquitous Web," IEEE Internet Computing (special issue on Internet Predictions), 14 (1), January/February 2010, pp. 88–91.

2 Laurent Lefort *et al*, Semantic Sensor Network XG Final Report, W3C Incubator Group Report 28 June 2011, http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/

3 Michael Compton *et al*, The SSN ontology of the W3C semantic sensor network incubator group, Journal of Web Semantics, Volume 17, December 2012.

4 Moshe Vardi, Computing for Humans, Communications of the ACM, Vol. 54, No. 12, December 2011, page 5.

Participants

- Rajendra Akerkar
Vestlandsforsking – Sogndal, NO
- Harith Alani
The Open University – Milton Keynes, GB
- Pramod Anantharam
Wright State University – Dayton, US
- Payam M. Barnaghi
University of Surrey, GB
- Ciro Cattuto
ISI Foundation, IT
- Benoit Christophe
Bell Labs – Nozay, FR
- Edward Curry
National University of Ireland – Galway, IE
- Emanuele Della Valle
Polytechnic Univ. of Milan, IT
- Schahram Dustdar
TU Wien, AT
- Frieder Ganz
University of Surrey, GB
- Michael Granitzer
Universität Passau, DE
- Manfred Hauswirth
National University of Ireland – Galway, IE
- Laura Hollink
Free Univ. of Amsterdam, NL
- Andreas Hotho
Universität Würzburg, DE
- Geert-Jan Houben
TU Delft, NL
- Ramesh Jain
Univ. of California – Irvine, US
- Mohan S. Kankanhalli
National Univ. of Singapore, SG
- Artem Katasonov
VTT – Espoo, FI
- Claudia Müller-Birn
FU Berlin, DE
- Axel Polleres
Wirtschaftsuniversität Wien
- RangaRao Venkatesha Prasad
TU Delft, NL
- Matthew Rowe
Lancaster University, GB
- Amit P. Sheth
Wright State University – Dayton, US
- Vivek K. Singh
MIT, US
- Steffen Staab
Universität Koblenz-Landau, DE
- Markus Strohmaier
Universität Koblenz-Landau, DE
- Kerry Taylor
CSIRO – Canberra, AU
- Josiane Xavier Parreira
National University of Ireland – Galway, IE
- Koji Zettsu
NICT – Kyoto, JP