DAGSTUHL
REPORTS

Volume 4, Issue 6, June 2014

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/dagpub/2192-5283

Publication date
March, 2015

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`.

Digital Object Identifier: 10.4230/DagRep.4.6.i

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
http://www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 14231

Scientific Visualization

Edited by

Min Chen[1], Charles D. Hansen[2], Penny Rheingans[3], and Gerik Scheuermann[4]

1 **University of Oxford, UK,** `min.chen@oerc.ox.ac.uk`
2 **University of Utah – Salt Lake City, US,** `hansen@cs.utah.edu`
3 **University of Maryland Baltimore County, US,** `rheingan@cs.umbc.edu`
4 **Universität Leipzig, DE,** `scheuermann@informatik.uni-leipzig.de`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14231 "Scientific Visualization". It includes a discussion of the motivation and overall organization, an abstract from each of the participants, and a report from each of the working groups.

Seminar June 1–6, 2014 – http://www.dagstuhl.de/14231
1998 ACM Subject Classification I.3.8 Applications
Keywords and phrases data visualization, multi-fields, uncertainty, environmental visualization
Digital Object Identifier 10.4230/DagRep.4.6.1

1 Executive Summary

Charles D. Hansen
Min Chen
Penny Rheingans
Gerik Scheuermann

Scientific Visualization (SV) is the transformation of digital data, derived from observation or simulation, into readily comprehensible images, and has proven to play an indispensable part of the scientific discovery process in many fields of contemporary science. Since its inception two decades ago, the techniques of Scientific Visualization have aided scientists, engineers, medical practitioners, and others in the study of a wide variety of data including, for example, high-performance computing simulations, measured data from scanners (CT, MR, confocal microscopy, satellites), internet traffic, and financial records. One of the important themes being nurtured under the aegis of Scientific Visualization is the utilization of the broad bandwidth of the human sensory system in steering and interpreting complex processes and simulations involving voluminous data across diverse scientific disciplines. Since vision dominates our sensory input, strong efforts have been made to bring the mathematical abstraction and modeling to our eyes through the mediation of computer graphics. This interplay between various application areas and their specific problem-solving visualization techniques has been the goal of all the Dagstuhl Scientific Visualization seminars and was emphasized in the seminar which took place June 1–6, 2014.

Our seminar was focused on four research themes that will have significant impact in the coming years. These four themes reflect the heterogeneous structure of Scientific Visualization and the current unsolved problems in the field. They represent cross-cutting topic areas

where applications influence basic research questions on one hand while basic research drives applications on the other. This cross-cutting feature makes Dagstuhl a unique setting in the research community, as the scientific coverage of the seminar is broader than other more focused workshops and seminars hosted at Dagstuhl while much more focused and forward-looking than general conferences. Our four themes were:

Uncertainty Visualization: Decision making, especially rapid decision making, is always made under uncertain conditions. As former English Statesman and Nobel Laureate (Literature), Winston Churchill said, "True genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information." and echoed by Nobel Prize winning physicist Richard Feynman, "What is not surrounded by uncertainty cannot be the truth." Uncertainty visualization seeks to provide a visual representation of errors and uncertainty for three-dimensional visualizations. Challenges include the inherent difficulty in defining, characterizing, and controlling comparisons among different data sets and in part to the corresponding error and uncertainty in the experimental, simulation, and/or visualization processes.

Integrated Multi-field Visualization: The output of the majority of computational science and engineering simulations is typically a combination of fields, generally called multi-field data, involving a number of scalar fields, vector fields, or tensor fields. Similarly, data collected experimentally is often multi-field in nature (and from multiple sources). The ability to effectively visualize multiple fields simultaneously, for both computational and experimental data, can greatly enhance scientific analysis and understanding. Multi-scale problems with scale differences of several orders of magnitude in computational fluid dynamics, material science, nanotechnology, biomedical engineering and proteomics pose challenging problems for data analysis. The state of the art in multi-scale visualization considerably lags behind that of multi-scale simulation. Novel solutions to multi-scale and multi-field visualization problems have the potential for a large impact on scientific endeavors.

Environmental Scientific Visualization: Environmental scientific visualization or environmental visualization refers to a collection of visualization applications that deal with captured and simulated data in climate research, atmospheric and environmental sciences, earth science, geophysics and seismic research, oceanography, and the energy industry (e. g., oil, gas and renewable energy). Research in these application domains has a huge impact on mankind, and typically faces serious challenges of data deluge (e. g., very large volumes of multi-spectral satellite images, large data collections from different sensor types, ensemble computation of very large simulation models, scattered, time-varying, multi-modal data in seismic research). In comparison with biomedical visualization and small-to-medium scale computational fluid dynamics, the effort for developing visualization techniques for such applications has not been compatible with the importance and scale of the underlying scientific activities in these application domains. Scientific progress in the areas of the environment and sustainability is critical in the solution of global problems and scientific visualization has great potential to support this progress.

Scientific Foundation of Visualization: The rapid advances in scientific visualization have resulted in a large collection of visual designs (e. g., for flow visualization), algorithms (e. g., for volume rendering), and software tools and development kits. There have also been some scattered investigations into the theoretic and perceptual aspects of visualization. However, many fundamental questions remain unanswered, such as, why is one visual design more effective than another, can visual designs be optimized and how, what is the role of visualization in a scientific workflow and how can such a role be formalized in a

scientific workflow, can visualization quality be measured quantitatively and how, and what is the most effective way to conduct perceptual and usability studies involving domain experts? With the experience of delivering technical advances over the past two decades, it is timely for the visualization community to address these fundamental questions with a consorted effort. Such an effort will be critical to the long-term development of the subject, especially in building a scientific foundation for the subject.

The format of the seminar was two-part: having groups of four to five shorter talks followed by a panel of the speakers which encouraged discussion and breakout groups on the four topics as well as topics which came up at the meeting. The scientific presentations were scheduled at the beginning of the week in order to simulate the discussions from a broad perceptive. Unlike the typical arrangement, all presentations in each session were given in sequence without a short Q&A session at the end of each talk. Instead, all speakers of a session were invited to sit on the stage after the presentation, and answer questions in a manner similar to panel discussions. This format successfully brought senior and junior researchers onto the same platform, and enabled researchers to seek a generic and deep understanding through their questions and answers. It also stimulated very long, intense, and fruitful discussions that were embraced by all participants. The breakout groups focused on the general themes and are reported in the following sections.

2 Table of Contents

3 Overview of Talks

3.1 Implications of Numerical and Data Intensive Technology Trends on Environmental Scientific Visualization

James Ahrens (Los Alamos National Lab., US)

Technology trends in numerically and data intensive computing have the potential to reshape and significantly improve how we visualize and analyze the massive data streams resulting from environmental scientific simulations. Next generation exascale supercomputers are bound by power and storage constraints, requiring our current visualization approach to transform from the general post-processing analysis of raw results to the automated generation of in situ reduced-sized data products during a simulation run. Current data intensive technology trends provide inexpensive access to powerful commodity parallel computing resources for an important class of structured problems. In addition, intuitive, web-based and query-driven interfaces to access and understand data are now the norm. In this talk, I will describe these trends in more detail and challenge the community to think about how these trends integrate with their research approaches. As an example, I will present a novel scientific visualization and analysis process that leverages both numerical and data intensive technology trends.

3.2 Reconstruction of Functions from Simplified Morse-Smale Complexes

Georges-Pierre Bonneau (INRIA Rhône-Alpes, FR)

We will give an overview of our on-going work on topology-based visualization of scalar data. In this area we propose the use of piecewise polynomials interpolants to reconstruct data based on their simplified Morse-Smale complexes. We show how it is possible to define monotonic polynomial interpolants that can be used as patches to represent the data inside each Morse-Smale cells.

3.3 Exploring Glyph-based visualization for Multivariate and Multidimensional data

Rita Borgo (Swansea University, UK)

Simultaneous visualization of multi-dimensional and multivariate data is a complex task. An adequate choice of visual encoding must keep into consideration both expressiveness (e. g. number of parameters easily representable), and comprehensibility of design (e. g. interpretation). Glyph, or iconic, visualization represents an attempt at encoding multivariate information in a comprehensible format, allowing multiple values to be encoded in the

parameters of the glyphs. Geometric encoding allows to integrate multiple variables within a single item and therefore the creation of a unique image, or signature, to be constructed for each data point. However, as the number of data points displayed increases, the amount of visible variation per glyph decreases, potentially obscuring the visibility of interesting structures and patterns in the data. When locality is lost global structures can still emerge by exploiting local characteristic of glyph's geometry to obtain texturing effects, going beyond the expressive extent of the single glyph.

The Glyph paradigm remains to be productive in different contexts from scientific and information visualization to computer vision. Several questions still remain open in terms of design, usability, learnability. In this talk I will review most relevant contributions in glyph-based visualization, report and discuss on recent results and set out future line of work to bring forward research in the field.

3.4 Evolutionary Visual Exploration

Nadia Boukhelifa (University Paris-Sud – Gif sur Yvette, FR)

Joint work of Boukhelifa, Nadia; Bezerianos, Anastasia; Cancino, Waldo; Lutton, Evelyne
Main reference N. Boukhelifa, Waldo Gonzalo Cancino Ticona, A. Bezerianos, E. Lutton, "Evolutionary Visual Exploration: Evaluation With Expert Users," iComputer Graphics Forum, 32(3pt1):31–40, 2013.
URL http://dx.doi.org/10.1111/cgf.12090

The purpose of visual exploration is to find meaningful patterns in the data which can then lead to insight. In a high-dimensionality context, this task becomes rather challenging as viewers may be faced with a large space of alternative views on the data. In this talk I will describe a framework that combines visual analytics with stochastic optimisation to aid the exploration of multidimensional datasets characterised by a large number of possible views or projections. I will present initial results and highlight some challenges in designing visual analytics tools that combine user input and automatically calculated metrics to guide user exploration.

3.5 <Mathematical, Visual> Foundations of <Visualisation, Mathematics>

Hamish Carr (University of Leeds, UK)

We are accustomed to thinking of visualisation as being built on mathematics, but the truth is that mathematics is also built on visualisation and visual thinking. As a result, visualisation drives mathematics nearly as much as mathematics drives visualisation. I will discuss some aspects of this reciprocal relationship.

3.6 Analyzing User Interactions for Data and User Modeling

Remco Chang (Tufts University, US)

User interactions with a visualization system reflect a great deal of the user's reasoning process and personality. In this talk, I will present techniques that we have developed to analyze the user's interactions in order to (a) model the data in the form of metric learning that reflect a user's understanding of high-dimensional data, and (b) model the user and learn the user's individual differences and analysis behavior. In addition, I will discuss the relevant cognitive traits and states that influence a user's analysis process and conclude by suggesting how this research form the basis of mixed-initiative visual analysis systems.

3.7 Some thoughts on using signs to think

Jian Chen (University of Maryland, Baltimore Country, US)

From Bertin's semiotics theory to MacKinlay's automatic design of encoding based on symbolic rankings to the commercial software of Tableau, the study of signs has had great impact on how we understand large abstract information in the so-called information visualization. In scientific visualization, implicit and explicit in our visualization of spatial data are also symbols. With the realm of the computing moves from petascale into the exascale, beyond the challenges of efficiently executing scientific rendering and simulation, perhaps even greater, there are challenges of visualization and interpretation of results produced by massive heterogeneous datasets. Deriving insights for diverse analytical tasks and results have been forced to struggle with evolving visualizations that are beyond our mental grasp.

In this talk, I will address evolving visualization issues. I think the cornerstone is to support human limited working memory with sign manipulation. I will discuss how the theory of sign can be extended to unify scientific and information visualizations and analytical process. I will discuss issues related to the tradeoffs between physical large space and virtual large space + spatial index. And I will also discuss the categories of visual dimensions or metaphors or abstractions that we use to make data comprehensible in bat flight motion analysis.

3.8 Exascale Computing and Uncertainty Visualization

Hank Childs (University of Oregon, US)

Exascale computers – computers that can do 10^{18} floating point operations per second – are predicted to arrive in the next four to six years. To field such a machine, hardware architects must make difficult decisions on where to spend their money, between network, I/O, memory,

and compute. Of these four factors, I/O bandwidth is increasingly viewed as a luxury, and disks are not keeping pace with overall machine improvement. Reduced I/O bandwidth creates an I/O bottleneck, and data compression is one technique frequently considered for this emerging problem. With this presentation, I will discuss the opportunity for uncertainty visualization in this compression process and the resulting visualizations.

3.9 Representing Chronological Events with Heatmaps

João Luiz Dihl Comba (Federal University of Rio Grande do Sul, BR)

Heatmaps are well-known visual representations of data that employ a color- coded array of values disposed in a matrix format. In this talk I will review my recent experience on using heat maps to describe chronological events over different types of datasets. In the first example, I will show how heatmaps were used to encode heart-rate effort of several people during a running race. There are many ways to explore the data disposed in this heat map, and I will discuss several interesting alternatives to look at this data from different perspectives. The second example is related to the problem of monitoring forest ecosystems using digital cameras, which allows the study several aspects of tree phenology, such as leaf expansion and leaf fall. Since phenological phenomena are cyclic, the comparative analysis of successive years is largely used to identify interesting variation on annual patterns that have been related, for instance, to changes on temperature due to global warming. Usually, phenologists draw, for each year, a 2D-plot of the average of a given quantity in an image (e. g., average of green information) against the days of the year. I will show how a special-type of heat map, called Chronological Percentage Maps (CPMs), is a more expressive and compact visual representation to support phenological analysis from vegetation digital images. We demonstrate the use of CPMs in three different datasets, comprising data of up to 9 consecutive years, and discuss the further applications on phenology.

3.10 A hierarchical approach to topological data analysis and visualization

Leila De Floriani (University of Genova, IT)

Topology plays a very relevant role in analyzing shapes defined by a finite set of data points in three dimensions and higher and discretized as simplical or cell complexes. In this talk, I discuss the application of algebraic topology tools, namely homology and topological structural descriptors rooted in Morse theory, to the analysis and understanding of such shapes. The fundamental issues in computing topological invariants and descriptors for real data sets are the size and the dimension of the data. This poses challenging theoretical and computational problems. This talk focuses on hierarchical approaches to homology computation and to the construction of multi-resolution topological descriptors which highly improve both the effectiveness and the efficiency of such topological tools.

3.11 Recent Multifield Applications in Biophysics and Future Visualization Engineering Research

Thomas Ertl (Universität Stuttgart, DE)

This talk reports on recent multifield visualizations performed in the context of a biophysics application and on an engineering-type research proposal for quantifying visual computing systems. Our physics colleagues in the collaborative research center on particle simulation are interested in understanding new methods for DNA sequencing based on nanopores. They perform molecular dynamics simulations of DNA strands getting pulled through a nanopore by an electric field and surrounded by Ka+ ions. The simulation confirms that DNA consisting of CG base pairs only results in a lower measured current than DNA consisting of AT base pairs only. A visual analysis of the aggregated density and velocity fields confirms that this is due to the lower mobility of ions moving close to the CG DNA groove than the more outside and more freely moving ions in the AT case. Closer inspection of the ion flux vector glyphs reveals an unexpected phenomenon where ions seem to move against the electric field. Predicting the application performance of a visualization application running on a complex visual computing hardware is difficult due the manifold variations of algorithms, application parameters, and hardware setups. We propose to predict such interactive application performance based on highly parameterized performance models and collections of performance statistics. We also argue that the engineering aspects of building systems in general have been underrated in visualization research.

3.12 Contractible Parallel Coordinates for Sparse Modeling

Issei Fujishiro (Keio University, JP)

Sparse modeling is intended to exploit the inherent sparseness that can be commonly observed in huge quantities of high-dimensional scientific data, and thereby enabling to extract the maximum amount of information from the data effectively and efficiently. In order to come up with a visualization platform for facilitating such sparse modeling, we have focused primarily on the contractility of parallel coordinates. In this talk, we will present two promising schemes: one is to use spectral graph analysis of correlation among axes and the other is to rely on biclustering to identify clusters of data samples and groups of highly-correlated axes simultaneously. Enhanced visualization capabilities of contractible parallel coordinates will also be discussed.

3.13 Topological Visualization of Multivariate Data

Christoph Garth (TU Kaiserslautern, DE)

The talk briefly describes recent research on the characterization of joint extremal structures of multiple scalar functions over a common domain. Adapted from concepts introduced in the field of non-linear optimization, and motivated by applications in flow and ensemble visualization, Pareto sets over multivariate scalar fields capture jointly minimal and maximal structures. Furthermore, a brief survey is given on open problems in multivariate topological visualization.

3.14 Interactive Visualization of High Resolution Planetary Data

Andreas Gerndt (DLR – Braunschweig, DE)

Earth and planetary exploration missions produce huge amount of data from different sensors and cameras. Those datasets are stored in open access databases but are not exploited at all as appropriate architectures and interactive tools for remote access and analysis are missing. One approach is to convert terrain data in LOD data structure for interactive visualization in virtual environments. On top of that, additional methods can offer tools to measure features and retro-deform the surface to assess geodetic hypotheses. Haptics rendering can improve perception and speed-up the workflow. Such environments can be used to incorporate more data types like sub-surface data and atmospheric science data. Also rover operations can be planned in advance and during on- going missions. But not only the data but also the scientist are distributed all over the world. 3D tele-presence can also bridge this problem to discuss scientific findings in immersive virtual environments.

3.15 Towards Comprehensible Modeling

Michael Gleicher (University of Wisconsin – Madison, US)

Building (mathematical) models is one of the main ways we deal with large amounts of data. There is a diverse range of usages of models, and a myriad of mathematical modeling methods have been developed. Some goals of modeling are well understood, for example predictive accuracy, descriptive fidelity, and robust generalizability. However, one aspect of modeling seems to be under-explored: comprehensibility, or, more generally, usability. Comprehensibility of models is likely to become more important as we expand the range of applications and users where modeling is applied, enlarge the range and sophistication of modeling techniques, and attempt to model increasingly complex phenomena and larger data sets.

In this talk, I will raise the issue of comprehensibility in modeling as a new core challenge for visualization research. I will attempt to raise some of the questions involved, in part through examples of our early attempts to address them. I will discuss our limited understanding of model comprehensibility, I will describe some initial efforts to give users control over the tradeoffs between the comprehensibility of models and more traditional model qualities, such as predictive accuracy, through the development of new analytic approaches. I will show some of our attempts to create visualization tools specifically designed to help users understand models.

3.16 Comparative and Quantitative Visualization in Material Sciences

Eduard Groeller (TU Wien, AT)

Materials like multi-material components (MMC) and carbon fiber reinforced polymers (CFRP) require novel non-destructive testing approaches. 3D XRay Computed Tomography (XCT) and X-Ray Fluorescence (XRF) are scanning modalities for the analysis and visualization of features and defects in industrial work pieces. Several application scenarios in the area of nondestructive testing are treated in this respect. Examples are porosity maps, mean objects, and fuzzy CT metrology. The rich data sources require integrated and aggregated visualization approaches. Despite various causes of uncertainty, quantitative visual representations are necessary. Additionally interactive visual inspections allow the domain expert to cope with data complexity. Due to the rapid development of scanning devices, material sciences and non-destructive testing constitute a challenging application domain for innovative visualization research. Potential research directions will be discussed at the end of the talk.

3.17 Uncertainty in medical imaging on the example of Electrical Impedance Tomography

Hans Hagen (TU Kaiserslautern, DE)

Electrical Impedance Tomography (EIT) is a fast, cheap, risk-free, and convenient imaging technique for conductivity changes in the body. It suffers from low spatial resolution and noise, and only displays a 2D projection of the 3D conductivity changes. Furthermore, image reconstruction from measurements is an ill-posed inverse problem.

Additionally, several causes introduce further uncertainty: Often, the source of an effect is not clear, for example breathing, cardiac activity, organ movement, or a change of the medical condition. Spatial correspondence to organ structures are unclear due to the low image resolution and the projection into 2D. Most importantly for lung researchers and clinicians, the anatomical lung boundary cannot be determined reliably.

In this talk, two of our efforts to tackle this uncertainty will be reported: First, a 3D visualization of patient-specific CT data, augmented with a multi-material segmentation, with an embedding of the time-dynamic EIT images. Second, a study concerning the spatial precision of lung and heart shape in EIT images compared to reference CT data.

3.18 Semi-abstract visualization of rich scientific data

Helwig Hauser (University of Bergen, NO)

In scientific visualization, we are used to mapping the spatial aspects of the data to the axes of the visualization space. As scientific data become more information-rich (e. g., by being time-dependent, multi-variate, or ensemble data), we also map non-spatial aspects of scientific data, following visualization designs which otherwise are known from information visualization (instead of a spatial visualization, or in addition). It seems worthwhile to also consider mixed strategies, where we consider semi-spatial, semi-abstract mapping strategies for visualizing scientific data and it may be promising to explore according opportunities more (as compared to classical mappings) even more in the future. In addition to bringing this thought (and discussion) to the seminar, we also look at selected examples of visualization designs which were inspired by this idea.

3.19 Liberate Visualization!

Hans-Christian Hege (Konrad-Zuse-Zentrum – Berlin, DE)

For cross-disciplinary exchange and public communication of an academic discipline, necessary prerequisites are clearly defined terms, conceptual clarity, a well-conceived taxonomy and a comprehensible division into sub-areas. Our young and transforming discipline does not yet fulfill these preconditions. This becomes particularly clear when looking at the division of the field into subfields, which currently is determined more by sociological and political than substantive considerations.

Instead of delving into the debate about a sensible definition of our sub-areas and their denominations, in this talk the title of our discipline, "visualization", shall be discussed. Its etymological roots, the Latin verb 'videre' and the noun 'visus', date back at least 3000 years. These Latin terms carried a multiplicity of meanings, which is still reflected in the variety of meanings of the English term 'vision'. The derived late Latin term "visualia", meaning 'organ of sight' or 'eye', has been used since the 4th century AC and 'visualis', meaning 'being attained by sight', since the 6th century AC. Both terms are precursors of the English words 'visual', 'to visualize' and 'visualization', which appeared in scientific texts in the 15th century, in the year 1817 and in 1883, respectively. The new English terms inherited the semantic ambiguities of the Latin antecedents. Particularly the two fundamental kinds of seeing are not distinguished: (1) seeing with the mind's eye, i. e. imagining things or conceiving mental images based on internal information, and (2) perceiving with the physical eye, i. e. sensing external visual information. Nowadays, were much finer distinctions are made, these broad terms often are less suited.

For instance, regarding external visualizations, one should differentiate whether the input information is internally represented (art works; visualizations of concepts, knowledge or mental images), or externally (photography; imaging; data visualization). The semantics of "visualization" in the public, however, is far from all these meanings. Concluding from the bestselling English books carrying "visualization" in their title, it denotes a specific internal

visualization, namely a technique for focusing on positive mental images in order to achieve particular goals. But even in science, when talking or writing about external visualization, the term "visualization" often does not mean "data visualization", but "imaging", i. e. capturing physical observables with a device.

In conclusion, "visualization" and its Latin antecedents are very old terms, whose multiple meanings have changed only gradually since their invention. Nevertheless, since about 25 years our small community in computing science entitles its discipline "visualization" significantly confining the traditional meaning of the term. In order to facilitate cross-disciplinary and public communication, we should instead agree on a technical term that denotes our field of activity and thereby reliberate "visualization".

3.20 Simplified vector field representations

Mario Hlawitschka (Universität Leipzig, DE)

We create simplified vector field representations by approximating fields using a pre-defined allowable error. We use these fields to derive visualizations of the data in a simplified way. On the other hand, these approximations can be seen as a simplification of the data that will serve as the basis for efficient implementations of various data analysis techniques.

3.21 Moment Invariants for Flow Fields Analysis

Ingrid Hotz (DLR – Braunschweig, DE)

The analysis of flow data is often guided by the search for characteristic structures with semantic meaning. In this work we consider structures as non-local flow patterns which can be defined in an analytical way with respect to some flow model; or they could be identified by a human observer. Flow analysis then means finding similar structures in the same or other datasets. The major challenges related to this task are to specify the notion of similarity and define respective pattern descriptors. While the descriptors should be invariant to certain transformations, as rotation and scaling, they should provide a similarity measure with respect to other transformations, as deformations. In this work, we use moment invariants as pattern descriptors.

3.22 Taking Stock of Visualization Research and Education

Christopher R. Johnson (University of Utah, US)

It has been more than ten years since I gave a presentation at a Dagstuhl Scientific Visualization Workshop that resulted in my Visualization Viewpoints article on visualization

research challenges: C.R. Johnson. "Top Scientific Visualization Research Problems," In IEEE Computer Graphics and Applications: Visualization Viewpoints, Vol. 24, No. 4, pp. 13-17. July/August, 2004.

In this presentation, I will take stock of visualization research and education and explore both recent trends and possible future needs as we continue to advance as a field.

3.23 The Difficulties with Ensemble Visualization

Kenneth Joy (University of California – Davis, US)

Whereas historically, most visualization techniques have focused on the analysis of the output of simulations, advances in computational power now enable domain scientists to address conceptual and parametric uncertainty by running simulations multiple times in order to sufficiently sample the uncertain input space, or the uncertain model space. While these approaches help address conceptual model and parametric uncertainties, the ensemble datasets produced by this technique present a special challenge to visualization researchers as the ensemble dataset records a distribution of possible values for each location in the domain. Contemporary visualization approaches that rely solely on summary statistics (e. g., mean and variance) cannot convey the detailed information encoded in ensemble distributions that are paramount to ensemble analysis; summary statistics provide no information about modality classification and modality persistence. In this presentation, we review a number of techniques that address these model and parametric uncertainty analysis problems, and give examples of their usage.

3.24 Verifiable Visualization: Lessons Learned

Robert Michael Kirby (University of Utah, US)

Joint work of Kirby, Robert Michael; Etiene, Tiago Etiene; Silvia, Claudio

Visualization is often employed as part of the simulation science pipeline. It is the window through which scientists examine their data for deriving new science, and the lens used to view modeling and discretization interactions within their simulations. We advocate that as a component of the simulation science pipeline, visualization itself must be explicitly considered as part of the Validation and Verification (V&V) process. In this talk, we define V&V in the context of computational science, discuss the role of V&V in the scientific process, and present arguments for the need for "verifiable visualization". Using paradigms expressed within the CS&E community, we will attempt to express what a common "V&V in V" language might look like (as a component of possible scientific foundations of visualization). We will then summarize three verification case studies applied to visualization: verification of geometric accuracy in the isosurface extraction process, verification of topological consistency in the isosurface extraction process, and verification of the volume rendering visualization pipeline. We will conclude with some lessons learned in our search for "verifiable visualizations".

3.25 YURT: YURT Ultimate Reality Theater – Why?

David H. Laidlaw (Brown University, US)

L I will motivate and describe a novel 3D stereoscopic display currently nearing completion at Brown University. The display will match many of the perceptual abilities of the human visual system, and so will be, in a sense, an "ultimate" display device. The properties include: one arc-minute of spatial resolution, stereo, 60 frames per second, and a field of view that covers over 90% of the sphere of all viewing directions. The one gap is over the back of the head of a viewer looking at the main wall. The geometry of the display is similar to a yurt, with a full surround curved main wall approximately 16 feet in diameter, a conical partial ceiling, and a fully back-projected floor. All projection is done via 69 stereo 1920x1080 120 Hz projectors. Imagery will be blended to create seamless imagery, a first example of which I will show. We have dubbed the display YURT, for "YURT Ultimate Reality Theater."

3.26 Attribute Space Analysis for Multivariate SciVis Data

Heike Leitte (Universität Heidelberg, DE)

The interactive combination of scivis and infovis techniques proved a valuable tool for multifield analysis in scientific visualization. Many features only become apparent in attribute space and can be readily visualized through linking and brushing. One major issue of this technique is that it assumes a clear separation of features in attribute space, which is commonly not the case for data originating from scientific simulations or measurements on continuous domains. This data often forms large, more or less homogeneous structures in attribute space that are difficult to project. In my talk, I will explore novel directions to describe multidimensional pointclouds that take the continuous nature of the data into account.

3.27 Understanding, Uncertainty and Predictive Analytics

Ross Maciejewski (ASU – Tempe, US)

A key analytical task across many domains is model building and exploration for predictive analysis. Data is collected, parsed and analyzed for relationships, and features are selected and mapped to estimate the response of a system under exploration. One hypothesis is that allowing for a visual exploration of the data and model being used for predictive analytics may enable users to better refine their predictions. In order to explore how predictions might be performed in such a manner, we have developed a visualization system for box office prediction. A user study focusing on social media data as a predictor for movie box-office success was then performed to explore methods of successful interaction and visualization to improve users' understanding of a predictive model.

3.28 The Application/Design Study Playbook

Georgeta Elisabeta Marai (University of Pittsburgh, US)

The Visualization field needs both Design Studies and Technique papers to maintain its vitality. A Design Study typically contributes a domain analysis; a design which is a novel combination of known techniques, developed in collaboration with the domain experts; an implementation of that design; user feedback; and a summary of the design lessons learned. These can be valuable contributions: for example, the domain analysis is a basis on which other, and presumably better, visualization tools can later be built. The analysis typically points out a new type of data, or a new set of requirements, or tasks, or "something" interesting for which the SciVis field does not have a ready solution. This talk will focus on the Design Study Playbook, an outline of what a Design Study should contain to be more relevant to the SciVis community.

3.29 keyvis.org: Visualization as Seen Through its Research Paper Keywords

Torsten Moeller (Universität Wien, AT)

We present the results of a comprehensive analysis of visualization paper keywords supplied for 4366 papers submitted to five main visualization conferences. We describe main keywords, topic areas, and 10-year historic trends from two datasets: (1) the standardized PCS taxonomy keywords in use for paper submissions for IEEE InfoVis, IEEE Vis-SciVis, IEEE VAST, EuroVis, and IEEE PacificVis since 2009 and (2) the author-chosen keywords for papers published in the IEEE Visualization conference series (now called IEEE VIS) since 2004. Our analysis of research topics in visualization can serve as a starting point to (a) help create a common vocabulary to improve communication among different visualization sub-groups, (b) facilitate the process of understanding differences and commonalities of the various research sub-fields in visualization, (c) provide an understanding of emerging new research trends, (d) facilitate the crucial step of finding the right reviewers for research submissions, and (e) it can eventually lead to a comprehensive taxonomy of visualization research. One additional tangible outcome of our work is an application that allows visualization researchers to easily browse the 2600+ keywords used for IEEE VIS papers during the past 10 years, aiming at more informed and, hence, more effective keyword selections for future visualization publications.

3.30 Visualization of Uncertainty: Measures other than Mean

Kristi Potter (University of Utah, US)

The visualization of uncertainty often uses mean and standard deviation to define and quantify uncertainty, however this measure is not always appropriate. In this talk I will present two different approaches to visualizing uncertainty. The first is an approach to comparing a collection of PDFs that does not assume a normal distribution and the other uses entropy, from the field of information theory, to express the uncertainty within the data.

3.31 Paving the Road for Data-Driven Visualization Models

Timo Ropinski (Linköping University, SE)

Visualization is often quoted as a key technology enabling our information- based society to cope with the big data challenge arising from inexpensive data acquisition and storage. However, despite this potential and more than three decades of computer-based visualization research, to a large extent it is not possible to predict the outcome and the benefits of a certain visualization algorithm applied to a particular data set. Even when considering a specific application domain or task, it cannot be determined beforehand to which extent an applied visualization technique will suffice. This is in particular remarkable, as methodologies for analyzing and predicting the complexity of algorithms are as old as computing theory itself. Based on the known model of computation, it is a standard process to perform an asymptotic analysis to investigate algorithm behavior. However, despite the advances in our understanding of the human visual system in the last decades, we still do not have sufficiently detailed computational models to predict the effectiveness of a visualization when viewed by a human observer. This observation is underlined when looking back 30 years, when David Marr initially proposed the idea to formulate computational models of visual processing. Despite this intriguing idea and great enthusiasm spanning several research communities, today we are still far away from such a unified computational model. While this clearly shows the difficulties related to computational models in this area, fortunately, today's information technology gives us the possibility to tackle this problem from a different perspective. In a similar way as modern biology is making sense of the complexity of living processes through high- throughput data acquisition technologies, the effectiveness of visualizations could be investigated by enabling large-scale empirical studies, which facilitate the acquisition of amounts of user data in a dimension that has not been possible before. To realize this acquisition and derive a better understanding of the value of particular visualizations, we have started to develop concepts and technologies to enable high-throughput visualization-centered empirical studies, which enable us and other researchers, to acquire massive amounts of visual response data from visualization users. To collect this data, we have developed a semi-automated study conduction interface, which is directly integrated into a visualization researcher's workflow. Through this interface researchers can initiate and analyze large- scale crowd-sourced user studies with minimal effort, and thus contribute to a massive collection of user response data. Based on the thus acquired data, we plan to develop data-driven visualization models, with the long term goal to support predictive visualization. In my talk I will present our goals as well as the current status of the project.

3.32 The Uncertainty Paradox in Visualization

Holger Theisel (Universität Magdeburg, DE)

Data sets in Scientific Visualization are large and continuously growing, their visualization – even without uncertainty – is challenging. Considering uncertainty, the amount of data even increases. The more complex we model uncertainty, the more data we have to visualize, and the more the data describing the uncertainty exceeds the actual amount of data itself.

Looking at the information to be processed, the situation may change: considering the uncertainty may lead to less information to be presented than in the certain data itself. This gives the potential chance to create uncertainty- scalable visualization techniques. For such techniques, considering uncertainty should simplify the visualization instead of making them more complicated. They should converge to classic visualization techniques for low uncertainty and extremely simple visualizations for high uncertainty. We discuss opportunities to come up with uncertainty- scalable visualization techniques.

3.33 Visualizing Spatio-Angular Fields

Amitabh Varshney (University of Maryland, College Park, US)

Joint work of Varshney, Amitabh; Bista, Sujal; Gullapalli, Rao; Zhuo, Jiachen

Spatio-angular fields are emerging as a new data type in several visualization applications. In this talk, I shall outline some of our initial results in managing, analyzing, and visualizing spatio-angular fields derived from Diffusion Kurtosis Imaging (DKI) as well as identify some future directions.

Diffusion kurtosis imaging has started being used in the medical imaging community as it can reveal subtle changes in both gray and white matter. It has shown promising results in studies on changes in gray matter and mild traumatic brain injuries, where DTI is often found to be inadequate. However, the highly detailed spatio-angular fields in DKI datasets present a special challenge for visualization. Traditional techniques that use glyphs are often inadequate for expressing subtle changes in the DKI fields. Here I shall outline some of our results into the study of the micro-structural properties of the brain.

3.34 Physics-Based Fluid Simulation Coupled to 4D MRI Blood Flow

Anna Vilanova Bartroli (TU Delft, NL)

Modern MRI measurements deliver volumetric and time-varying blood-flow data. Visual analysis of these data potentially leads to a better diagnosis and risk assessment of various cardiovascular diseases. Recent advances have improved the speed and quality of the imaging data considerably. Nevertheless, the data remains compromised by noise and a lack of spatio-temporal resolution. Besides imaging data, also numerical simulations are employed. These

are based on mathematical models of specific features of physical reality. However, these models are simplifications of the reality, and require parameters and boundary conditions. Data assimilation can bring measured data and physically-based simulation together, and benefit from both methodologies. Our first steps in this direction and the challenges of this approach will be presented.

3.35 Eye-Tracking Studies in Scientific Visualization

Daniel Weiskopf (Universität Stuttgart, DE)

In this talk, I discuss the role of eye-tracking in user studies in visualization research, in particular, for scientific visualization. We have been witnessing a trend toward the increasing use of experiments that assess user performance and user experience in scientific visualization. So far, however, much less work utilizes eye-tracking measurements in studies. I discuss possible roadblocks in the use of eye-tracking for scientific visualization research along with opportunities and directions for future empirical research.

3.36 Ensemble Visualization

Ruediger Westermann (TU München, DE)

To predict and quantify the uncertainty in numerical simulations of a physical phenomenon, multiple simulations are carried out using different physical or computational models, perturbed initial states or input parameter settings. This results in so-called ensembles, comprising members showing possible occurrences of the phenomenon. Ensemble visualization is of ever increasing relevance in a number of scientific domains, such as meteorology, fluid dynamics, or geology. It aims at identifying similarities and differences among ensemble members and revealing the sensitivity of these members to the input parameters. It further proposes means to track the space-time evolution of specific features and to identify where and how these features diverge in different ensemble members. There is a direct link between ensemble visualization and parameter space navigation: While the latter goes beyond ensemble visualization as it looks at the parameter space and its connection to the output, at the same time it relies upon ensemble visualization techniques to quantify the similarity between different outputs or between an output and a given reference, and thus to help the user controlling the parameter space navigation.

In "Future Challenges for Ensemble Visualization" by Harald Obermaier and Kenneth I. Joy it was recently proposed to classify ensemble visualization techniques into two categories: - Feature-based visualization extracts features from individual ensemble members and compares them across the ensemble. - Location-based visualization compares ensemble properties at fixed locations in the dataset.

Since feature-based visualization can also compare ensemble properties at fixed locations, I recommend renaming the second category into "Summary- based visualization", which

compares sample properties at fixed locations and also investigates the spatial relationships between the samples at different locations in the dataset.

Lastly I would like to bring the reader's attention to an interesting problem in ensemble visualization. A widely used ensemble visualization technique simply displays simultaneously in one image the feature present in all ensemble members. Spaghetti plots are an example. An alternative approach is to assume a stochastic uncertainty model, and to consider ensemble members as realizations of a multivariate random number exhibiting certain distributions. Given such a model, one can try to derive probabilities of feature occurrences from the multivariate random field. The problem is, that the field realizations are no longer consistent with the underlying (physical) models used to generate the ensemble members. For instance, let us assume an ensemble comprising incompressible flow fields, and a multivariate vector valued random variable exhibiting a distribution derived from this ensemble. One particular realization of the random variable will not necessarily satisfy the compressibility constraint, resulting in a flow field which would never occur this way in a direct simulation. In my opinion, this mismatch between stochastic uncertainty modeling and the real outcomes of the data generation processes needs some further investigations, in particular regarding the consideration of correlations in the data to further constrain the stochastic data generation process.

3.37 Why you should (probably) not be doing "uncertainty visualization"

Ross Whitaker (University of Utah, US)

The visualization of stochastic or uncertain data is typically referred to "uncertainty visualization". However, this terminology implies associated set of assumptions about the paradigm for visualization, which is typically to display an answer that has been modulated or augmented by an associated uncertainty. This however, asserts the existence of a renderable answer, which defies one of the underlying goals or principles of visualization, which is the exploration of data to obtain a holistic understanding or to discover properties that have no associated, a priori hypothesis. An alternative paradigm, is "variability visualization" where the goal of the visualization is to explore or better understand the set of possible outcomes, or the probability distribution, associated with a set of data. One example of such an approach is the method of contour boxplots, which relies on a generalization of data depth, from descriptive statistics, to render the variability of solutions in an ensemble of isocountours.

3.38 Science Portal – Scientific Visualization in the Public Space

Anders Ynnerman (Linköping University, SE)

This talk will show how medical volume visualization can be used in knowledge dissemination in public spaces and discuss the specific challenges posed when working with museum curators and producers to develop robust exhibits based on state-of-the-art visualization techniques.

The main technology used is interactive multi-touch tables which allow visitors to science centers and museums to interactively and intuitively explore the normally invisible and learn about the inside workings of the human body, exotic animals, natural history subjects or even mummies, such as a recent installation at the British Museum. The talk will apart from showing a few interesting examples discuss requirements on the production pipeline from discovery to gallery – The Science Portal.

3.39 Exploring Multivariate and Ensemble Data

Xiaoru Yuan (Peking University, CN)

Understanding multivariate and ensemble data is very challenging due to the complexity of the data nature. The increasing data size further pose serious constrain to visualization solutions on exploring such data sets. We will discuss a few approaches we are trying to handle such problems.

4 Working Groups

4.1 Scientific Foundation of Visualization

A common theoretical and perceptual foundation supports the research across a broad range of types of visualization and application domains. This foundation includes the essential nature of visualization, models of how visualization works, fundamental mathematics, and best practices in design, collaboration, and application. Because of the rather large size of this group, three separate subgroups were formed to explore different aspects of the foundations of visualization. These groups discussed issues of perception and cognition, technology to support visualization, and visualization education. The reports of the discussions of each of these subgroups are below.

Perception and Cognition: The discussion of the perception and cognition subgroup addressed the issues of transferable knowledge from the psychology literature, difficulties of transferring knowledge between domains, approaches for interfacing with the perception community, motivation for collaboration with the perception community, metrics for measuring the quality of a visualization, and resources for supporting researchers in conducting and reporting user studies. Specific questions of interest in transferring perceptual knowledge to visualization research include how do we perceive time in moving animations?, how should color maps and shading be combined?, and how can depth perception be used to design better visualizations? The group identified three grand challenges on the topic of perception and cognition: how to optimize time in the understanding process, how to transfer knowledge from psychology, and why visualization works. The group decided to follow up with a panel, proposed and accepted, on the topic of "New Perceptual and Cognitive Issues for Visualization" that will appear at the 2014 IEEE VIS conference in Paris, France and an article for submission to IEEE CG&A Visualization Viewpoints.

Technology: The technology subgroup concentrated on identifying knowledge and tools from other disciplines of significant relevance to visualization. The discussion addressed the importance of design principles, perceptual mechanisms including Gestalt laws and pre-attentive processing, interaction design, tools such as Design Studio and Tableau, story-telling, acceleration data structures, and application areas and case studies. Algorithms and data analysis techniques of relevance include text mining, feature extraction, 2D geometry, the rendering pipeline, and pre-processing using sampling, interpolation, and filtering.

Education: The discussion of the education subgroup worked to identify knowledge elements that every student should know upon exiting a visualization course. These elements include characteristics of discrete and continuous data, the basics of numerics and signals, fundamentals of the image generation pipeline, basics of perception, classes of data and corresponding methods, data structures, and a core set of display methods. Other possible course components include interaction principles, collaboration skills, visualization toolkits, visual design principles, volume rendering and transfer functions, and evaluation. The group decided to collect examples of syllabi from visualization courses in order to create a repository, as well as provide material to conduct an analysis of commonalities and differences across courses.

4.2 Uncertainty Visualization

The goal of visualization is to effectively and accurately communicate data. Visualization research has often overlooked the errors and uncertainty which accompany the scientific process and describe key characteristics used to fully understand the data. The lack of these representations can be attributed, in part, to the inherent difficulty in defining, characterizing, and controlling this uncertainty, and in part, to the difficulty in including additional visual metaphors in a well designed, potent display. However, the exclusion of this information cripples the use of visualization as a decision making tool due to the fact that the display is no longer a true representation of the data. This systematic omission of uncertainty commands fundamental research within the visualization community to address, integrate, and expect uncertainty information.

The breakout group had a lively discussion of fundamental research issues. Uncertainly in visualization has many forms from parameter space analysis to ensemble visualization to the visualization of uncertainty. In parameter space analysis, sensitivity is one aspect of uncertainty but uncertainty connecting the inputs and outputs is also key. One method to quantify uncertainty is multiobjective optimization coupled with the parameter spaces. In ensemble visualization, uncertainty can come from the sensitivity of ensembles of data that capture the main uses, e. g. elections, weather, climate, etc. Having the ability to visualize the uncertainty aids in risk decision making by understanding the set of possible realities. In the statistics community, there are several types of statistics such as inferential, explanatory, descriptive, or predictive. The group discusses whether uncertainty in visualization capture these concerns as well.

It is our job in visualization to understand uncertainties in the applications and use visualization to present, explain and analyze for risk management and deeper fundamental understanding of the applications. This can be accomplished by quantifying, representing, and displaying the uncertainties in the visualization pipeline in addition to the application.

The group agreed that there are many avenues for future research in uncertainty in

visualization and how to best utilize, present and analyze that information. In addition to the concepts outlined in the discussion, many areas of future research were not formally represented by the participants and also form the basis for important future research problems.

4.3 Integrated Multi-field Visualization

This working group consisted of 18 Dagstuhl attendees. This section of the report was written based on Ross Maciejewski's minutes for the two meetings held by the group. The group first reviewed the work by the working group on multi-field visualization in Dagstuhl 2011 seminar, and in particular Part II of the Springer book resulting from Dagstuhl 2011. The group then identified and discussed interests and challenges in the area of multi-field visualization. These included the followings:

- The increasing challenge as our scientific data is becoming richer with different types of fields and multiple-fields of the same kind, making it harder to explore high-dimensional multi-field time series data.
- Mathematical aspects of multi-field data and their need to be informed by applications.
- The challenge of integrating multiple data soruces for the same phenomen, given the limited bandwidth of different visual channels in a single view and the changing validity over time.
- Integrating the different components of space-weather simulation data with 21 different models (multi-scale and multi-modal data) into a consistent view.
- Comparing and correlating data from different sources in the domain of space engineering.
- Understanding the multiple fields making up an underlying model in physics applications.
- The image analysis of bio-medical video data, which often result in many attribute fields.
- The challenge of how to visualize, characterize and compare datasets with an increasing number of variables per point in a volume.
- Developing the capabilities to show the interaction between different variables in order to help domain experts understand how vector fields are changed by tensor fields.
- Building a unified framework for both measured and derived data in order to support the data exploration process in multi-field visualization.
- The integration of spatial and non-spatial information to support the collaboration between structural and non-structural biologists.
- The topological point of view, e. g., looking at persistent homology on two fields (let alone three) is extremely difficult.
- Showing the interplay of multi-variables in order to enable the creation of models that can explain these interplays and interaction, which may be difficult (or impossible) to explain mathematically.
- Visualizing complex geometry and the overlap between geometric structures.
- Particle based simulation where the simulation results are aggregated to create a time-varying field, as well as how this interplays with another field, e. g., comparing physical quantities between fields.
- The challenge of extending basic knowledge about scalar and vector field topology to multi-field topology for use in applications including material sciences and manufacturing.
- Developing new interdisciplinary tools for subspace exploration that would allow us to identify a good subspace that extracts a maximal amount of information using methods beyond topology.

The group then discussed various possible products that the group or different sub-groups (teams) may be able to deliver collaboratively. The options include (a) a viewpoint article that typically involves six to eight people; (b) the visualization corner that could be done by two to three people; (c) writing a survey paper on that topic; (d) producing an edited book on multi-field visualization; (e) writing a small book on a specific aspect of multi-field visualization; and (f) organizing a tutorial or workshop on multi-field visualization. A number of specific planned actions were identified, and group members volunteered to take a lead to coordinate actions or to organize a team to take the action forward. These specific plans include:

- A CG&A viewpoint article that may select a number of interesting examples of multi-field data (e. g., medical, material, biology, geography) and define the challenges involved in such data. **Action:** This effort will be coordinated by Anders Ynnerman.
- A workshop on multi-field visualization. As TopoInVis is more mathematically-focused, the meeting concluded that we could organize a workshop in conjunction with EuroVis 2015, targeting four page short papers. **Action:** Ingrid Hotz will take the lead in forming a workshop organization team.
- A tutorial and/or book, the former in conjunction with his ongoing grant, for the latter in conjunction with a book series led by Tamara Munzner. **Action:** Hamish Carr will coordinate this effort.

4.4 Environmental Scientific Visualization

Environmental visualization refers to a collection of visualization applications that deal with captured and simulated data in climate research, atmospheric and environmental sciences, earth science, geophysics and seismic research, oceanography, and the energy industry (e. g., oil, gas and renewable energy). Research in these applications has a huge impact on mankind, and typically faces serious challenges of data deluge (e. g., very large volumes of multi-spectral satellite images, large data collections from different sensor types, ensemble computation of very large simulation models, multi-model and multi-physics simulations, scattered, time-varying, multi-modal data in seismic research). Scientific progress in the areas of the environment and sustainability is critical in the solution of global problems and scientific visualization has great potential to support this progress.

The breakout group consisted of 14 seminar participants with interest in this broad application domain. In a first step, we discussed current application domains that the group is actively looking at. These areas are climate research including atmosphere, ice, ocean, and land modeling; geology, geography, and geophysics; soil and groundwater research; energy resources and waste management; land use research; biodiversity and ecosystem services; urban data; and finally disaster research with a focus on Tsunami modeling and earthquake modeling. Obviously, this broad agenda requires a lot of different visualization methodologies and opens a wide range of challenges and possibilities, even without looking at the various research challenges in sciences concerned with environmental questions outside this concrete list.

In a second step, our group explained to each other the specific problems that the members are working on. This allowed everyone to get a better understanding of the term "environmental visualization" from a practical point of view. Furthermore, we used it to derive specific visualization challenges that appear in many, if not most of, these applications. Seminar participants were looking into the visualization of fly-over data from airplanes;

geological data from petroleum engineering companies; biological growth monitoring data; species monitoring data; city traffic data and related air pollution data; data on the water, light, and energy consumption of a specific region; earthquake and Tsunami modeling data; and groundwater simulation data of a US state.

We have identified the following main challenges for visualization:

- Visualization in environmental sciences has to address very diverse audiences. First, there are the experts, i. e. scientists and engineers, as in many scientific visualization applications. Second, there is the general public with a high interest in the topic but substantially less prior knowledge on the subject. And third, there are the policy makers who use visualization as a basis for decisions, looking for information on the necessity of decisions and on information about the consequences of decisions. There is a growing demand by the researchers and engineers for the automatic generation of larger reports on a specific environmental question, using updated observations, new measurements or new simulations. Visualization is also asked as a method to narrow the gap between the scientists and policy makers. Regarding the policy makers, we identified a big gap between the visual needs of this group and typical scientific visualization methods. We foresee a large demand for research in this direction.
- As a more specific challenge, we identified the necessity to separate the visual communication of scientific facts from the presentation of derived decisions or recommendations.
- We have also noted that most examples in the group show that one has to solve a serious data integration task prior to visualization.

Overall, the group agreed that environmental visualization is still in its infancy, and will get a lot more attention in the near future because of the high demand in society with respect to the underlying big questions.

Participants

- James Ahrens
Los Alamos National Lab., US
- Georges-Pierre Bonneau
INRIA Rhône-Alpes, FR
- Rita Borgo
Swansea University, GB
- Nadia Boukhelifa
University Paris-Sud – Gif sur Yvette, FR
- Hamish Carr
University of Leeds, GB
- Remco Chang
Tufts University, US
- Jian Chen
University of Maryland, US
- Min Chen
University of Oxford, GB
- Hank Childs
University of Oregon, US
- João Luiz Dihl Comba
Federal University of Rio Grande do Sul, BR
- Leila De Floriani
University of Genova, IT
- Thomas Ertl
Universität Stuttgart, DE
- Issei Fujishiro
Keio University, JP
- Christoph Garth
TU Kaiserslautern, DE
- Andreas Gerndt
DLR – Braunschweig, DE
- Michael Gleicher
University of Wisconsin – Madison, US
- Eduard Gröller
TU Wien, AT
- Hans Hagen
TU Kaiserslautern, DE
- Charles D. Hansen
University of Utah, US
- Helwig Hauser
University of Bergen, NO
- Hans-Christian Hege
Konrad-Zuse-Zentrum – Berlin, DE
- Mario Hlawitschka
Universität Leipzig, DE
- Ingrid Hotz
DLR – Braunschweig, DE
- Christopher R. Johnson
University of Utah, US
- Kenneth Joy
Univ. of California – Davis, US
- Robert Michael Kirby
University of Utah, US
- David H. Laidlaw
Brown University, US
- Heike Leitte
Universität Heidelberg, DE
- Ross Maciejewski
ASU – Tempe, US
- Georgeta Elisabeta Marai
University of Pittsburgh, US
- Torsten Möller
Universität Wien, AT
- Kristi Potter
University of Utah, US
- Penny Rheingans
University of Maryland, US
- Timo Ropinski
Linköping University, SE
- Gerik Scheuermann
Universität Leipzig, DE
- Claudio T. Silva
New York University, US
- Holger Theisel
Universität Magdeburg, DE
- Amitabh Varshney
University of Maryland, US
- Anna Vilanova Bartroli
TU Delft, NL
- Daniel Weiskopf
Universität Stuttgart, DE
- Rüdiger Westermann
TU München, DE
- Ross Whitaker
University of Utah, US
- Anders Ynnerman
Linköping University, SE
- Xiaoru Yuan
Peking University, CN

Report from Dagstuhl Seminar 14232

Design and Synthesis from Components

Edited by

Jakob Rehof[1] and Moshe Y. Vardi[2]

1 **TU Dortmund, DE,** jakob.rehof@cs.tu-dortmund.de
2 **Rice University, US,** vardi@cs.rice.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14232 "Design and Synthesis from Components" which took place from June 1st to June 6th, 2014. The seminar aimed at bringing together researchers from the component-oriented design community, researchers working on interface theories, and researchers working in synthesis, in order to explore the use of component- and interface design in program synthesis. The seminar program consisted of 6 tutorial talks (1 hour) and 16 contributed talks (30 mins) as well as joint discussion sessions. This report documents the abstracts of the talks as well as summaries of discussion sessions.

Seminar June 1–6, 201401 – http://www.dagstuhl.de/14232
1998 ACM Subject Classification I.2.2 Automatic Programming – Program synthesis, D.2.2 Design Tools and Techniques, F.3.1 Specifying and Verifying and Reasoning about Programs
Keywords and phrases Component design, Component-based synthesis
Digital Object Identifier 10.4230/DagRep.4.6.29
Edited in cooperation with Dror Fried

1 Executive Summary

Jakob Rehof
Moshe Y. Vardi

The purpose of the seminar was bringing together researchers from the component-oriented design community, researchers working on interface theories, and researchers working in synthesis, in order to explore the use of component- and interface design in program synthesis.

The seminar proposal was motivated by a recently developing trend in component-based synthesis, which is seen both as creating a need and providing the potential for a cross-community effort. Traditionally, synthesis has been pursued in two distinct and somewhat independent technical approaches. In one approach, synthesis is characterized by temporal logic and automata theoretic methods, whereas in the other synthesis is characterized by deductive methods in program logics and in type theory considered under the Curry-Howard isomorphism. Recent work in component-oriented design has spurred the idea of *component-based synthesis*, where systems are synthesized relative to a given collection (library, repository) of components, within both technical approaches. Recent results in both communities show that this development allows the two communities to communicate more intensely on the common ground of component-orientation to their mutual benefit. The trend opens the door to a new attack on the great challenges of synthesis (including computational complexity and complexity of specification) by exploiting component design.

Design and Synthesis from Components, *Dagstuhl Reports*, Vol. 4, Issue 6, pp. 29–47
Editors: Jakob Rehof and Moshe Y. Vardi
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The seminar program consisted of 6 tutorial talks (1 hour) and 16 contributed talks (30 mins) as well as joint discussion sessions. Two slots for joint discussions were pre-planned for each day but were used flexibly and dynamically, depending on the development of discussions and reactions to the talks. It was felt that the mixture of tutorials, talks and joint discussion slots turned out to be an altogether very good instrument for making intensive exchanges among all seminar participants possible. It seems to be the general impression that the seminar was very succesful in meeting the challenge of bringing together researchers from quite a diverse range of technical fields, spanning from software engineering to mathematical logic. The seminar was succesful in generating several concrete cross-community collaboration projects which would not have been likely to have come into existence by way of traditional conferences.

Joint discussions were summarized by Dror Fried (Rice University) who is gratefully acknowledged for undertaking the role of "seminar collector".

2 Table of Contents

3 Overview of Talks

3.1 Coordinated Composition of Components

Farhad Arbab (CWI – Amsterdam, NL)

Modeling components as units of behavior offers a rich framework where composition operators can coordinate the behavior of such constituents into the behavior of arbitrarily more complex systems. Our work on Reo, its semantics, and tools serves as a concrete instance of such a framework. The emphasis in Reo is on the externally observable behavior of components and their coordinated composition into more complex concurrent systems. This emphasis highlights the eminent role of protocols in concurrent systems of components and services, and makes concurrency protocols the central focus in Reo.

At its core, Reo proffers an interaction-based model of concurrency where more complex protocols result from composition of simpler, and eventually primitive, protocols. In Reo, combining a small set of user-defined synchronous and asynchronous primitives, in a manner that resembles construction of electronic circuits from gates and elements, yields arbitrarily complex concurrency protocols. Semantics of Reo preserves synchrony and exclusion through composition. This form of compositionality makes specification of protocols in Reo simpler than in conventional models and languages, which offer low-level synchronization constructs (e.g., locks, semaphores, monitors, synchronous methods). Moreover, the high-level constructs and abstractions in Reo also leave more room for compilers to perform novel optimizations in mapping protocol specifications to lower-level instructions that implement them. In on- going work we currently develop code generators that produce executables whose performance and scalability on multi-core platforms compare favorably with hand-crafted, hand-optimized code.

3.2 Synthesis for Communication-centred Programming

Mariangiola Dezani (University of Turin, IT)

Joint work of Coppo, Mario; Dezani, Mariangiola; Venneri, Betti
Main reference M. Coppo, M. Dezani-Ciancaglini, B. Venneru, "Self-Adaptive Monitors for Multiparty Sessions," in Proc. of the 22nd Euromicro Int'l Conf. on Parallel, Distributed, and Network-Based Processing (PDP'14), pp. 688–696, IEEE, 2014; pre-print available from author's webpage.
URL http://dx.doi.org/10.1109/PDP.2014.18
URL http://www.di.unito.it/~dezani/papers/cdv14.pdf

The increasing number of heterogeneous devices interacting in networks claims for a new programming style, usually called communication-centered programming. In this scenario possible participants to choreographies expose their interfaces, describing the communications they offer. Interaction protocols can be synthesised through a phase of negotiation between participants, in which different pieces of code can be composed in order to get the desired behaviour. At run time some unexpected event can make the current choreography no longer executable. In this case the participants should be able to adapt themselves in order to successfully continue the interaction. In this adaptation both new interfaces and new codes of participants could need to be synthesised.

3.3 ArchiType: Automatic Synthesis of Component & Connector-Software Architectures with Bounded Combinatory Logic

Boris Duedder (TU Dortmund, DE)

Joint work of Duedder, Boris; Moritz, Martens; Rehof, Jakob

Combinatory logic synthesis is a new type-based approach towards automatic synthesis of software from components in a repository. In this talk we demonstrate how the type-based approach can naturally be used to exploit taxonomic conceptual structures in software architectures and component repositories to enable automatic composition and configuration of components, and also code generation, by associating taxonomic concepts to architectural building blocks such as, in particular, software connectors. Components of a repository are exposed for synthesis as typed combinators, where intersection types are used to represent concepts that the specify intended usage and functionality of a component. An algorithm for solving the type inhabitation problem in combinatory logic – does there exist a composition of combinators with a given type? – is then used to automate the retrieval, composition, and configuration of suitable building blocks with respect to a goal specification. Since type inhabitation has high computational complexity, heuristic optimizations for the inhabitation algorithm are essential for making the approach practical. We discuss particularly important (theoretical and pragmatic) optimization strategies and evaluate them by experiments. Furthermore, we apply this synthesis approach to define a method for software connector synthesis for realistic software architectures based on a type theoretic model. We conduct experiments with a rapid prototyping tool that employs this method on complex concrete ERP- and e-Commerce- systems and discuss some results.

3.4 Petri Games: Synthesis of Distributed Systems with Causal Memory

Bernd Finkbeiner (Universität des Saarlandes, DE)

Joint work of Finkbeiner, Bernd; Olderog; Ernst-Rüdiger

We introduce Petri games as a new foundation for the synthesis of distributed systems. The players of a Petri game consist of the system processes and the external environment, all represented as tokens on a Petri net. The players memorize their causal history and communicate it to each other during each synchronization.

Petri games lead to new decidability results and algorithms for the synthesis of distributed systems. Unlike the classic approaches, which are based on a fixed ordering of the relative informedness of the processes, we can synthesize systems with dynamically changing information flow, as in client-server protocols, where the information source moves back and forth between client and server, or in token ring protocols, where the identity of the sender changes as the token moves.

3.5 Towards Improving Extensibility and Reuse of Modules Within a Product Line

George T. Heineman (Worcester Polytechnic Institute, US)

The principle of modularity is the basis for both object-oriented (OOD) and component-based design (CBD) but there is a tension between extensibility and reuse. OOD often leads to rich frame-works that enable new classes to be designed as extensions to existing classes but there is little reuse of individual classes out-side of the framework. CBD often leads to third party assembly through well-designed interfaces but it is often impossible to extend components. The Model View Controller (MVC) paradigm bridges these two domains because it can be described as both a Design Pattern (in the OOD realm) and an Architectural Pattern (in the CBD realm). While MVC supports rich extensibility in both models and views, it is striking that it seems to naturally lead to controllers that cannot be reused or extended. We combine the MVC paradigm with feature-oriented programming (FOP) to bring reusability back to controllers. We demonstrate the effectiveness of our approach using a product- line example of a solitaire game engine.

3.6 On coercion synthesis for regular expressions

Fritz Henglein (University of Copenhagen, DK)

Regular expressions (REs) are usually interpreted as languages. This is, however, an inadequate theoretical basis for programming applications where parsing, extracting and transforming data, not just membership testing, is required.

REs can be interpreted more intensionally as types, each representing a set of parse trees, such that establishing language containment corresponds to finding some coercion: a function transforming parse trees according to one RE to parse trees in the other RE without changing the underlying string. Coercions can be found by constructive interpretation of axiomatizations of regular expression containment.

This puts the particular axiomatization at center stage: We are not only interested in determining whether or not some coercion exists (whether or not a particular RE containment holds), but actually synthesizing its actual code; furthermore, since the synthesized coercions are actually executed, finding (in particular: not ruling out in the axiomatization) coercions with good computational properties – e.g. streaming execution on a bit-serialized representation of syntax trees and always running in worst-case linear time – is important.

The basic questions are then: How can one efficiently synthesize coercions, which themselves need to be efficient, by proof search in an axiomatization of regular expression containment? More basically, what is a good axiomatization for doing this? And why is regular expression containment an interesting synthesis case study for synthesis? We present preliminary results and some thoughts on these questions.

3.7 Application-layer Connector Synthesis

Paola Inverardi (University of L'Aquila, IT)

Joint work of Inverardi, Paola; Massimo, Tivoli; Romina, Spalazzese; Marco, Autili
Main reference P. Inverardi, M. Tivoli, "Automatic synthesis of modular connectors via composition of protocol mediation patterns," in Proc. of the 2013 Int'l Conf. on Software Engineering (ICSE'13), pp. 3–12, IEEE/ACM, 2013.
URL http://dl.acm.org/citation.cfm?id=2486790

The heterogeneity characterizing the systems populating the Ubiquitous Computing environment prevents their seamless interoperability. Heterogeneous protocols may be willing to cooperate in order to reach some common goal even though they meet dynamically and do not have a priori knowledge of each other. Despite numerous efforts have been done in the literature, the automated and run-time interoperability is still an open challenge for such environment. We consider interoperability as the ability for two Networked Systems (NSs) to communicate and correctly coordinate to achieve their goal(s). In this tutorial, I report the main outcomes of our past and recent research on automatically achieving protocol interoperability via connector synthesis. We consider application-layer connectors by referring to two conceptually distinct notions of connector: coordinator and mediator. The former is used when the NSs to be connected are already able to communicate but they need to be specifically coordinated in order to reach their goal(s). The latter goes a step forward representing a solution for both achieving correct coordination and enabling communication between highly heterogeneous NSs. In the past, most of the works in the literature described efforts to the automatic synthesis of coordinators while, in recent years the focus moved also to the automatic synthesis of mediators. Within the Connect project, by considering our past experience on automatic coordinator synthesis as a baseline, we propose a formal theory of mediators and a related method for automatically eliciting a way for the protocols to interoperate. The solution we propose is the automated synthesis of emerging mediating connectors (i.e., mediators for short).

3.8 Towards Understanding Superlinear Speedup by Distillation

Neil D. Jones (University of Copenhagen, DK)

Joint work of Jones, Neil D.; Hamilton, Geoff W.

Distillation is a transformation method that can yield superlinear program speedups – a feat beyond earlier fully automatic program transformations such as partial evaluation or supercompilation. Bisimulation is a key to correctness of distillation, i.e., that it preserves semantics.

Relation to Dagstuhl 14232: an optimiser synthesises an efficient program from a less efficient one. A first question: In what sense can equivalent programs with asymptotically different runtimes be bisimilar?

The talk describes current work on such questions, partly theoretical and partly computer experiments, on some "old chestnut" programs well-known from program transformation literature (naive reverse, factorial sum, Fibonacci, and palindrome detection).

Using complexity-theoretic tools, we see that a sizable class of first-order exponential-time programs can be converted into second-order polynomial-time equivalents. The effect is to trade time for space, in effect replacing CONS or a Turing machine tape by first-order functions as arguments in a CONS-free program. Finally, a conjecture: that distillation can automatically realise many such superlinear speedups.

3.9 Synthesis of Reactive Systems Components with Data

Bengt Jonsson (Uppsala University, SE)

We consider automated synthesis of reactive components. Synthesis of reactive components is emerging to solve many tasks in software development, in embedded systems, for device drivers, for protocol converters, in service composition, etc. For the synthesis of finite-state components from finite-state specifications, there is currently an elaborate body of theory, which typically performs synthesis by constructing a winning strategy in a two-player game. In this presentation, we consider to extend synthesis of reactive components to take into accoun also data from potentially unbounded domains . Data from potentially unbounded domains are a natural ingredient in the specification and synthesis of many classes of systems. For example, a protocol mediator must transfer messages and data items correctly between the mediated components, a device driver must deliver the right data in the expected order.

We present techniques for synthesizing reactive components, and show how they can be used to automatically synthesize "missing" glue components in systems of communicating components: Given specification of a collection components, and a specification of the overall system, we can synthesize a most general glue component to satisfy the specification. We also show how the problem of finding a suitable system specification can be automated to a large degree. The techniques are natural when involved component specifications are deterministic. For the nondeterministic case, the synthesis problem is undecidable, and we consider how to restrict the solution space to obtain a decidable synthesis problem.

3.10 Application of Combinatory Logic Synthesizer in Robotics

Moritz Martens (TU Dortmund, DE)

Joint work of Martens, Moritz; Düdder, Boris; Rehof, Jakob

We present an application of Combinatory Logic Synthesizer, a type based synthesis tool, to the synthesis of workflows. The approach is illustrated by means of synthesis of workflows to control LegoNXT robots.

3.11 Towards Synthesis of Uniform Strategy against Temporal Epistemic Logic

Hongyang Qu (University of Sheffield, GB)

Joint work of Busard, Simon; Pecheur, Charles; Qu, Hongyang; Raimondi, Franco
Main reference S. Busard, C. Pecheur, H. Qu, F. Raimondi, "Reasoning about Strategies under Partial Observability and Fairness Constraints," in Proc. of the 1st Int'l Workshop on Strategic Reasoning (SR'13), EPTCS, Vol. 112, pp. 71–79, 2013.
URL http://dx.doi.org/10.4204/EPTCS.112.12

In this talk, I will first give an introduction of Interpreted Systems (IS) and epistemic logic. The former provides a formal semantics for modelling multi- agent systems, and the latter specifies knowledge based properties, which are often combined with temporal logics, such CTL, LTL and ATL. The second part of the talk will focus on difficulties in searching for a uniform strategy for temporal epistemic logic formulas. I will present two semantics for uniform strategy, each of which requires different solutions.

References

1 Alessio Lomuscio, Franco Raimondi, Model checking knowledge, strategies, and games in multi-agent systems, in: 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), ACM, 2006, pp. 161–168.

3.12 Beyond Two-player Zero-sum Games: Motivations and Highlights of Recent Results

Jean-François Raskin (Université Libre de Bruxelles, BE)

Two-player zero-sum games played on graphs is the classical setting for studying the reactive synthesis problem. In this talk, I will report on recent work directions that consider richer settings. To illustrate those new research directions, I will present informally with the help of examples the results contained in three recent publications whose abstract are reproduced below:

1. Krishnendu Chatterjee, Laurent Doyen, Emmanuel Filiot, Jean-François Raskin. *Doomsday Equilibria for Omega-Regular Games.* VMCAI 2014, LNCS, Vol. 8318, pp. 78–97. http://dx.doi.org/10.1007/978-3-642-54013-4_5

   Two-player games on graphs provide the theoretical framework for many important problems such as reactive synthesis. While the traditional study of two-player zero-sum games has been extended to multi-player games with several notions of equilibria, they are decidable only for perfect-information games, whereas several applications require imperfect-information games. In this paper we propose a new notion of equilibria, called doomsday equilibria, which is a strategy profile such that all players satisfy their own objective, and if any coalition of players deviates and violates even one of the players objective, then the objective of every player is violated. We present algorithms and complexity results for deciding the existence of doomsday equilibria for various classes

of omega-regular objectives, both for imperfect-information games, and for perfect-information games.We provide optimal complexity bounds for imperfect-information games, and in most cases for perfect-information games.

2. Veronique Bruyere, Emmanuel Filiot, Mickael Randour, Jean-François Raskin. *Meet Your Expectations With Guarantees: Beyond Worst-Case Synthesis in Quantitative Games.* STACS 2014, LIPIcs, Vol. 25, pp. 199–213. http://dx.doi.org/10.4230/LIPIcs.STACS.2014.199

   Classical analysis of two-player quantitative games involves an adversary (modeling the environment of the system) which is purely antagonistic and asks for strict guarantees while Markov decision processes model systems facing a purely randomized environment: the aim is then to optimize the expected payoff, with no guarantee on individual outcomes. We introduce the beyond worst-case synthesis problem, which is to construct strategies that guarantee some quantitative requirement in the worst-case while providing an higher expected value against a particular stochastic model of the environment given as input. We consider both the mean-payoff value problem and the shortest path problem. In both cases, we show how to decide the existence of finite-memory strategies satisfying the problem and how to synthesize one if one exists. We establish algorithms and we study complexity bounds and memory requirements.

3. Romain Brenguier, Jean-François Raskin, Mathieu Sassolas. *The Complexity of Admissibility in Omega-Regular Games.* CoRR abs/1304.1682(2013) – to appear in LICS'14. http://arxiv.org/abs/1304.1682v3

   Iterated admissibility is a well-known and important concept in classical game theory, e.g. to determine rational behaviors in multi-player matrix games. As recently shown by Berwanger, this concept can be soundly extended to infinite games played on graphs with omega-regular objectives. In this paper, we study the algorithmic properties of this concept for such games. We settle the exact complexity of natural decision problems on the set of strategies that survive iterated elimination of dominated strategies. As a byproduct of our construction, we obtain automata which recognize all the possible outcomes of such strategies.

3.13 Combinatory Logic Synthesis

Jakob Rehof (TU Dortmund, DE)

Joint work of Düdder, Boris; Martens, Moritz; Urzyczyn, Paweł

Combinatory logic synthesis has been proposed recently as a type-theoretic research programme in automatic synthesis of compositions from collections of components. Composition synthesis is based on the idea that the inhabitation (provability) relation in combinatory logic can be used as a logical foundation for synthesizing expressions satisfying a goal type (specification) relative to a given collection of components exposed as a combinatory type environment.

It is shown that, under the semantics of relativized inhabitation, already simple types are a Turing-complete logic programming language for computing (synthesizing) compositions, where collections of combinatory types are programs and types are rules in such programs. In order to enhance the ability to express semantic specifications we introduce intersection types

into composition synthesis, and we survey recent results on expressive power, algorithmics, and complexity. It is shown that modal types lead to a natural foundation for introducing meta- programming combinators, resulting in a highly flexible framework for composition synthesis. Based on a prototype implementation of the framework (CL)S, Combinatory Logic Synthesizer, we illustrate with practical examples as time permits.

3.14 In search for a strategy to search for a strategy

Sven Schewe (University of Liverpool, GB)

Joint work of John Fearnley, Doron Peled, Schewe, Sven
Main reference J. Fearnley, D. Peled, S. Schewe, "Synthesis of Succinct Systems," in Proc. of the 10th Int'l Symp. on Automated Technology for Verification and Analysis (ATVA'12), LNCS, Vol. 7561, pp. 208–222, Springer, 2012.
URL http://dx.doi.org/10.1007/978-3-642-33386-6_18

In synthesis, we seek a strategy to control a system to satisfy its objectives. But how do we search? Much research has been done to establish that the problem is hard, and just how hard it is. While this has been used to argue against synthesis, synthesis algorithms exists, but currently those with two arms and two legs have the upper hand. I would like to wonder with you if arms and legs are really necessary for synthesis, or if we can search for a search strategy.

References

1 B. Finkbeiner and S. Schewe. Uniform distributed synthesis. In *Proc. of IEEE LICS 2005*, pages 321–330. IEEE Computer Society Press.
2 B. Finkbeiner and S. Schewe. Bounded synthesis. *International Journal on Software Tools for Technology Transfer*, 15(5-6):519–539, 2013.
3 J. Fearnley, D. Peled, and S. Schewe. Synthesis of Succinct Systems. *Proc. of ATVA 2012*, pages 42–56.

3.15 Workflow Synthesis – Concepts and Experience

Bernhard Steffen (TU Dortmund, DE)

Joint work of Steffen, Bernhard; Tiziana Margaria, Johannes Neubauer, Stefan Naujokat
Main reference T. Margaria, D. Meyer, C. Kubczak, M. Isberner, B. Steffen, "Synthesizing Semantic Web Service Compositions with jMosel and Golog," in 8th Int'l Semantic Web Conf. (ISWC'09), LNCS, Vol. 5823, pp. 392–407, Springer, 2009.
URL http://dx.doi.org/10.1007/978-3-642-04930-9_25

The marriage of component-based design and software synthesis is promising, in particular in special scenarios like workflow-design where where components can often nicely be regarded as abstract functions. In this very much service-oriented setting linear time synthesis of chains of interactive activities and automatic processing steps leads to a completely new way of organization and management, which allows one to much better adapt to changing situations simply via appropriate 'replanning'. Based on an adequate ontology-based domain modelling this can be done in a very situation and process-aware fashion which may in particular also take legal procedural constraints into account.

Another interesting application domain are scientific workflows, where chains of analysis steps need to be arranged in a tailored fashion.

In both cases, linear time synthesis leads to a 'development' style characterized by constraint-driven search of adequate solutions: rather than programming, the user needs to define abstract requirements, and select the best fitting proposed solutions, perhaps after some steps where he refined his requirements to better tailor the search. Combined with an easy pattern-based interface for entering constraints this development style has proven to be very effective for scientist without programming knowledge and in scenarios where the available libraries processing components are continuously evolving.

3.16 What are "good" strategies in infinite games?

Wolfgang Thomas (RWTH Aachen, DE)

Research on infinite (two-person) games shifted over the past decades from the study of existence problems (in set theory: determinacy) via algorithmic questions (in computer science: construction of automata realizing winning strategies) to a more refined perspective: How to construct strategies that are "good" in terms of (1) efficiency, or (2) appropriate format. We first review results and open problems on item (1), regarding minimization of memory size of controllers, and regarding optimization of behavior (explained in the example of reducing waiting times in "request-response games"). Then we discuss item (2): representations of strategies that are alternatives to transition systems (automata with output), namely strategy machines (which are based on Turing machines, Gelderie 2012-2014), Boolean programs (Madhusudan 2011, Brütsch 2014), and logic formulas (Rabinovich, Ths. 2007). This research can be understood as approaches to the problem stated already at the end of the classic paper of Büchi-Landweber (1969): to obtain a comprehensive view of the space of all winning strategies of a given game.

3.17 Automated Synthesis of Service Choreographies

Massimo Tivoli (University of L'Aquila, IT)

Joint work of Tivoli, Massimo; Autili, Marco; Inverardi, Paola
Main reference M. Autili, D. Di Ruscio, A. Di Salle, P. Inverardi, M. Tivoli, "A Model-Based Synthesis Process for Choreography Realizability Enforcement," in Proc. of the 16th Int'l Conf. on Fundamental Approaches to Software Engineering (FASE'13), LNCS, Vol. 7793, pp. 37–52, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-37057-1_4

Modern service-oriented systems are often built by reusing, and composing together, existing services distributed over the Internet. Service choreography is a possible form of service composition aiming at specifying the interactions between the participant services from a global perspective. In this talk, I overview a method for the distributed enforcement of service choreographies via automated synthesis of coordination delegates. When interposed among the participant services, coordination delegates intercept the service interaction and mediate it in order to realize the specified choreography. This method is implemented as

part of a model-based tool chain released to support the development of choreography-based systems within the EU CHOReOS project.

3.18 Component-Based System Design with Interfaces

Stavros Tripakis (University of California – Berkeley, US)

Joint work of Tripakis, Stavros; Lublinerman, Roberto
Main reference S. Tripakis, R. Lublinerman, "Modular Code Generation from Synchronous Models: Abstraction and Compositionality," pre-print.
URL http://www.dagstuhl.de/mat/Files/14/14232/14232.TripakisStavros.Paper.pdf

Model-based design (MBD) is a design methodology that relies on three key elements: modeling (how to capture the system that we want), analysis (how to be sure that this is the system that we want before actually building it), and synthesis (how to build a "low-level" implementation of the system from a "high-level" model/specification). This talk discusses some of our work on MBD with a focus on compositionality. Compositional methods, which allow to assemble smaller components into larger systems both efficiently and correctly, are not simply a desirable feature in system design: they are a must for building large and complex systems. A key ingredient for compositionality is that of an "interface". An interface abstracts a component, exposing relevant information while hiding internal details. We give an overview of the many uses of interfaces in MBD, from modular code generation from hierarchical models, to incremental design with interface theories, to Ptolemy simulation and FMI co-simulation, to multiview modeling.

References

1 R. Lublinerman, and S. Tripakis. *Modularity vs. Reusability: Code Generation from Synchronous Block Diagrams*. Design, Automation, and Test in Europe (DATE'08).
2 R. Lublinerman, and S. Tripakis. *Modular Code Generation from Triggered and Timed Block Diagrams*. 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'08).
3 R. Lublinerman, C. Szegedy, and S. Tripakis. *Modular Code Generation from Synchronous Block Diagrams – Modularity vs. Code Size*. 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'09).
4 S. Tripakis, B. Lickly, T. A. Henzinger, and E. A. Lee. *A Theory of Synchronous Relational Interfaces*. ACM Transactions on Programming Languages and Systems (TOPLAS), 33, 4, 2011.
5 S. Tripakis, C. Stergiou, C. Shaver, and E. A. Lee. *A modular formal semantics for Ptolemy*. Mathematical Structures in Computer Science, 23, 2013.
6 S. Tripakis, C. Stergiou, M. Broy, and E. A. Lee. *Error-Completion in Interface Theories*. International SPIN Symposium on Model Checking of Software – SPIN 2013.
7 D. Broman, C. Brooks, L. Greenberg, E. A. Lee, S. Tripakis, M. Wetter, and M. Masin. *Determinate Composition of FMUs for Co-Simulation*. Proceedings of the 13th ACM & IEEE International Conference on Embedded Software (EMSOFT'13).
8 J. Reineke, and S. Tripakis. *Basic Problems in Multi-View Modeling*. Tools and Algorithms for the Construction and Analysis of Systems – TACAS 2014.

3.19 Inhabitation problems

PawełUrzyczyn (University of Warsaw, PL)

This talk is about a specific version of synthesis: the synthesis of a proof for a given formula. Under the Curry-Howard Isomprphism, formulas correspond to types and their proofs correspond to terms (programs) of appropriate types. Therefore the provability problem is usually equivalent to an inhabitation problem (is a given type non-empty?) in a corresponding lambda-calculus.

The talk covers the following issues:

- introduction to the Curry-Howard Isomprphism;
- Ben-Yelles inhabitation algorithm for simple types and its complexity;
- extensions to arbitrary propositional connectives; and first order quantifiers;
- the automata-theoretic and the game-theoretic paradigm of proof search.

3.20 Compositional Temporal Synthesis

Moshe Y. Vardi (Rice University, US)

Synthesis is the automated construction of a system from its specification. In standard temporal-synthesis algorithms, it is assumed the system is constructed from scratch. This, of course, rarely happens in real life. In real life, almost every non-trivial system, either in hardware or in software, relies heavily on using libraries of reusable components. Furthermore, other contexts, such as web-service orchestration and choreography, can also be modeled as synthesis of a system from a library of components.

In this talk we describe and study the problem of compositional temporal synthesis, in which we synthesize systems from libraries of reusable components. We define two notions of composition: data-flow composition, which we show is undecidable, and control-flow composition, which we showis decidable. We then explore a variation of control-flow compositional synthesis, in which we construct reliable systems from libraries of unreliable components.

Joint work with Yoad Lustig and Sumit Nain.

3.21 Compositional Controller Synthesis for Stochastic Games

Clemens Wiltsche (University of Oxford, GB)

Joint work of Basset, Nicolas; Kwiatkowska, Marta; Wiltsche, Clemens

Design of autonomous systems is facilitated by automatic synthesis of correct-by- construction controllers from formal models and specifications. We focus on stochastic games, which can model the interaction with an adverse environment, as well as probabilistic behaviour arising from uncertainties. We propose a synchronising parallel composition for stochastic games that

enables a compositional approach to controller synthesis. We leverage rules for compositional assume-guarantee verification of probabilistic automata to synthesise controllers for games with multi-objective quantitative winning conditions. By composing winning strategies synthesised for the individual components, we can thus obtain a winning strategy for the composed game, achieving better scalability and efficiency at a cost of restricting the class of controllers.

3.22 Programming with Millions of Examples

Eran Yahav (Technion – Haifa, IL)

We present a framework for data-driven synthesis, aiming to leverage the collective programming knowledge captured in millions of open-source projects. Our framework analyzes code snippets and extracts partial temporal specifications. Technically, partial temporal specifications are represented as symbolic automata where transitions may be labeled by variables, and a variable can be substituted by a letter, a word, or a regular language. Using symbolic automata, we consolidate separate examples to create a database of snippets that can be used for semantic code-search and component synthesis. We have implemented our approach and applied it to analyze and consolidate millions of code snippets.

3.23 A combinatorial view of module composition for OO programming languages

Ugo de Liguoro (University of Turin, IT)

Joint work of de'Liguoro, Ugo; Tzu-Chun Chen
Main reference U. de'Liguoro, T. Chen, "Semantic Types for Classes and Mixins," pre-print.
URL http://www.di.unito.it/~deligu/papers/UdLTC14.pdf

Taking a lambda calculus with records as the basic model, we discuss the choice of a set of combinators representing various possibilities in composing software modules, designed as components to build class hierarchies (e.g. traits or mixins). We also consider a suitable extension of intersection type discipline to be thought of as a specification language for module properties, aiming at extending Rehof's method of program synthesis by inhabitation to the case of class based programming languages.

References

1 Ugo de'Liguoro. *Characterizing convergent terms in object calculi via intersection types* Proc. of TLCA'01, LNCS 2044, pp. 315–328, 2001.
2 Ugo de'Liguoro, Tzu-Chun Chen. *Semantic Types for Classes and Mixins* Proc. of ITRS'14, EPTCS, to appear.

4 Joint Discussions

Dror Fried

4.1 Component Orientation and Complexity (Tuesday 6/3/2014)

The major problem that is addressed in this discussion is in fact the core of this seminar: there are different teams, with different models of computation. There is a need to find a common ground of formalism in order to enable models comparison.

A short talk initiated by Prof. Farhad Arbab has presented an idea of the differences between direct and indirect methods of constructions, as well as the role of the components as facilitating the transition between these two models. Then, the components are combined by a protocol that expresses relationship such as: synchrony/asynchrony, exclusion, grouping, etc.

Other ideas that were brought up during this discussion:

- The users eventually care about the behaviour, not about the construction of the system. However, sometimes it is hard to tell what the user is really interested in.
- The cost, including the hidden cost of the components, plays a major role in the actual implementation.
- Eventually every software changes. We want to minimize these changes in a rapidly changing environment. How do we response to changes? Do we need to this response to be in the overall specification or in every component locally?
- How do components in uence the computational complexity? In theory component are harder to analyse because they hide information and one has to take all possibles into account. However, practice might show otherwise. Perhaps we should consider only the interaction protocol in terms of complexity, and not be concerned with the insides of the components. For example, as we usually use an existing code as a component (from a library) rather than writing one from scratch, we shouldn't be concerned with the analysis of that specific piece of code.

4.2 Challenges (Wednesday 6/4/2014)

The challenges that we should seek, and which of these challenges we focus on, is the topic of this discussion. Some of these challenges are relevant to many fields in computer science, in which scientists are not appreciated as expanding humanity knowledge, but rather as deliverers of tools that work.

- Perhaps it is better to seek a "political challenge". Something big that will draw a lot of attention. For example, the Automatic Theorem Prover. People will get fascinated, and we will get a lot of appreciation and attention. Most grand-challenges are engineering, not scientific. We should look for an engineering grand-challenge.
- However, such projects can easily go down the drain. Projects that aim too high are often not trusted, thus not approved. Perhaps we should instead be realistic. Realize that we don't have high esteem as mathematicians have, and we are only gray-collar workers who provide actual tools, instead of expanding the knowledge of humanity.
- Another approach is to show people what we have done so far, instead of showing what we plan to do. Society has kept us so far because we deliver goods, and so far we have met society's requests.
- We should do things gradually. Like SMT. It took 50 years for people to notice, and now it is big. At first, SMT was just obsession of a few people, but it advanced well,

and now it works. Perhaps this is an example that there is a strategic research as well. Specifically, we should start with small programs, take one step at a time. On the other hand, the criteria of success is dynamic, so it is hard to strategically aim in advance for a specific goal.

4.3 Benchmarks (Thursday 6/5/2014)

We have discussed the possibility of creating benchmarks that will serve as a common ground for to compare different methods. The main problem with this approach is that currently each has his own formalism in his own world. There is not one (or two) specific formats on which we can agree. This seminar is a first step, but we still need to find a way to communicate in the same language. However, one attempt that we can already try is to integrate works in Combinatory Logic with the SyGus format, and to participate in the SyGus competition. However, perhaps it will be beneficial to start with a real world problem, something concrete. Then just try to solve that problem by using various tools. We can get inspiration from the Theorem Proving community that does the same thing.

4.4 Conclusion (Thursday 6/5/2014)

In the last discussion we have expressed various ideas that were brought during the seminar:

- In comparison to 5–7 years ago, the Computer Science area uses more components. Therefore we are now more aware that these component problems exist. On the semantic level, it seems that there is a convergence of what these components do. Everybody has a clear notion what they mean by component: philosophically we agree. The differences are "only" technical.
- Still, technicality matters as component are abstract and one need something practical to work with: benchmarks, scenarios. For example – formalize what client-server architecture means. We should then ask questions such as: how does the logical for design aspects the complexity of the synthesis problem? How does experimental knowledge for software engineers affect the synthesis problem?
- From a previous experience: a low-level formalism of a declarative program may lead to a chaos (for example: by defining state as boolean logic). The reason is that the higher the formalism is, the more global mistakes there are. These global mistakes are sometimes easier to fix than local ones. Sometimes it's easier to nail the big bugs than the small.
- We need to put thought on how to test our tools. We need to test the specification as well. However performing many tests should not be our main objective. What about the option of synthesizing tests with the problem? Is this something that can be considered?
- Let's be realistic: Formal unified specification is hard. Languages are not user friendly. However, we should start think what people like to write their specification with (counter example: LTL). The industry developed their own languages- there are many, and each for a local use.
- Sometimes we mix architecture with structure. Architecture is far more than that. The attempt to formalize architecture will result in formalizing the structure, not more. It is also related to architecture patterns. You cannot formalize that. Then perhaps architecture pattern can be a part of the synthesis. Think of the architecture as an extra constraint for the synthesis. However, this is not a composition, but rather a construct of the architecture process. For example: we don't necessarily have a client-server application but we need some of it as constraints.

Participants

- Farhad Arbab
CWI – Amsterdam, NL
- Christel Baier
TU Dresden, DE
- Ugo de'Liguoro
University of Turin, IT
- Mariangiola Dezani
University of Turin, IT
- Laurent Doyen
ENS – Cachan, FR
- Boris Düdder
TU Dortmund, DE
- Bernd Finkbeiner
Universität des Saarlandes, DE
- Dror Fried
Rice University, US
- George T. Heineman
Worcester Polytechnic Inst., US
- Fritz Henglein
University of Copenhagen, DK
- Paola Inverardi
University of L'Aquila, IT
- Neil D. Jones
University of Copenhagen, DK
- Bengt Jonsson
Uppsala University, SE
- Axel Legay
INRIA Bretagne Atlantique – Rennes, FR
- Moritz Martens
TU Dortmund, DE
- Hongyang Qu
University of Sheffield, GB
- Jean-François Raskin
Université Libre de Bruxelles, BE
- Jakob Rehof
TU Dortmund, DE
- Sven Schewe
University of Liverpool, GB
- Joseph Sifakis
VERIMAG – Gières, FR
- Bernhard Steffen
TU Dortmund, DE
- Wolfgang Thomas
RWTH Aachen, DE
- Massimo Tivoli
University of L'Aquila, IT
- Stavros Tripakis
University of California – Berkeley, US
- PawełUrzyczyn
University of Warsaw, PL
- Moshe Y. Vardi
Rice University, US
- Clemens Wiltsche
University of Oxford, GB
- Eran Yahav
Technion – Haifa, IL

Report from Dagstuhl Seminar 14241

Challenges in Analysing Executables: Scalability, Self-Modifying Code and Synergy

Edited by

Roberto Giacobazzi[1], Axel Simon[2], and Sarah Zennou[3]

1 **Università degli Studi di Verona, IT,** roberto.giacobazzi@univr.it
2 **TU München, DE,** Axel.Simon@in.tum.de
3 **Airbus Group Innovations-Suresnes, FR,** sarah.zennou@eads.net

Abstract

This report summarizes the program and the outcomes of the Dagstuhl Seminar 14241, entitled "Challenges in Analysing Executables: Scalability, Self-Modifying Code and Synergy". The seminar brought together practitioners and researchers from industry and academia to discuss the state-of-the art in the analysis of binaries, the handling of the most challenging malware and the ever-lasting problem of scalability. The meeting created new links within this very diverse community and highlighted the broad interest in dealing with obfuscated code.

Seminar June 9–13, 2014 – http://www.dagstuhl.de/14241
1998 ACM Subject Classification B.2.2 Worst-case analysis, D.2.4 Formal methods, D.3.2 Macro and assembly languages, D.3.4 Debuggers and Interpreters, D.4.5 Fault-tolerance and Verification, D.4.6 Information flow controls and Invasive software, D.4.8 Modelling and prediction, D.4.9 Linkers and Loaders, F.3.2 Operational semantics and Program analysis, I.2.2 Program modification
Keywords and phrases Executable analysis, reverse engineering, malware detection, control flow reconstruction, emulators, binary instrumentation
Digital Object Identifier 10.4230/DagRep.4.6.48
Edited in cooperation with Ed Robbins

1 Executive Summary

Axel Simon

As a follow-up on the previous Dagstuhl Seminar 12051 on the analysis of binaries, the interest in attending this new seminar was very high. In the end, less than half the people that we considered inviting could attend, namely 44 people. In contrast to the previous seminar that ran for 5 days, this seminar was a four-day seminar due to a bank holiday Monday. Having arranged the talks by topic, these four days split into two days on the analysis of binaries and into (nearly) two days on obfuscation techniques.

The challenges in the realm of general binary analysis have not changed considerably since the last gathering. However, new analysis ideas and new technologies (e. g. SMT solving) continuously advance the state-of-the-art and the presentations where a reflection thereon. With an even greater participation of people from industry, the participants could enjoy a broader view of the problems and opportunities that occur in practice. Given the tight focus on binary code (rather than e. g. Java byte code), a more detailed and informed discussion

ensued. Indeed, the different groups seem to focus less on promoting their own tools rather than seeking collaboration and an exchange of experiences and approaches. In this light, the seminar met its ambition on synergy. It became clear that creating synergy by combining various tools is nothing that can be achieved in the context of a Dagstuhl Seminar. However, the collaborative mood and the interaction between various groups give hope that this will be a follow-on effect.

The second strand that crystallized during the seminar was the practical and theoretic interest in code obfuscation. Here, malware creators and analysts play an ongoing cat-and-mouse game. A theoretic understanding of the impossibility of winning the game in favor of the analysts helps the search for analyses that are effective on present-day obfuscations. In practice, a full understanding of some obfuscated code may be unobtainable, but a classification is still possible and useful. The variety of possible obfuscations creates many orthogonal directions of research. Indeed, it was suggested to hold a Dagstuhl Seminar on the sole topic of obfuscation.

One tangible outcome of the previous Dagstuhl Seminar is our GDSL toolkit that was presented by Julian Kranz. We believe that other collaborations will ensue from this Dagstuhl Seminar, as the feedback was again very positive and many and long discussions where held in the beautiful surroundings of the Dagstuhl grounds. The following abstracts therefore do not reflect on the community feeling that this seminar created. Please note that not all people who presented have submitted their abstracts due to the sensitive nature of the content and/or the organization that the participants work for.

2 Table of Contents

3 Overview of Talks

3.1 Binary-level analysis for safety-critical systems

Sebastien Bardin (CEA LIST, FR)

Joint work of Bardin, Sebastien; Védrine, Franck; Herrmann, Philippe

We present in this talk the work done at CEA LIST on binary-level analysis of safety-critical programs. Our efforts have been focused on Intermediate Languages for modelling the semantics of machine code instructions, generation of test data through dynamic symbolic execution and safe recovery of the Control-Flow Graph. We present our solutions, their limitations and a few case-studies on safety-critical programs from aeronautics and energy. We conclude by presenting some directions we want to explore and some ongoing work.

References

1 Sebastien Bardin, Philippe Baufreton, Nicolas Cornuet, Philippe Herrmann, Sebastien Labbé: Binary-Level Testing of Embedded Programs. QSIC 2013:11–20. IEEE
2 Sebastien Bardin, Philippe Herrmann: OSMOSE: automatic structural testing of executables. Softw. Test., Verif. Reliab. 21(1):29–54 (2011)
3 Sebastien Bardin, Philippe Herrmann, Jérôme Leroux, Olivier Ly, Renaud Tabary, Aymeric Vincent: The BINCOA Framework for Binary Code Analysis. CAV 2011: 165-170. Springer
4 Sebastien Bardin, Philippe Herrmann, Franck Védrine: Refinement-Based CFG Reconstruction from Unstructured Programs. VMCAI 2011:54–69. Springer

3.2 High-level semantics for low-level code

Frederic Besson (IRISA – Rennes, FR)

The "flat" memory model – where memory is as an array of bytes – is the archetype of a memory model that is not suitable for static analysis. The reason is that a local loss of precision has a dramatic (in the sense of drama) effect on the rest of the analysis. Memory models based on regions – address is an offset with respect to an abstract pointer – have proved valuable for analysing higher-level languages. For binary programs, this model is too abstract and fails to give a concrete semantics to certain programs. Borrowing ideas from symbolic execution, we show how to enhance this idealised model to capture low-level idioms. The semantics is executable and leverages SMT solvers. We have an implementation of this enhanced memory model for the CompCert C compiler. See http://www.irisa.fr/celtique/ext/csem/.

3.3 Verified Abstract Interpretation Techniques for Disassembling Low-level Self-modifying Code

Sandrine Blazy (IRISA – Rennes, FR)

Joint work of Blazy, Sandrine; Laporte, Vincent; Pichardie, David
Main reference S. Blazy, V. Laporte, D. Pichardie, "Verified Abstract Interpretation Techniques for Disassembling Low-level Self-Modifying Code," in Proc. of the 5th Int'l Conf. on Interactive Theorem Proving (ITP'14), LNCS, Vol. 8558, pp. 128–143, Springer, 2014.
URL http://dx.doi.org/10.1007/978-3-319-08970-6_9

Static analysis of binary code is challenging for several reasons. In particular, standard static analysis techniques operate over control flow graphs, which are not available when dealing with self-modifying programs which can modify their own code at runtime. We formalize in the Coq proof assistant some key abstract interpretation techniques that automatically extract memory safety properties from binary code. Our analyzer is formally proved correct and has been run on several self-modifying challenges, provided by Cai et al. in their PLDI 2007 paper.

3.4 CopperDroid: On the Reconstruction of Android Malware Behaviors

Lorenzo Cavallaro (RHUL – London, GB)

Joint work of Fattori, Aristide; Tam, Kimberly; Khan, Salahuddin J.; Reina, Alessandro; Cavallaro, Lorenzo

Today mobile devices and their application marketplaces drive the entire economy of the mobile landscape. For instance, Android platforms alone have produced staggering revenues exceeding 5 billion USD, which unfortunately attracts cybercriminals with malware now hitting the Android markets at an alarmingly rising pace.

To better understand this slew of threats, in this talk I present CopperDroid, an automatic VMI-based dynamic analysis system to reconstruct the behavior of Android malware. Based on the key observation that all interesting behaviors are eventually expressed through system calls, CopperDroid presents a novel unified analysis able to capture both low-level OS-specific and high-level Android-specific behaviors. To this end, CopperDroid presents an automatic system call-centric analysis that faithfully reconstructs events of interests, including IPC and RPC interactions and complex Android objects, to describe the behavior of Android malware regardless of whether it is initiated from Java or native code execution. CopperDroid's analysis generates detailed behavioral profiles that abstract a large stream of low-level – sometimes uninteresting – events into concise high-level semantics, which are well-suited to provide effective insights.

Extensive evaluation on more than 2,900 Android malware samples, show that CopperDroid faithfully describes OS- and Android-specific behaviors and, through the use of a simple yet effective app stimulation technique, successfully triggers and discloses additional behaviors on more than 60 precent (on average) of the analyzed malware samples, qualitatively improving code coverage of dynamic-based analyses.

References

1 Aristide Fattori and Kimberly Tam and Salahuddin J. Khan and Alessandro Reina and Lorenzo Cavallaro. *CopperDroid: On the Reconstruction of Android Malware Behaviors*. Technical Report MA-2014-01 Royal Holloway University of London, Februrary, 2014

2 Alessandro Reina and Aristide Fattori and Lorenzo Cavallaro. *A System Call-Centric Analysis and Stimulation Technique to Automatically Reconstruct Android Malware Behaviors*. Proceedings of the 6th European Workshop on System Security (EUROSEC), April, 2013

3.5 Practical Problems in Automated Static Analysis of Malware

Cory Cohen (Software Engineering Institute – Pittsburgh, US)

Operational malware analysts have different priorities and motivations than most academic researchers in the static analysis of binaries. This presentation will highlight some of those differences, and provide suggestions on how to promote adoption of research prototypes for operational use. It also presents some background on the current state of malware, examples of problem areas interesting to malware analysts, and discusses stack delta analysis algorithms as an example of these ideas.

3.6 Evaluating the strength of software protections

Bjorn De Sutter (Ghent University, BE)

Main reference ASPIRE EU FP7 Project – Advanced Software Protection: Integration, Research and Exploitation.
URL http://www.aspire-fp7.eu

An overview of the ASPIRE project is presented, focusing on the major goals of developing a protection tool flow, a reference architecture for protected applications, and decision support to select and apply protections. Then the challenge of modelling and evaluating the protection strength against attacks is discussed, and an overview is given of different types of evaluation metrics, ranging from software engineering metrics, compiler-based code analysis metrics, formal modelling metrics, tool-based metrics, and human experiments and human comprehension modelling. Their strong points and weak points are discussed, after which the talk concludes with a list of open challenges in the domain of protection strength evaluation.

3.7 Understanding Programs that Don't Want to be Understood

Saumya K. Debray (University of Arizona – Tucson, US)

Joint work of Yadegari, Babak; Johannesmeyer, Brian; Whitely, Benjamin

Malicious software are usually obfuscated to avoid detection and resist analysis. When new malware is encountered, such obfuscations have to be penetrated or removed ("deobfuscated")

in order to understand the internal logic of the code and devise countermeasures. This talk discusses a generic approach for deobfuscation of obfuscated executable code. Our approach does not make any assumptions about the nature of the obfuscations used, but instead uses semantics-preserving program transformations to simplify away obfuscation code. We have applied a prototype implementation of our ideas to a variety of different kinds of obfuscation, including emulation-based obfuscation, emulation-based obfuscation with runtime code unpacking, and return-oriented programming. Our experimental results are encouraging and suggest that this approach can be effective in extracting the internal logic from code obfuscated using a variety of obfuscation techniques, including tools such as Themida that previous approaches could not handle.

3.8 Static analysis of avionics software at Airbus

David Delmas (Airbus S.A.S. – Toulouse, FR)

Analysis of executables rely on static analyzers based on Abstract Interpretation: aiT WCET (for time-critical applications) and Stackanalyzer (for most avionics software). Run-time error analysis is performed at source code level using Astrée. These analyzers scale up to very large avionics software, while remaining sufficiently precise for industrial use. More local analyzes rely on tools such as Caveat, Frama-C and Fluctuat. Formal verification at source code level is still valid at binary level, provided a certified compiler is used, e. g. CompCert.

Considering raising security concerns and upcoming related standards for avionics, more binary analyses will be needed. Integration of third party binaries onto a software platform requires the verification of conformance to platform interface requirements. Detection of memory vulnerabilities and CFG reconstruction are of interest to audit third party software. Also, CFI/SFI sandboxing approaches, and related automatic verification, are of interest for some software platforms. Among new challenges is the security evaluation of some non-critical equipments, using standard connected PCs/tablets, and external Java bytecode.

For the future, timing analysis of multithreaded applications running on top of embedded OS and complex processors remains also an objective. Side-channel analyzes, e. g. information leakage through caches, is of interest. At source code level, formal verification of correct implementation of a given security policy on large complex systems is a long term challenge.

The main industrial requirements for such analyses are automation, scalability and precision. In safety-related domains, soundness is paramount.

3.9 Insight: A(nother) Binary Analysis Framework

Emmanuel Fleury (University of Bordeaux, FR)

We aim to have a full and efficient platform to easily try out novel algorithms or techniques. For this, we provide a full C++ framework designed for Unix systems (*BSD, Linux, MacOS X, ...) which contains a wide-spectrum binary format loaders (ELF, PE, Mach-O, ...), a decoder translating from assembly code (i386, amd64, ...) into our intermediate language,

an interpreter to execute the program over a (potentially abstract) domain and several facilities to simplify, manipulate or transform the graph and the expressions extracted from the original program.

This talk introduces the Insight framework and includes a small demonstration of our interactive symbolic debugger.

3.10 Decompilation into Logic using HOL4

Anthony Fox (University of Cambridge, GB)

Joint work of Fox, Anthony; Myreen, Magnus

We present formal ISA models and tools that support the decompilation of machine-code into HOL4 functions. This decompilation is validated through the use of certificate theorems. We provide an overview of our ISA models and list some notable case studies. The HOL4 tools are demonstrated.

3.11 Similarity analysis in Big Code of executables

Roberto Giacobazzi (University of Verona, IT)

Data-sets in huge software enclaves, such as code, specifications, analyses, etc. put forward new and unconventional challenges to traditional Big-Data research. If Big-Data requires adequate infrastructures and abstractions for mining and learning information from huge data-sets, in Big-Code we need to include interpretation in order to be able to extract and represent the extensional meaning of programs. Any Big-Code analytics is therefore necessarily based on a form of interpretation and analysis, able to mine semantics and returning approximate information about programs behavior. We introduce a mixed syntactic/semantics approximation model based on symbolic automata for similarity analysis of large enclaves of binary executables. Following the structure of their control flow graph, disassembled binary executables are represented as symbolic automata, where nodes are program points and predicates represent the semantics of each basic block. Approximation is made by abstract interpretation, acting on these symbolic automata both at syntactic and semantic level. At syntactic level, the code of basic blocks is approximated by extracting their BinJuice. At semantic level the data information is abstracted in standard numerical domains. Simplification operations of the resulting abstract symbolic automata are discussed in order to extract common signatures of similar executables.

3.12 Large Scale Concurrent ISA Semantics (via the Sail DSL)

Kathryn E. Gray (University of Cambridge, GB)

There has been much work on formal models of ISA behaviour and on domain- specific languages for expressing them, but it has not addressed the relaxed- memory concurrency of multiprocessors such as IBM Power and ARM. On the other side, recent work by Sarkar et al. has established semantics for the latter but is not integrated with a large-scale ISA model. We discuss what is necessary to combine a large-scale ISA model with a realistic concurrency semantics and our work (in progress) to that end: using our Sail language to express a Power ISA model that we semi-automatically extract from the IBM documentation.

3.13 Dynamic Analysis: Knowing When to Stop

Paul Irofti (Bucharest, RO)

I made a lot of progress on the emulator since my last talk two years ago at the 12051 Seminar "Analysis of Executables: Benefits and Challenges". It is now a mature production-ready project (see the 'Other' document for a whitepaper on it) and I want to talk about the problems I faced, focusing on one in particular which is the stopping problem.

The classic scenario is that an executable gets loaded and emulated until the executable exit by itself. But there are times when the executable takes longer to be emulated than you'd want it to or, worse yet, the emulation process gets hogged somewhere due to anti-debugging techniques or bugs in the actual program. That's why most dynamic analysis solutions in the malware industry employ some sort of watchdog-like mechanism that forces a stop in emulation after a certain threshold is reached. These solutions involve time-based or emulated instruction-based thresholds that are either non-deterministic or unfair to certain machines (be it really fast machines or older, slower ones).

And so, I want to talk about a solution that provides a deterministic and fair on all systems mechanism of stopping the emulation process.

3.14 Generic Decoder Specification Language

Julian Kranz (TU München, DE)

Joint work of Kranz, Julian; Sepp, Alexander; Simon, Axel
Main reference A. Sepp, J. Kranz, A. Simon, "GDSL: A Generic Decoder Specification Language for Interpreting Machine Language," in Proc. of the 3rd Workshop on Tools for Automatic Program Analysis (TAPAS'12), ENTCS, Vol. 289, pp. 53–64, Elsevier, 2012; pre-print available from author's webpage.
URL http://dx.doi.org/10.1016/j.entcs.2012.11.006
URL http://www2.in.tum.de/bib/files/sepp12gdsl.pdf

Analysing binary code begins with the interpretation of a low-level binary input stream. This process consists of two steps: turning the byte stream into a sequence of instructions,

i. e. giving syntax to it, and giving meaning to the instructions, i. e. translating them into some kind of semantics representation. Addressing the first step, we present our functional language called GDSL which is geared to the simple and effective specification of binary decoders. To this end, the language is equipped with special syntax that allows the easy access to slices of bytes. As a proof of its practicability, GDSL ships with a complete decoder for Intel x86 implemented in GDSL. We demonstrate the effectiveness of the GDSL compiler by comparing our decoding performance to Intel's XED decoder. Regarding the translation to semantics, we present the minimalistic language RReil which is designed to be suited for binary analysis. Using a simple optimization, we achieve a translation output size of roughly only three semantic primitives per x86 instruction.

3.15 Formal Specification and Validation of ARM Machine-Code Analyses

Alexey Loginov (GrammaTech Inc.- Ithaca, US)

We will describe the foundations of extending GrammaTech's machine-code analysis to the ARM instruction-set architecture (ISA). We will start with our approach to the creation of a trustworthy specification of ARM instruction semantics. We will then describe our efforts on validating every step in the creation of intermediate representations of ARM binaries. The goal is to construct intermediate representations that enable sound, yet precise, static analysis. Our validation covers instruction decoding, disassembly, concrete instruction semantics, as well as abstract analyses. Finally, we will describe our approach to ISA-independent regression testing of static analyses, such as Value-Set Analysis (VSA). This approach enabled rapid adaptation of existing x86-only regression tests to testing static analyses of ARM binaries.

3.16 CoDisasm :a disassembly of self-modifying binaries with overlapping instructions

Jean-Yves Marion (LORIA – Nancy, FR)

Disassembly is a key task in software debugging and malware analysis. It involves the recovery of assembly instructions from binary machine code. It can be problematic in the case of malicious code, as malware writers often employ techniques to thwart correct disassembly by standard tools. Nonetheless, disassembly is a crucial step in malware reverse engineering. Correct disassembly of binaries is necessary to produce a higher level representation of the code and thus allow the analysis to develop high-level understanding of its behavior and purpose.

In this paper, we focus on the disassembly of self-modifying binaries with overlapping instructions. Current state-of-the-art disassemblers fail to interpret these two common forms of obfuscation, causing an incorrect disassembly of large parts of the input.

We have developed a standalone disassembler called CoDisasm that implements this approach, together with a plug-in for the popular reversing engineering tool IDA called

BinViz to visualize the code waves generated by CoDisasm and to visualize overlapping instructions. Our approach substantially improves the success of disassembly when confronted with both self-modification and code overlap in analyzed binaries. Experimental results on about five hundred malware samples show that our approach correctly recovers large parts of the code. To our knowledge, no other disassembler thwarts both of these obfuscations methods together.

3.17 Worst Case Execution Time for Safty Critical Systems

Florian Martin (AbsInt – Saarbrücken, DE)

Joint work of Martin, Florian; Daniel Kästner, Markus Pister, Gernot Gebhard, Christian Ferdinand

All contemporary safety standards require to demonstrate the absence of functional and non-functional safety hazards. In real-time systems this includes demonstrating the absence of critical timing hazards.

To meet this verification objective it is necessary to show the correctness of the timing behavior with ad- equate confidence. Adequate confidence means that the evidence provided can be trusted beyond reason- able doubt. There are two main sources of doubt: the logical doubt associated with the validity of the reasoning and the epistemic doubt associated with uncertainty about the underlying assumptions. A fundamental timing property is the per-task worst-case execution (WCET). It is an ingredient for determining all higher-level timing concepts like worst-case response times, and system-wide end-to-end times. This talk gives an overview of the challenges in ensuring timeliness of real-time software focusing on the worst-case execution time problem. It describes the principles of abstract interpretation-based WCET analysis and summarizes the confidence argument for applying it in the certification process of safety-critical software, addressing both logical and epistemic doubt.

3.18 Obfuscation as Incomplete Approximation

Isabella Mastroeni (University of Verona, IT)

Main reference R. Giacobazzi, N. Jones, I. Mastroeni, "Obfuscation by Partial Evaluation of Distorted Interpretation," in Proc. of the ACM SIGPLAN 2012 Workshop on Partial Evaluation and Program Manipulation (PEPM'12), pp. 63–72, ACM, 2012.
URL http://dx.doi.org/10.1145/2103746.2103761

We present a novel approach to automatically generating obfuscated code P' from any program P whose source code is given. Start with a (program-executing) interpreter *interp* for the language in which P is written. Then "distort" *interp* so it is still correct, but its specialization P' w.r.t. P is transformed code that is equivalent to the original program, but harder to understand or analyze. Potency of the obfuscator is proved with respect to a general model of the attacker, modeled as an approximate (abstract) interpreter. A systematic approach to distortion is to make program P obscure by transforming it to P' on which (abstract) interpretation is incomplete. Interpreter distortion can be done by making residual in the specialization process sufficiently many interpreter operations to defeat an

attacker in extracting sensible information from transformed code. Our method is applied to: code flattening, data-type obfuscation, and opaque predicate insertion. The technique is language independent and can be exploited for designing obfuscating compilers.

3.19 Sendmail crackaddr – Static Analysis strikes back

Bogdan Mihaila (TU München, DE)

Joint work of Mihaila, Bogdan; Sepp, Alexander; Simon, Axel
Main reference A. Sepp, B. Mihaila, A. Simon, "Precise Static Analysis of Binaries by Extracting Relational Information," in Proc. of the 18th Working Conference on Reverse Engineering (WCRE'11), pp. 357–366, IEEE, 2011.
URL http://dx.doi.org/10.1109/WCRE.2011.50

The "sendmail crackaddr" bug from 2003 is an example for a vulnerability that is difficult to prove using static analysis. In the course of analyzing the simplified version of this famous example we discovered that it is surprisingly easier than expected to separate the vulnerable from the non-vulnerable example. We show that it can be solved using abstract interpretation and show what invariants are inferred by our binary analysis framework: Bindead. Though the results are promising, there is still some work necessary to apply the same methods to the original example.

References

1 B. Mihaila, A. Sepp and A. Simon. *Widening as Abstract Domain.* In G. Brat, N. Rungta and A. Venet, editors, NASA Formal Methods, volume 7871 of LNCS, pages 170–186, Moffett Field, California, USA, May 2013. Springer.
2 A. Sepp, B. Mihaila and A. Simon. *Precise Static Analysis of Binaries by Extracting Relational Information.* In M.Pinzger and D. Poshyvanyk, editors, Working Conference on Reverse Engineering, Limerick, Ireland, October 2011. IEEE Computer Society.

3.20 Through the Lens of Abstraction

Aditya Thakur (University of Wisconsin – Madison, US)

Joint work of Thakur, Aditya; Reps, Thomas

This talk explores the use of abstraction in two areas of automated reasoning: verification of programs, and decision procedures for logics.

Establishing that a program is correct is undecidable in general. Program- verification tools sidestep this tar-pit of undecidability by working on an abstraction of a program, which over-approximates the original program's behavior. The theory underlying this approach is called abstract interpretation, and is around forty years old. However, harnessing abstraction to develop a scalable and precise abstract interpreter still remains a challenging problem.

This talk also exposes the use of abstraction in the design and implementation of decision procedures. I call such an abstraction-centric view of decision procedures Satisfiability Modulo Abstraction. Abstraction provides a new language for the description of decision procedures, leading to new insights and new ways of thinking.

The common use of abstraction also brings out a non-trivial and useful relationship between program verification and decision procedures.

3.21 Turbulence – an assembly level obfuscator

Axel Tillequin (Airbus Group – Suresnes, FR)

Turbulence is an assembly level obfuscator used inside Airbus Group for IP protection. It has been developed since 2005. It seamlessly integrates into the gcc toolchain and manages to automatically determine how many obfuscating iterations are needed by reaching a fixed point on the distribution of its internal set of obfuscating transforms. Open questions are related to finding a good measure of complexity of some obfuscated code in order to provided an adaptive approach of reaching an uniform complexity by orienting the obfuscator on badly obfuscated parts.

3.22 Interactive static analysis

Franck Vedrine (CEA – Gif-sur-Yvette, FR)

Only few abstract interpreters have an interactive interface. In this talk we present the concepts behind the interactive interface of the Fluctuat static analyzer for C programs [1], and possible usages. While already useful for source-level analysis, we do think that interactivity is even more interesting for binary-level analysis, where it is not so evident for a user to define an analysis scenario or to insert annotations at a dedicated point. The implementation into CFGBuiler is future work.

References

1 Franck Vedrine, Eric Goubault, Sylvie Putot, Tristan Le Gall: *Interactive Analysis in FLUCTUAT*. Tools for Automatic Program Analysis – TAPAS 2012, Deauville, France 2012

3.23 Jackdaw: Automatic, unsupervised, scalable extraction and semantic tagging of (interesting) behaviors

Stefano Zanero (Politecnico di Milano University, IT)

Joint work of Polino, Mario; Scorti, Andrea; Maggi, Federico; Zanero, Stefano

When analyzing (malicious) software, hybrid static-dynamic program analysis techniques help the analyst in finding interesting behaviors. One of the key requirements of these methods is a catalog of patterns or specifications of such interesting behaviors, which need to be created manually.

Due to the rising number of complex malicious software and the growth of their potential, unknown yet interesting behaviors, automatic techniques are needed to build their specifications, present them to the analyst, and create a catalog of matching rules and relevant implementations (e. g., variants).

We propose Jackdaw, an automatic behavior extractor and semantic tagger. Our system first exploits jointly static control-flow analysis and dynamic data- flow analysis on malware samples to find interesting, connected sequences of API calls that are potential behaviors. Then, it maps these building blocks to their implementation(s), taking care of capturing and modeling the distinct characteristics of each variant's implementation. Finally, it associates semantic information to the behaviors, so as to create compact and descriptive summary that help the analysts in the first phases of reverse engineering. To do this, it matches relevant code against Web knowledge bases.

We tested Jackdaw on 1,272 distinct variants drawn from 17 families. We compared the behaviors extracted automatically by Jackdaw against a ground truth of 44 behaviors created manually by expert analysts: Jackdaw matched 77.3% of them. We also discover 466 novel behaviors, among which manual exploration reveals interesting findings. Manual analysis confirms also that the semantic tags that Jackdaw attaches to the behaviors are meaningful.

3.24 Binary analysis and manipulation at Airbus Group Innovations

Sarah Zennou (Airbus Group – Suresnes, FR)

Joint work of Zennou, Sarah; Biondi, Philippe; Mehrenberger, Xavier; Tillequin, Axel

This talk presents research activities at the research center of Airbus Group that are linked to the topics of the seminar: malware classification, scalable static analyses and obfuscation.

Participants

- Davide Balzarotti
EURECOM – Biot, FR
- Sébastien Bardin
CEA LIST, FR
- Frederic Besson
IRISA – Rennes, FR
- Sandrine Blazy
IRISA – Rennes, FR
- Juan Caballero
IMDEA Software Institute – Madrid, ES
- Lorenzo Cavallaro
RHUL – London, GB
- Aziem Chawdhary
University of Kent, GB
- Cory Cohen
Software Engineering Institute – Pittsburgh, US
- Mila Dalla Preda
University of Verona, IT
- Bjorn De Sutter
Ghent University, BE
- Saumya K. Debray
Univ. of Arizona – Tucson, US
- David Delmas
Airbus S.A.S. – Toulouse, FR
- Thomas Dullien
Google Switzerland, CH
- Emmanuel Fleury
University of Bordeaux, FR
- Anthony Fox
University of Cambridge, GB
- Roberto Giacobazzi
University of Verona, IT
- Kathryn E. Gray
University of Cambridge, GB
- Paul Irofti
Bucharest, RO
- Yan Ivnitskiy
Trail of Bits Inc. – New York, US
- Andy M. King
University of Kent, GB
- Tim Kornau
Google Switzerland, CH
- Julian Kranz
TU München, DE
- Colas Le Guernic
Direction Generale de l'Armement, FR
- Junghee Lim
GrammaTech Inc.- Ithaca, US
- Alexey Loginov
GrammaTech Inc.- Ithaca, US
- Federico Maggi
Politecnico di Milano Univ., IT
- Jean-Yves Marion
LORIA – Nancy, FR
- Florian Martin
AbsInt – Saarbrücken, DE
- Isabella Mastroeni
University of Verona, IT
- Bogdan Mihaila
TU München, DE
- Magnus Myreen
University of Cambridge, GB
- Gerald Point
University of Bordeaux, FR
- Edward Robbins
University of Kent, GB
- Bastian Schlich
ABB AG Forschungszentrum Deutschland – Ladenburg, DE
- Alexander Sepp
TU München, DE
- Axel Simon
TU München, DE
- Aditya Thakur
University of Wisconsin – Madison, US
- Axel Tillequin
Airbus Group – Suresnes, FR
- Franck Védrine
CEA – Gif-sur-Yvette, FR
- Aymeric Vincent
University of Bordeaux, FR
- Xueguang Wu
TU München, DE
- Brecht Wyseur
NAGRA Kudelski Group SA – Cheseaux, CH
- Stefano Zanero
Politecnico di Milano Univ., IT
- Sarah Zennou
Airbus Group – Suresnes, FR

Report from Dagstuhl Seminar 14261

Software Development Analytics

Edited by

Harald Gall[1], Tim Menzies[2], Laurie Williams[3], and Thomas Zimmermann[4]

1 **Universität Zürich, CH,** gall@ifi.uzh.ch
2 **North Carolina State University, US,** tim@menzies.us
3 **North Carolina State University, US,** williams@csc.ncsu.edu
4 **Microsoft Research – Redmond, US,** tzimmer@microsoft.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14261 "Software Development Analytics". We briefly summarize the goals and format of the seminar, the results of the break out groups, and a draft of a manifesto for software analytics. The report also includes the abstracts of the talks presented at the seminar.

Seminar June 22–27, 2014 – http://www.dagstuhl.de/14261
1998 ACM Subject Classification D.2 Software Engineering
Keywords and phrases software development, data-driven decision making, analytics, empirical software engineering, mining software repositories, business intelligence, predictive analytics
Digital Object Identifier 10.4230/DagRep.4.6.64
Edited in cooperation with Reid Holmes (University of Waterloo, CA)

1 Executive Summary

Harald Gall
Tim Menzies
Laurie Williams
Thomas Zimmermann

Software and its development generate an inordinate amount of data. For example, check-ins, work items, bug reports and test executions are recorded in software repositories such as CVS, Subversion, GIT, and Bugzilla. Telemetry data, run-time traces, and log files reflect how customers experience software, which includes application and feature usage and exposes performance and reliability. The sheer amount is truly impressive:

- As of July 2013, Mozilla Firefox had 900,000 bug reports, and platforms such as Sourceforge.net and GitHub hosted millions of projects with millions of users.
- Industrial projects have many sources of data at similar scale.

But how can this data be used to improve software? Software analytics takes this data and turns it into actionable insight to inform better decisions related to software. Analytics is commonly used in many businesses—notably in marketing, to better reach and understand customers. The application of analytics to software data is becoming more popular.

To a large extent, software analytics is about what we can learn and share about software. The data include our own projects but also the software projects by others. Looking back

at decades of research in empirical software engineering and mining software repositories, software analytics lets us share all of the following:

- **Sharing insights.** Specific lessons learned or empirical findings. An example is that in Windows Vista it was possible to build high-quality software using distributed teams if the management is structured around code functionality (Christian Bird and his colleagues).
- **Sharing models.** One of the early models was proposed by Fumio Akiyama and says that we should expect over a dozen bugs per 1,000 lines of code. In addition to defect models, plenty of other models (for example effort estimation, retention and engagement) can be built for software.
- **Sharing methods.** Empirical findings such as insights and models are often context-specific, e. g., depend on the project that was studied. However, the method ("recipe") to create findings can often be applied across projects. We refer to *"methods"* as the techniques by which we can transform data into insight and models.
- **Sharing data.** By sharing data, we can use and evolve methods to create better insight and models.

The goal of this seminar was to build a roadmap for future work in this area. Despite many achievements, there are several challenges ahead for software analytics:

- How can we make data useful to a wide audience, not just to developers but to anyone involved in software?
- What can we learn from the vast amount of unexplored data?
- How can we learn from incomplete or biased data?
- How can we better tie usage analytics to development analytics?
- When and what lessons can we take from one project and apply to another?
- How can we establish smart data science as a discipline in software engineering practice and research as well as education?

Seminar Format

In this seminar, we brought together researchers and practitioners from academia and industry who are interested in empirical software engineering and mining software repositories to share their insights, models, methods, and/or data. Before the seminar, we collected input from the participants through an online survey to collect relevant themes and papers for the seminar. Most themes from the survey fell into the categories of method (e. g., measurement, visualization, combination of qualitative with quantitative methods), data (e. g., usage/telemetry, security, code, people, etc.), and best practices and fallacies (e. g., how to choose techniques, how to deal with noise and missing data, correlation vs. causation). A theme that also emerged in the pre-Dagstuhl survey was analytics for the purpose of theory format, i. e., *"data analysis to support software engineering theory formation (or, data analytics in support of software science, as opposed to software engineering)"*.

At the seminar, we required that attendees

1. discuss the next generation of software analytics;
2. contribute to a *Software Analytics Manifesto* that describes the extent to which software data can be exploited to support decisions related to development and usage of software.

Attendees were required to outline a set of challenges for analytics on software data, which will help to focus the research effort in this field. The seminar provided ample opportunities for discussion between attendees and also provide a platform for collaboration between attendees since our time was divided equally between:

1. Plenary sessions where everyone gave short (10 minute) presentations on their work.
2. Breakout sessions where focus groups worked on shared tasks.

Our schedule was very dynamic. Each day ended with a "think-pair-share" session where some focus for the next day was debated first in pairs, then shared with the whole group. Each night, the seminar organizers would take away the cards generated in the "think-pair-share" sessions and use that feedback to reflect on how to adjust the next day's effort.

2 Table of Contents

3 The Manifesto

To compile the manifesto, we followed three steps:

1. In a think-pair-share session we collected 150 statements that participants felt should be part of the manifesto. Over the Dagstuhl week, the organizers sorted the cards into categories.
2. A breakout group compiled a draft manifesto, which was then used in a plenary session to establish core groups that should be part of a manifesto. The resulting groups were definitions, general statements, people, data, methods, results and outcomes.
3. After the seminar, the organizers selected representative statements, which were then rated by 22 attendees as part of a post-Dagstuhl survey ("In your opinion, how important is it to include this statement for a manifesto on data science in software engineering?").

In the rest of this section, we list the statements that were rated favoribly by 66.6% of the survey participants as *Essential (E)* or *Worthwhile (W)*. Statements that were rated by 40.0% of survey participants as *Essential* are printed in bold.

3.1 Statements defining data science and analytics

- **Software analytics is to utilize data-driven approaches to obtain insightful and actionable information to help software practitioners with their data related tasks** (E: 0.682, W: 0.136)
- Data science in SE should lead to: (one of) Insights about users; Advise for practitioners (which tools to use, design); Theory for researchers; Innovations for all (E: 0.364, W: 0.318)

3.2 General statements

- **Your project has a history. Learn from it. Decide from it. Embrace it.** (E: 0.429, W: 0.381)
- **What counts is insights not numbers!** (E: 0.429, W: 0.381)
- **Exploration matters.** (E: 0.4, W: 0.4)
- We strive to do the best we can with the evidence at hand, but we accept that that evidence may be incomplete, noisy, and even wrong (E: 0.364, W: 0.455)
- We will be able to gain insights from the past to improve the future. (E: 0.333, W: 0.381)
- Data, analyses, methods and results have to be publicly shared (E: 0.227, W: 0.455)
- SE data science should be actionable, reproducible. Should not be about finding a way to apply your hammer but finding solutions to real problem. (E: 0.19, W: 0.524)
- Good data science does not get in the way of developing software (distraction, additional data collection) but supports it (makes it more efficient) (E: 0.143, W: 0.571)
- Measure before action; act; measure again. (E: 0.136, W: 0.545)
- Generalizations should be viewed with a healthy skepticism (E: 0.095, W: 0.571)

3.3 Statements about people

- **Data doesn't decide, people do.** (E: 0.545, W: 0.136)
- Good data science takes the people into the account not just code and checkins (aka, in a study, ask the people involved if your results make sense) (E: 0.364, W: 0.5)

3.4 Statements about data

- **Understand where the data comes from, accepting that it may be gamed** (E: 0.5, W: 0.318)
- **Quality of data is more than quantity** (E: 0.409, W: 0.409)
- **Impact requires actionable data** (E: 0.4, W: 0.2)
- Data is merely one component of the large amount of insight, experience, and knowledge that informs decisions (E: 0.318, W: 0.409)
- Do check your data multiple times (E: 0.227, W: 0.591)

3.5 Statements about methods

- **Engage domain experts in validation of analysis** (E: 0.571, W: 0.381)
- **Interpretation and visualization is central to data science** (E: 0.5, W: 0.364)
- **We value qualitative study as much as quantitative study; often that's where the insights come from** (E: 0.476, W: 0.429)
- **Replicate and triangulate** (E: 0.455, W: 0.364)
- Big data research should not only consider machine generated data. Qualitative data are of equal importance. (E: 0.381, W: 0.381)
- Effect size matters (E: 0.35, W: 0.45)
- Actionable impact over sophisticated method. (E: 0.333, W: 0.429)
- When it comes to metrics, more is not necessarily better. (E: 0.286, W: 0.381)
- Context must accompany every method. (E: 0.238, W: 0.571)
- Data Science is integrating and analyzing data from different sources. (E: 0.19, W: 0.524)

3.6 Statements about results and outcomes

- **Communicating results is as important as computing results** (E: 0.524, W: 0.333)
- **Analytics should lead to action** (E: 0.476, W: 0.286)
- Make results actionable and relevant (E: 0.381, W: 0.429)
- Publish what didn't work. (E: 0.364, W: 0.545)
- Data science should produce actionable findings (E: 0.333, W: 0.476)
- Value usefulness to decide over precision or correctness (E: 0.318, W: 0.682)

4 Follow-up Work

At the seminar, it was recognized that the community needs a web portal to store and distribute its community product. That portal is currently being developed.

Also, attendees commented there were many "best practices" that were unknown to the broader community. This resulted in an all-too-varied performance result when newcomers struggled to apply analytics to their particular project. Hence, it was decided to co-write a book **"Perspectives on Data Science for Software Engineering"** where each (small) chapter would be written by one Dagstuhl seminar attendee as well as other well-known people in the field.

5 Overview of Talks

5.1 Emotion Mining for Software Developers

Bram Adams (Polytechnique Montreal, CA)

Joint work of Murgia, Alessandro; Tourani, Parastou; Adams, Bram; Ortu, Marco
Main reference A. Murgia, P. Tourani, B. Adams, M. Ortu, "Do Developers Feel Emotions? An Exploratory Analysis of Emotions in Software Artifacts," in Proc. of the 11th Working Conf. on Mining Software Repositories (MSR'14), pp. 262–271, ACM, 2014.
URL http://dx.doi.org/10.1145/2597073.2597086

Software development is a collaborative activity in which developers interact to create and maintain a complex software system. Human collaboration inevitably evokes emotions like joy or sadness, which can affect the collaboration either positively or negatively, yet not much is known about the individual emotions and their role for software development stakeholders. We analyzed whether development artifacts like issue reports carry any emotional information about software development. This is a first step towards verifying the feasibility of an automatic tool for emotion mining in software development artifacts: if humans cannot determine any emotion from a software artifact, neither can a tool. Analysis of the Apache Software Foundation issue tracking system shows that developers do express emotions (in particular gratitude, joy and sadness), yet more investigation is needed before building a fully automatic emotion mining tool.

5.2 Software analytics with email data

Alberto Bacchelli (TU Delft, NL)

Joint work of Bacchelli, Alberto; Lanza, Michele; Humpa, Viteszlav
Main reference A. Bacchelli, M. Lanza, V. Humpa, "RTFM (Read the Factual Mails) – Augmenting Program Comprehension with Remail," in Proc. of the 15th European Conf. on Software Maintenance and Reengineering (CSMR'11), pp. 15–24, IEEE, 2011; pre-print available from author's webpage.
URL http://dx.doi.org/10.1109/CSMR.2011.6
URL http://sback.it/publications/csmr2011.pdf

The evolution of software systems leaves its traces in a number of artifacts. Some artifacts are made of structured data (e. g., source code) that is easily parseable, while others are made of unstructured data (e. g., documentation and emails) that is more difficult to analyze. Nevertheless unstructured data contain precious knowledge to support software engineering.

In this talk I provide initial evidence that email data can be effectively used as a valuable target for software analytics. In particular, I will present anecdotal evidence on the usefulness of email data in supporting four program comprehension tasks, namely (1) finding Entry Points in a software system, (2) conducting Software Evolution Analysis, (3) improving Expert Finding techniques, and (4) recovering Additional Documentation about system's entities. The aim of the presentation is to trigger future collaboration and project in using unstructured software data to support software engineering.

5.3 Software Analytics to Build Recommender Systems

Ayse Bener (Ryerson University – Toronto, CA)

The goal of evidence based analytics for software systems is to create methods, techniques, tools and processes to improve processes and/or to efficiently allocate resources. We need to migrate from prediction to recommendation by building models that focus on reasons and casual relationships. We need to integrate prediction models into business rules and processes to improve software development processes as well as modeling people aspects. There are also other challenges that both research and practice should consider. One of these challenges is reproducibility of approaches since no one shares enough data and there are no standard process/ framework to conduct analytics. Another challenge is the lack of integration of research tools into development platforms such as Jira, GitHub, Eclipse, etc.

5.4 Data-Driven Engineering at Microsoft

Trevor Carnahan (Microsoft Research – Redmond, US)

An inside look into Microsoft efforts to apply analytics into data-driven engineering of software and a fresh look at a modern developer's responsibilities and activities. This talk starts with general decision making data concepts at work. Then the talk outlines a few data systems developed in SQL Server and in Tools for Software Engineers currently in use for decisions. It finishes with the impact of devops and more service and cloud development to challenge conceptions of a software engineer.

5.5 Misconceptions about mining bug repositories

Serge Demeyer (University of Antwerp, BE)

In this talk I present a few misconceptions that some researchers have about the records maintained in software repositories in general and bug repositories in particular. These misconceptions were harvested from a series of focus groups we organised with industrial team leads about promising research mature enough to be applied in practice.

In no particular order (and without criticising researchers working on these problems) we received the following feedback.

- Recommenders for the component of a particular bug are more useful than recommenders for the best person to fix.
- The time to fix is often seen as the time between opening and closing a bug. A more actionable definition is the time between a developer acknowledging that he will fix a bug and the time he submits it for review.

- Accuracy (precision / recall in all its variations) is not the primary criterion to be optimised. Equally important is the learning curve, i. e. how many bug reports you need for training before the recommender achieves a reasonable accuracy.

5.6 Rationalism vs. Empiricism in Software Engineering

Premkumar T. Devanbu (University of California – Davis, US)

Researchers in software engineering have each subscribed almost exclusively to one of two approaches: Rationalist, and Empiricist. This talk is a plea for more overlap and interaction, specially from the latter to the former.

5.7 A tale of two datasets

Georgios Gousios (TU Delft, NL)

Main reference G. Gousios, "The GHTorent dataset and tool suite," in Proc. of the 10th Working Conference on Mining Software Repositories (MSR'13), pp. 233–236, IEEE/AMC, 2013.
URL http://dl.acm.org/citation.cfm?id=2487085.2487132

What drives reuse of shared research artifacts? In my talk, I argue that the openness of the construction process plays a central role in the dissemination of research artifacts and their acceptance and trust by other researchers.

5.8 The trouble with performance analytics

Abram Hindle (University of Alberta, CA)

URL http://softwareprocess.es

Performance Analytics are continually challenged by the lack performance information within existing software repositories. The lack of logging by developers and users leads to this problem. Until developers are motivated by better analytics and tools, they will not be motivated to log performance information and aggregate it. It is up to us as researchers to generate this data, demonstrate how great the tools and analytics are with this information, and then motivate developers to log this information in the hope of using our methods and tools.

5.9 Applying Qualitative Analytics

Reid Holmes (University of Waterloo, CA)

Joint work of Baysal Olga; Holmes, Reid; Godfrey, Mike
Main reference O. Baysal, R. Holmes, M. Godfrey, "No Issue Left Behind: Reducing Information Overload in Issue Tracking," in Proc. of the 22nd ACM SIGSOFT Int'l Symposium on the Foundations of Software Engineering (FSE'14), 2014, to appear.

Modern software development processes generate large amounts of metadata. While it is tempting to aggregate this data in forms that are amenable to graphical representations (e. g., charts of defects fixed over time), these representations are not useful for developers as they address their day-to-day tasks.

This talk describes a research project that examined the kinds of questions industrial developers want to answer using modern issue tracking systems along with other questions they would like to ask but are unable to ask using existing tools. Ultimately we find that developers want support for expressing queries for addressing specific tasks along with maintaining overall situational awareness of both their own issues along with other issues that are relevant to their interests. We have created a model of these information needs and have built a prototype tool called Dash that fulfills these shortcomings by providing a qualitative (rather than quantitative) projection of issue tracker data. We have iterated on this tool several times with industrial developers and it has currently been deployed within Mozilla so we can gather longitudinal usage data to validate and improve our information model.

5.10 Information > Tool or Information → Tool?

Miryung Kim (University of Texas – Austin, US)

Main reference M. Kim, T. Zimmermann, N. Nagappan, "An Empirical Study of RefactoringChallenges and Benefits at Microsoft," Trans. on Software Engineering, 40(7):633–649, 2014.
URL http://dx.doi.org/10.1109/TSE.2014.2318734

In this talk, I present my work on systematic changes with the aim of having a discussion on whether information produced by development analytics is more important than building development tools or whether we should think about deriving information that could inform the design of development tools.

References

1 Kim, M.; Zimmermann, T.; Nagappan, N., "An Empirical Study of Refactoring Challenges and Benefits at Microsoft," IEEE Transactions on Software Engineering, 40(7):633–649, DOI: 10.1109/TSE.2014.2318734.

5.11 Thoughts on selling software analytics to software companies

Andrew J. Ko (University of Washington – Seattle, US)

I co-founded a company that provides contextual crowdsourced help to SaaS companies, providing a stream of insights about the questions, confusions, and struggles that users are having on their site. In interacting with customers, I have found that very few customers have mature practices or understanding of metrics, measurement, hypotheses, or experiments. This heavily skews what can be sold, how customers perceive value, and what types of analytics they understand and desire.

5.12 An ARCADE for Architecture Analytics

Nenad Medvidovic (University of Southern California, US)

From its very inception, the study of software architecture has recognized architectural decay as a regularly-occurring phenomenon in long-lived systems. At the same time, there is a relative dearth of empirical data about the nature of architectural change and the actual extent of decay in existing systems. In this talk, I present a workbench developed to help us take a step toward addressing that scarcity, by automating the study of architectural change and decay. The workbench, ARCADE ("Architecture Recovery, Change, and Decay Evaluator") enabled us to conduct a pilot study of the evolution of software architectures from several hundred versions belonging to 12 open- source systems totalling over 112 million source lines of code. To lay the groundwork for ARCADE, we previously performed an extensive evaluation of state-of-the-art techniques for obtaining a software architecture from a system's implementation and (2) cataloged symptoms of architectural decay. This talk reported on several results from the study, which revealed a number of unexpected findings regarding the frequency of architectural changes in software systems, the rate at which architectural decay occurs, the nature of architectural decay, and its relationship to code-level decay.

5.13 Software Effort Estimation Models – Past, Present and Future

Leandro L. Minku (University of Birmingham, GB)

In this talk, I briefly go through some key points in terms of the past, present and future of software effort estimation models. I go from (1) conclusion instability to (2) ensembles [1, 3, 4] and locality [2, 3] to (3) the importance of concentrating more on temporal [5] and cross-company learning [5, 6], generating insights [5], and obtaining a better understanding of when, why and how our models work (or don't work). Even though the talk is in the

context of software effort estimation models, several of these ideas are also applicable to other types of software prediction models, such as defect predictors.

References

1 L.L. Minku and X. Yao. Software Effort Estimation as a Multi- objective Learning Problem, 22(4):35, ACM TOSEM 2013.
2 T. Menzies et al. Local versus Global Lessons for Defect Prediction and Effort Estimation. 39(6):822–834, IEEE TSE 2013.
3 L.L. Minku and X. Yao. Ensembles and Locality: Insight on Improving Software Effort Estimation, 55(8):1512–1528, IST 2013.
4 Y. Kultur, B. Turhan and A. Bener. Ensemble of Neural Networks with Associative Memory (ENNA) for Estimating Software Development Costs, 22(6):395–402, KBS 2009.
5 L.L. Minku and X. Yao. How to Make Best Use of Cross- company Data in Software Effort Estimation?, pp. 446–456, ICSE 2014.
6 E. Kocaguneli, T. Menzies and E. Mendes. Transfer Learning in Effort Estimation. ESE 2014 (in press).

5.14 Operational Data are not Experimental Data

Audris Mockus (Avaya – Basking Ridge, US)

Main reference A. Mockus, "Engineering big data solutions," in Proc. of the "Future of Software Engineering" (FOSE) track at the 36th Int'l Conf. on Software Engineering (ICSE'14), pp. 85–99, ACM, 2014; pre-print available at author's webpage.
URL http://dx.doi.org/10.1145/2593882.2593889
URL http://mockus.org/papers/BigData.pdf

The collection and use of low-veracity data in software repositories and other operational support systems is exploding. It is, therefore, imperative to elucidate basic principles of how such data comes into being and what it means. Are there practices of constructing software data analysis tools that could raise the integrity of their results despite the problematic nature of the underlying data? The talk explores the basic nature of data in operational support systems and considers approaches to develop engineering practices for software mining tools.

References

1 Audris Mockus. Engineering big data solutions. In *ICSE'14 FOSE*, 2014.
2 Audris Mockus. Missing data in software engineering. In J. Singer et al., editor, *Guide to Advanced Empirical Software Engineering*, pages 185–200. Springer-Verlag, 2008.
3 Audris Mockus. Software support tools and experimental work. In V Basili and et al., editors, *Empirical Software Engineering Issues: Critical Assessments and Future Directions*, volume LNCS 4336, pages 91–99. Springer, 2007.

5.15 Analytics on Ad Library Maintenance in Android Apps

Meiyappan Nagappan (Rochester Institute of Technology, US)

Main reference I. J. Mojica Ruiz, M. Nagappan, B. Adams, T. Berger, S. Dienst, A. E. Hassan, "On the Relationship between the Number of Ad Libraries in an Android App and its Rating," IEEE Software, published online, 1 page, 2014.
URL http://dx.doi.org/10.1109/MS.2014.79

With more than 75% of mobile apps today being free-to-download, advertisement within apps is one of the key business models to generate revenue. Advertisements are served through the embedding of specialized code, i. e., ad libraries. Unlike other types of libraries, developers cannot ignore new versions of the embedded ad libraries or new ad libraries without risking a loss in revenue. However, updating ad libraries also has expenses, which can become a major problem as ad library updates are becoming more prevalent in mobile apps.

We mined over 128,000 Android apps over 12 months. After removing apps that were considered as noise, an analysis of 13,983 versions of 5,937 Android apps shows that almost half (48.98%) of the studied versions had an ad library update (i. e., ad library was added, removed, or updated). Interestingly, in 13.75% of app updates (new version in the Google Play store) with at least one case of ad library update, we found no changes to the app's own API, which suggests substantial additional effort for developers to maintain ad libraries. We also explore the rationales for why such updates are carried out. Finally, we find no evidence that the number of ad libraries in an app is related to the ratings that an app can get. However, integrating certain specific ad libraries can negatively impact the rating of an app.

5.16 Are We Really Helping Developers?

Alessandro Orso (Georgia Institute of Technology, US)

Joint work of Parnin, Chris; Orso, Alessandro
Main reference C. Parnin, A. Orso, "Are Automated Debugging Techniques Actually Helping Programmers?" in Proc. of the Int'l Symp. on Software Testing and Analysis (ISSTA"11), pp. 199–209, ACM, 2011.
URL http://dx.doi.org/10.1145/2001420.2001445

This talk discusses some of the risks involved in defining and evaluating software engineering approaches without considering how (or whether) developers will use and benefit from them. As a specific example, the talks presents a human study that shows how a family of techniques that received a great deal of attention in the last decade seemed to be ineffective when evaluated on real users.

5.17 My flings with data analysis

Venkatesh-Prasad Ranganath (Kansas State University, US)

Joint work of Ranganath, Venkatesh-Prasad; Vallathol, Pradip; Gupta, Pankaj
Main reference V.-P. Ranganath, P. Vallathol, P. Gupta, "Compatibility Testing via Patterns-Based Trace Comparison," in Proc. of the 29th ACM/IEEE Int'l Conf. on Automated Software Engineering (ASE'14), pp. 469–478, ACM, 2014; pre-print available from author's webpage.
URL http://dx.doi.org/10.1145/2642937.2642942
URL http://research.microsoft.com/apps/pubs/?id=171616

This talk presents observations and opportunities in the space of using data analysis to enable, accomplish, and improve software engineering tasks such as testing. The observations and opportunities are drawn from efforts in an industrial setting.

5.18 Towards the Impact of Software Analytics

Guenther Ruhe (University of Calgary, CA)

Joint work of Ruhe, Guenther; Nayebi, Maleknaz; S M D. Al Alam
Main reference M. Nayebi, G. Ruhe, "Analytical Product Release Planning," to appear in C. Bird, T. Menzies, T. Zimmermann, eds., "The Art and Science of Analyzing Software Data", Morgan-Kaufmann.

This position statement raises some questions related to the impact of software data analytics. Questions being raised are:

- How much of the analytics results actually have been used?
- Which actual decisions have been supported?
- How much of the results was useful?
- How much data analytics is enough?

Two examples of ongoing research are presented:

1. Software release readiness
2. Analytical product release planning

5.19 Mere Numbers aren't Enough – Focus on Interpretation and Visualization

Per Runeson (Lund University, SE)

Software analytics involve data collection, analysis, interpretation and visualization. Current research focus to a vast majority on the data and analysis. However, interpretation and visualization are more related to the use and utility of the software analytics. Through a few examples [2, 3], I show how the interpretation and visualization parts play a significant role, and I encourage to take these aspects into account in future research [1].

This talk is based on the following publications:

References

1 Hassan, A. E., A. Hindle, P. Runeson, M. Shepperd, P. Devanbu, and S. Kim (2013). *Roundtable: What's Next in Software Analytics*. IEEE Software 30(4), 53–56.

2 Borg, M., P. Runeson, and A. Ardö(2013). *Recovering from a Decade: A Systematic Map of Information Retrieval Approaches to Software Traceability.* Empirical Software Engineering. DOI: 10.1109/MS.2006.147.

3 Engström, E., M. Mäntylä, P. Runeson, and M. Borg (2014). *Supporting Regression Test Scoping with Visual Analytics*. In: Proceedings of the 2014 IEEE International Conference on Software Testing, Verification, and Validation, pp.283-292. DOI: 10.1109/ICST.2014.41.

5.20 Composable Data Mining: Supporting Analytics for End Users

Anita Sarma (University of Nebraska – Lincoln, US)

Joint work of Sarma, Anita; Jose Ricardo da Silva, Leonardo Murta

There is a need for supporting end users in performing analytics of their own data. We provide a framework that translates the relationship among project elements into matrices and allows linear transformation of these matrices to combine relationships and provide insight. This talk shows how Dominoes can be used to identify expertise for a given project or an artifact (file) by considering not only the number of edits that a developer has made, but also the spread of their changes and thereby the breadth of their expertise. Our approach enables us to identify expertise over any given granularity and time period quickly.

5.21 42 years of Unix history in one repository

Diomidis Spinellis (Athens University of Economics and Business, GR)

The goal of the Unix history repository project is to create a git repository representing the Unix source code history, starting from the early 1970s and ending in the modern time. To fulfil this goal the project brings data from early system snapshots, repositories, and primary research. The project aims to put in the repository as much meta-data as possible, allowing the automated analysis of Unix history. This effort allows the exploration of programming style evolution, the consolidation of digital artefacts of historical importance, the collection and recording of history that is fading away, and the provision of a data set for digital archaeology and repository mining. The project has achieved its first major goal with the establishment of a continuous time-line from 1972 to 2014. The repository contains snapshots of V1, V3, V4, V5, V6, and V7 Research Edition, Unix/32V, all available BSD releases, the CSRG SCCS history, two releases of 386BSD, FreeBSD 1.0, and an import of the FreeBSD repository starting from its initial imports that led to FreeBSD 2.0. The files appear to be added in the repository in chronological order according to their modification (or commit) time, and large parts of the source code have been attributed to their actual authors. Commands such as git blame and (sometimes) git log produce the expected results. The community can contribute to the project by using it for research, adding material, and proposing corrections. The repository is available online at https://github.com/dspinellis/unix-history-repo.

5.22 Open Problems and Challenges in Software Analytics

Diomidis Spinellis (Athens University of Economics and Business, GR)

We identify open problems and challenges in software analytics in the areas of the domains that can be addressed, data analysis, and under-represented stakeholders. In terms of domains, we refer to the recent paper by Begel and Zimmermann on 145 questions for data scientists in software engineering, and outline a procedure that can be used to map these questions into domain challenges. In terms of data analysis, we identify the problems and challenges of linking persons with their actions in repositories, issue databases, mailing lists, and social networking sites, linking artefacts between systems that hold them (e. g. commits with issues they resolve), scalability of tools and techniques over large data sets, the sharing of industrial data, data privacy, the cleaning of noise, judging and dealing with data quality, judging the representativeness of data, and reproducing results. In terms of stakeholders, we highlight that build engineers, system administrators, people with multiple tasks in a software project, software designers/architects, and business/product managers, and support personnel are not well catered by current data analysis techniques.

References

1 Andrew Begel, Thomas Zimmermann. Analyze This! 145 Questions for Data Scientists in Software Engineering. In Proceedings of the *36th International Conference on Software Engineering (ICSE 2014)*, Hyderabad, India, June 2014.

5.23 Studying social media in software development: reflecting on research methods

Margaret-Anne Storey (University of Victoria, CA)

I present a brief overview of research on the impact of social media on software engineering. This research highlights how software engineers today actively benefit from a participatory culture of development, that is fuelled by socially enabled tools. I also provide a brief overview of the landscape of research methods, highlighting some methods in particular: grounded theory and mixed methods. As I describe these methods, I reflect on my experiences using those methods to investigate and learn about social media use in software engineering. Finally, I suggest that we try to adopt some of the ways open source developers benefit from their participatory culture, so that we can improve and accelerate the research we do. By collaborating more, we are more likely to be able to use multiple methods to investigate software development which will help balance the different limitations of methods.

5.24 The Graph

Burak Turhan (University of Oulu, FI)

Joint work of Turhan, Burak; Taipele, Taneli; Qvist, Mika; Kuutti, Kari
Main reference T. Taipale, B. Turhan, M. Qvist, "Constructing Defect Predictors and Communicating the Outcomes to Practitioners", in Proc. of the 7th Int'l Symp. on Empirical Software Engineering and Measurement (ESEM'13, Industry Track), pp. 357–362, IEEE, 2013.
URL http://dx.doi.org/10.1109/ESEM.2013.45

In this talk I'll share our experiences in conducting a software analytics project, e. g. bug prediction. Though the project was a huge success in terms of scholarly outcomes and performance measures, we had difficulty in communicating the results to the practitioners. It turned out that our predictions were perceived as stating the obvious – even through many different modes of communication/ representation; and the most useful and insightful representation was only a visualization of the issue reports without any predictions. Hence, exploration trumps prediction in our case!

5.25 Why Quality Models Don't Work and Why We Need Them Anyway

Stefan Wagner (Universität Stuttgart, DE)

Joint work of Wagner, Stefan; Lochmann, Klaus; Heinemann, Lars; Kläs, Michael; Trendowicz, Adam; Plösch, Reinhold; Seidl, Andreas; Goeb, Andreas; Streit, Jonathan
Main reference S. Wagner, K. Lochmann, L. Heinemann, M. Kläs, A. Trendowicz, R. Plösch, A. Seidl, A. Goeb, J. Streit, "The Quamoco Product Quality Modelling and Assessment Approach," in Proc. of 34th Int'l Conf. on Software Engineering (ICSE'12), pp. 1133–1142, IEEE, 2012
URL http://dx.doi.org/10.1109/ICSE.2012.6227106

Quality is a complex and multifaceted concept. We employ quality models to capture this complexity. In the project Quamoco, we built a detailed and operationalised quality model connecting low-level metrics with high-level quality attributes. We managed to build a model judged well understandable by developers and giving reasonable quality estimates. Yet, the effort that went into it is prohibitive for a real general model. Also, the many factors in the model makes it almost impossible to really validate it. So where should we go? Staying with predicting other low-level metrics (such as defects) or put in the effort to describe the relationships to high-level attributes?

5.26 Analytics in the Field

Patrick Wagstrom (IBM Watson Group, US)

As researchers we often develop elegant tools and models that explain phenomena behind software engineering. These models are purported to make the development process faster, easier, and more reliable. However, these research outputs are rarely taken up in industry. This talk, presented from the industry perspective, identifies some of the challenges around deploying research output in industry from the lens of understanding user interactions and desires.

5.27 Software Analytics in Practice

Dongmei Zhang (Microsoft Research – Beijing, CN)

Joint work of Zhang, Dongmei; Shi Han, Yingnong Dang, Jian-Guang Lou, Haidong Zhang, Tao Xie
Main reference D. Zhang, S. Han, Y. Dang, J.-G. Lou, H. Zhang, T. Xie, "Software Analytics in Practice," IEEE Software – Special Issue on the Many Faces of Software Analytics, 30(5):30–37, 2013; pre-print available from author's webpage.
URL http://dx.doi.org/10.1109/MS.2013.94
URL http://research.microsoft.com/en-us/groups/sa/ieeesoft13-softanalytics.pdf

In this short talk, I'll introduce our definition of software analytics, and explain the definition from five perspectives – research topics, target audience, input/output, technology pillars, and connection to practice. In particular, I'll discuss the importance of connection to practice, because (1) the data under study comes from real practice; (2) there are real problems to be answered using the data; (3) one of the success metrics of software analytics research is its influence and impact on the development practice.

6 Breakout Groups

We will now briefly describe six breakout sessions that were held. The materials produced by the groups are archived at http://www.dagstuhl.de/mat/index.en.phtml?14261.

- **Sharing data, methods, and models.** The breakout did a SWOT (Strength, Opportunity, Weakness and Threats) analysis of data, method and model sharing. In general, all sharing was deemed beneficial for sharing the workload, generalizability, validation, and collaboration. However, putting the shared artifacts in context was a threat in all cases.
- **Industry collaboration.** The breakout focused on issues related to this important collaboration. It was recognized that each of industry and research have different needs but working together is essential and beneficial. For example, research desires statistical significance while this is not a concern to industry. Trust must be established between both parties. Potential legal issues to enable the collaboration and data sharing may arise.
- **Development of a software analytics development community.** The breakout group formed a GitHub organization to maintain lists of Software Development Analytics Community blogs and other resources.
- **Tools and techniques.** Tools for quantitative and qualitative methods were discussed. It was suggested to create a wiki for the classification of methods, questions asked, good examples of case studies, pitfalls.
- **Analytics Framework.** The group discussed a spreadsheet to identify dimensions of analytics, to identify questions asked for each cell in the n-dimensional framework and to link papers that address those questions
- **Manifesto.** Significant time was spent to develop a data analytics manifesto, as will be discussed in the next section.

Another outcome of a breakout session were "Good Practices for Software Analytics Papers", which can be found at this URL: http://www.cs.mcgill.ca/~martin/blog/2014-06-26.html

Participants

- Bram Adams
Polytechnique Montreal, CA
- Alberto Bacchelli
TU Delft, NL
- Ayse Bener
Ryerson Univ. – Toronto, CA
- Trevor Carnahan
Microsoft Res. – Redmond, US
- Serge Demeyer
University of Antwerp, BE
- Premkumar T. Devanbu
Univ. of California – Davis, US
- Stephan Diehl
Universität Trier, DE
- Michael W. Godfrey
University of Waterloo, CA
- Alessandra Gorla
Universität des Saarlandes – Saarbrücken, DE
- Georgios Gousios
TU Delft, NL
- Mark Grechanik
Univ. of Illinois – Chicago, US
- Michaela Greiler
Microsoft Res. – Redmond, US
- Abram Hindle
University of Alberta, CA
- Reid Holmes
University of Waterloo, CA
- Miryung Kim
University of Texas – Austin, US
- Andrew J. Ko
University of Washington – Seattle, US
- Lucas M. Layman
Fraunhofer USA, US
- Andrian Marcus
Wayne State University, US
- Nenad Medvidovic
Univ. of Southern California, US
- Tim Menzies
West Virginia University – Morgantown, US
- Leandro L. Minku
University of Birmingham, GB
- Audris Mockus
Avaya – Basking Ridge, US
- Brendan Murphy
Microsoft Res. – Cambridge, GB
- Meiyappan Nagappan
Rochester Institute of Technology, US
- Alessandro Orso
Georgia Inst. of Technology, US
- Martin Pinzger
Universität Klagenfurt, AT
- Denys Poshyvanyk
College of William and Mary – Williamsburg, US
- Venkatesh-Prasad Ranganath
Kansas State University, US
- Romain Robbes
University of Chile, CL
- Martin Robillard
McGill University, CA
- Guenther Ruhe
University of Calgary, CA
- Per Runeson
Lund University, SE
- Anita Sarma
Univ. of Nebraska – Lincoln, US
- Emad Shihab
Concordia Univ. – Montreal, CA
- Diomidis Spinellis
Athens University of Economics and Business, GR
- Margaret-Anne Storey
University of Victoria, CA
- Burak Turhan
University of Oulu, FI
- Stefan Wagner
Universität Stuttgart, DE
- Patrick Wagstrom
IBM Watson Group, US
- Jim Whitehead
University of California – Santa Cruz, US
- Laurie Williams
North Carolina State Univ., US
- Dongmei Zhang
Microsoft Res. – Asia, CN
- Thomas Zimmermann
Microsoft Res. – Redmond, US

Report from Dagstuhl Seminar 14271

Scripting Languages and Frameworks: Analysis and Verification

Edited by

Fritz Henglein[1], Ranjit Jhala[2], Shriram Krishnamurthi[3], and Peter Thiemann[4]

1 **University of Copenhagen, DK,** henglein@diku.dk
2 **University of California – San Diego, US,** jhala@cs.ucsd.edu
3 **Brown University – Providence, US,** sk@cs.brown.edu
4 **Universität Freiburg, DE,** thiemann@informatik.uni-freiburg.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14271 "Scripting Languages and Frameworks: Analysis and Verification". The seminar brought together a broad spectrum of researchers working on the semantics, analysis and verification of scripting languages. In addition to talks describing the latest problems and research on the key issues, split roughly into four overarching themes: semantics, types, analysis, contracts, languages, and security, the seminar had breakout sessions devoted to crosscutting topics that were of broad interest across the community, including, how to create shared analysis infrastructure, how to think about the semantics of contracts and blame, and the role of soundness in analyzing real world languages, as well as several "tutorial" sessions explaining various new tools and techniques.

Seminar July 1–4, 2014 – http://www.dagstuhl.de/14271
1998 ACM Subject Classification D.3.3 Programming Languages, F.3.1 Logics and Meanings of Programs
Keywords and phrases Scripting Languages, Frameworks, Contracts, Types, Analysis, Semantics
Digital Object Identifier 10.4230/DagRep.4.6.84

1 Executive Summary

Fritz Henglein
Ranjit Jhala
Shriram Krishnamurthi
Peter Thiemann

In the past decade scripting languages have become more mature: the wild experimentation and almost wilful embrace of obfuscation by Perl has been replaced by the level-headed simplicity of Python and the embrace of programming language research roots by Ruby. As a result, these languages have moved into the mainstream: every Web user relies on JavaScript.

The Challenges of Scripting Languages Though scripting languages have become more mature, from the perspective of building robust, reliable software, they still suffer from several distinct problems, each of which creates new challenges for the research community.

- While these languages have textual definitions, they lack more formal descriptions, and in practice the textual "definitions" are themselves often in conflict with the normative

nature of the implementations. This is in contrast to languages like Standard ML where the formal definition comes first. *How far can we go in creating formal semantics from a combination of implementations and textual documents?*

- Tests – more than either implementations, textual definitions, or formal semantics – are becoming the norm for specification. For instance, the latest JavaScript standard explicitly embraces testing by publishing and regularly updating a conformance suite. Similarly, a team trying to create an alternate implementation of one of these languages may read the definition but what they really aspire to match is the test suite behavior. *How can we support test suites as a new avenue of programming language specification?*
- One of the reasons programmers find these languages enjoyable (initially) is that they offer a variety of "convenient" features, such as overloading. As programs grow, however, understanding the full – and unintended! – behaviors of programs becomes a non-trivial effort. *How can we design semantics and static and dynamic tools that can cope with the heavily understated and overloaded behaviors that make scripting languages attractive?*
- Programmers increasingly do not program in languages but in high-level frameworks built atop them. For instance, though "Ruby" is popular for Web programming, programmers rarely write Web applications directly in Ruby, but rather atop the higher-level Ruby on Rails platform. The result of imposing significantly higher-level interfaces is that they necessitate new reasoning modes. For instance, while the jQuery library is a pure JavaScript program, type-checking jQuery as if it were "merely" JavaScript would produce types that are both unreadably compex and relatively useless. *Can we build custom reasoning at the level of the frameworks, then we can provide views of these frameworks that are consistent with the level at which developers think of them, and can we check that the implementations adhere to these interfaces?*
- These languages and frameworks are themselves not enough. They all reside in an eco-system of a family of other languages and frameworks whose interdependencies are necessary for proper understanding of program execution. For instance, in the client-side Web, JavaScript – which has gotten significant attention from the research community – only runs in response to stimuli, which are obtained from the DOM. In turn, the DOM and JavaScript both depend on the style-sheets written in CSS. But in fact all three of these components – the JavaScript code, the CSS styling, and the DOM events – all depend on one another, because almost any one can trigger or modify the other. *Can we construct suitable abstractions such that each language can meaningfully talk about the others without importing an overwhelming amount of detail?*

This seminar brought together a wide variety of researchers working on the above questions. The seminar was organized into a series of short and long talks on topics related to the above overarching questions, and four breakout sessions focussing on broader questions and challenges. Next, we briefly summarize the talks and sessions. The contributed talks focussed on the following over arching themes – *semantics*, *type systems*, *program analysis*, *contracts*, *languages* and *security*.

2 Table of Contents

3 Overview of Talks: Semantics

3.1 Python, the Full Monty

Joe Gibbs Politz (Brown University – US)

Joint work of Krishnamurthi, Shriram; Politz, Joe Gibbs

We present a small-step operational semantics for the Python programming language. We present both a core language for Python, suitable for tools and proofs, and a translation process for converting Python source to this core. We have tested the composition of translation and evaluation of the core for conformance with the primary Python implementation, thereby giving confidence in the fidelity of the semantics. We briefly report on the engineering of these components. Finally, we examine subtle aspects of the language, identifying scope as a pervasive concern that even impacts features that might be considered orthogonal.

3.2 An Executable Formal Semantics of PHP

Daniele Filaretti (Imperial College London, GB)

Joint work of Filaretti, Daniele; Maffeis, Sergio
Main reference D. Filaretti, S. Maffeis, "An Executable Formal Semantics of PHP," in Proc. of the 28th Europ. Conf. Object-Oriented Programming (ECOOP'14), LNCS, Vol. 8586, pp. 567–592, Springer, 2014.
URL http://dx.doi.org/10.1007/978-3-662-44202-9_23
URL https://dfilaretti.files.wordpress.com/2014/02/dagstuhl2014.pdf

We describe the first executable formal semantics of a substantial core of PHP – validated by testing against the Zend Test suite.

3.3 JSCert, a two-pronged approach to JavaScript formalization

Alan Schmitt (INRIA Bretagne Atlantique – Rennes, FR)

Main reference M. Bodin, A. Charguéraud, D. Filaretti, P. Gardner, S. Maffeis, D. Naudziuniene, A. Schmitt, G. Smith, "A Trusted Mechanised JavaScript Specification," in Proc. of the 41st ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages (POPL'14), pp. 87–100, ACM, 2014.
URL http://dx.doi.org/10.1145/2535838.2535876

JSCert is a formalization of JavaScript that aims at being as close as possible to the specification while having an executable component to run against test suites.

4 Overview of Talks: Type Systems

4.1 Progressive Types

Joe Gibbs Politz (Brown University – US)

Joint work of Krishnamurthi, Shriram

As modern type systems grow ever-richer, it can become increasingly onerous for programmers to satisfy them. However, some programs may not require the full power of the type system, while others may wish to obtain these rich guarantees incrementally. In particular, programmers may be willing to exploit the safety checks of the underlying run- time system as a substitute for some static guarantees. Progressive types give programmers this freedom, thus creating a gentler and more flexible environment for using powerful type checkers. In this paper we discuss the idea, motivate it with concrete, real-world scenarios, then show the development of a simple progressive type system and present its (progressive) soundness theorem.

4.2 Safe TypeScript

Panagiotis Vekris (University of California – San Diego, US)

Joint work of Rastogi, Aseem; Swamy, Nikhil; Fournet, Cedric; Bierman, Gavin; Vekris, Panagiotis

Safe TypeScript is a gradual type system built on top of the TypeScript compiler framework that achieves type soundness by means of stricter static typing rules and a runtime mechanism for checks lying on the boundary between static and dynamic types. Safe TypeScript is geared towards efficiency: it uses differential subtyping, whereby only a minimum amount of runtime annotations are applied; and provides an erasure modality, which enables selective deletion of type annotations for type constructs that are meant to be dealt with entirely statically. The implemented Safe TypeScript compiler has been succefully used on hundreds of lines of existing TypeScript code, incurring with a modest overhead on sufficiently annotated input code.

4.3 Confined Gradual Typing

Éric Tanter (University of Chile, CL)

Joint work of Allende, Esteban; Fabry, Johan; Garcia, Ronald; Tanter, Éric
Main reference E. Allende, J. Fabry, R. Garcia, É. Tanter, "Confined Gradual Typing," in Proc. of the 2014 ACM Int'l Conf. on Object Oriented Programming Systems Languages & Applications (OOPSLA'14), pp. 251–270, ACM, 2014; pre-print available from author's webpage.
URL http://dx.doi.org/10.1145/2660193.2660222
URL http://pleiad.dcc.uchile.cl/papers/2014/allendeAl-oopsla2014.pdf

Gradual typing combines static and dynamic typing flexibly and safely in a single programming language. To do so, gradually typed languages implicitly insert casts where needed, to ensure

at runtime that typing assumptions are not violated by untyped code. However, the implicit nature of cast insertion, especially on higher-order values, can jeopardize reliability and efficiency: higher-order casts can fail at any time, and are costly to execute. We propose Confined Gradual Typing, which extends gradual typing with two new type qualifiers that let programmers control the flow of values between the typed and the untyped worlds, and thereby trade some flexibility for more reliability and performance. We formally develop two variants of Confined Gradual Typing that capture different flexibility/guarantee tradeoffs. We report on the implementation of Confined Gradual Typing in Gradualtalk, a gradually-typed Smalltalk, which confirms the performance advantage of avoiding unwanted higher-order casts and the low overhead of the approach.

4.4 Typing Scheme to Typing Racket

Sam Tobin-Hochstadt (Indiana University – Bloomington, US)

Joint work of Tobin-Hochstadt, Sam; Takikawa, Asumu; Felleisen, Matthias; Strickland, T. Stephen
Main reference A. Takikawa, T. S. Strickland, C. Dimoulas, S. Tobin-Hochstadt, M. Felleisen, "Gradual typing for first-class classes," in Proc. of the 27th Annual ACM SIGPLAN Conf. on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'12), pp. 793–810, ACM, 2012; pre-print available from author's webpage.
URL http://www.ccs.neu.edu/racket/pubs/oopsla12-tsdthf.pdf

We have extended Typed Racket extensively to include support for features that go beyond traditional Scheme, including first-class classes, delimited continuations, mixins, etc.

4.5 Type Systems for JavaScript: Variations on a Theme

Benjamin Lerner (Brown University, US)

Joint work of Lerner, Benjamin; Politz, Joe G.; Guha, Arjun; Krishnamurthi, Shriram
Main reference B. S. Lerner, J. G. Politz, A. Guha, S. Krishnamurthi, "TeJaS: retrofitting type systems for JavaScript," in Proc. of the 9th Symp. on Dynamic Languages (DLS '13), pp. 1–16, ACM, 2013.
URL http://dx.doi.org/10.1145/2508168.2508170

When JavaScript programmers write code, they often target not just the base language but also libraries and API frameworks that drastically change the style of their programs, to the point where they might well be considered as written in domain-specific languages rather than merely JS. Accordingly, the characteristic bugs for such applications varies by domain, and so any tools designed to help developers catch these bugs ought to be tailored to the domain. Yet these tools likely share a common core, since the underlying language is still JS.

We present a TeJaS, a framework for designing type systems for JavaScript that can be customized to analyze the idiomatic errors of various domains, and we illustrate its utility by describing systems for analyzing DOM-access errors in jQuery programs, and privacy violations in Firefox browser extensions running in private-browsing mode.

4.6 Flow Typing

Arjun Guha (University of Massachusetts – Amherst, US)

Joint work of Guha, Arjun; Saftiou, Claudiu; Krishnamurthi, Shriram
Main reference A. Guha, C. Saftoiu, S. Krishnamurthi, "Typing Local Control and State Using Flow Analysis," in Proc. of the 20th Europ. Symp. on Programming (ESOP'11), LNCS, Vol. 6602, pp. 256–275, Springer, 2011.
URL http://dx.doi.org/10.1007/978-3-642-19718-5_14

Programs written in scripting languages employ idioms that confound conventional type systems. In this paper, we highlight one important set of related idioms: the use of local control and state to reason informally about types. To address these idioms, we formalize run-time tags and their relationship to types, and use these to present a novel strategy to integrate typing with flow analysis in a modular way. We demonstrate that in our separation of typing and flow analysis, each component remains conventional, their composition is simple, but the result can handle these idioms better than either one alone.

4.7 Types for Ruby

Jeffrey Foster (University of Maryland, US)

This talk summarizes several years of work on ways to bring some of the benefits of static typing to Ruby. We discuss Diamondback Ruby, a pure static type inference system for Ruby; an extension that does profiling to account for highly dynamic language features; the Mix system, which combines type checking and symbolic execution; and, briefly, RubyDust and rtc, which use the ideas of Mix to provide type inference and checking, respectively, at run time for Ruby.

4.8 Refinement Types for an Imperative Scripting Language

Panagiotis Vekris (University of California – San Diego, US)

Joint work of Jhala, Ranjit

We present a refinement type checker for a scripting language employing various idioms of the JavaScript/TypeScript language family. Our type system consists of a base type system that includes, among others, object types, unions, intersection and higher order functions. On top of this base system lies our refinement type system whose language spans linear arithmetic and uninterpreted predicates. Subtyping on the base system is coercive and the casts added during base typechecking are expressed in the form of refinement type constraints along side value related constraints. These constraints are formulated into logical implications and are discharged by means of Liquid Types inference/checking. Examples outlined in this presentation include safe downcasts based on reflection and in-bounds array accesses.

4.9 Late Typing for Loosely Coupled Recursion

Ravi Chugh (University of California – San Diego, US)

URL http://goto.ucsd.edu/~ravi/research/dagstuhl-late.pptx.pdf

Flexible patterns of mutual recursion can be encoded in scripting languages by defining component functions independently and then "tying the knot" either by mutation through the heap or explicitly passing around receiver objects. We present a mechanism called late typing to reason about such idioms. The key idea is to, first, augment function types with constraints that may not be satisfied when functions are defined and, second, to check that these constraints are satisfied by the time the functions are called.

5 Overview of Talks: Program Analysis

5.1 Abstract Domains for Analyzing Hash Tables

Matthew Might (University of Utah, US)

Hash-table-like abstractions pervade scripting languages as fundamental data structures. (Consider objects in JavaScript, dictionaries in Python and hashes in Ruby.) Attempts to model these abstractions with the same abstract domains used to model abstractions of objects in languages like Java (in which fields and methods are fixed upon allocation) breaks these domains so as to cause catastrophic loss in precision or unsoundness. This talk looks at what is required to retain soundness while more precisely modeling the flexible nature of these structures.

5.2 Static Analysis for Open Objects

Arlen Cox (Colorado University – Boulder, US)

Joint work of Rival, Xavier

In dynamic languages, objects are open – they support iteration over and dynamic addition/deletion of their attributes. Open objects, because they have an unbounded number of attributes, are difficult to abstract without a priori knowledge of all or nearly all of the attributes and thus pose a significant challenge for precise static analysis. To address this challenge, this talks presents the HOO (Heap with Open Objects) abstraction that can precisely represent and infer properties about open-object-manipulating programs without any knowledge of specific attributes. It achieves this by building upon a relational abstract domain for sets that is used to reason about partitions of object attributes. An implementation of the resulting static analysis is used to verify specifications for dynamic language framework code that makes extensive use of open objects, thus demonstrating the effectiveness of this approach.

5.3 Soft Contract Verification

David van Horn (University of Maryland – College Park, US)

Behavioral software contracts are a widely used mechanism for governing the flow of values between components. However, run-time monitoring and enforcement of contracts imposes significant overhead and delays discovery of faulty components to run-time.

To overcome these issues, we present soft contract verification, which aims to statically prove either complete or partial contract correctness of components, written in an untyped, higher-order language with first-class contracts. Our approach uses higher-order symbolic execution, leveraging contracts as a source of symbolic values including unknown behavioral values, and employs an updatable heap of contract invariants to reason about flow-sensitive facts. We prove the symbolic execution soundly approximates the dynamic semantics and that verified programs can't be blamed.

The approach is able to analyze first-class contracts, recursive data structures, unknown functions, and control-flow-sensitive refinements of values, which are all idiomatic in dynamic languages. It makes effective use of an off-the-shelf solver to decide problems without heavy encodings. The approach is competitive with a wide range of existing tools—including type systems, flow analyzers, and model checkers—on their own benchmarks.

5.4 Type Refinement for Static Analysis of JavaScript

Ben Weidermann (Harvey Mudd College, US)

Static analysis of JavaScript has proven useful for a variety of purposes, including optimization, error checking, security auditing, program refactoring, and more. A technique called type refinement that can improve the precision of such static analyses for JavaScript without any discernible performance impact. Refinement is a known technique that uses the conditions in branch guards to refine the analysis information propagated along each branch path. The key insight of this paper is to recognize that JavaScript semantics include many implicit conditional checks on types, and that performing type refinement on these implicit checks provides significant benefit for analysis precision.

5.5 Dynamic Determinacy Analysis

Manu Sridharan (Samsung Research, US)

Joint work of Schäfer, Max; Sridharan, Manu; Dolby, Julian; Tip, Frank
Main reference M. Schäfer, M. Sridharan, J. Dolby, F. Tip, "Dynamic determinacy analysis," in Proc. of the 34th ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI'13), pp. 165–174, ACM, 2013.
URL http://dx.doi.org/10.1145/2499370.2462168

Programs commonly perform computations that refer only to memory locations that must contain the same value in any program execution. Such memory locations are *determinate* because the value they contain is derived solely from constants. We present a dynamic program analysis that computes a safe approximation of the determinacy of the memory locations referenced at each program point. We implemented this determinacy analysis for JavaScript on top of the `node.js` environment. In two case studies, we demonstrate how the results of determinacy analysis can be used for improving the accuracy of a standard static pointer analysis, and for identifying calls to `eval` that can be eliminated.

5.6 Performance Analysis of JavaScript

Manu Sridharan (Samsung Research, US)

Performance analysis for JavaScript is increasingly important, but difficult due to fragile interactions with JIT compilers and complex native APIs like the DOM. We propose an approach to profiling memory behavior of JavaScript code via heavyweight, platform-independent dynamic tracing and offline analysis, and we outline open challenges with this approach.

5.7 Checking Correctness of TypeScript Interfaces for JavaScript Libraries

Anders Møller (Aarhus University, DK)

Joint work of Møller, Anders; Feldthaus, Asger
Main reference A. Feldthaus, A. Møller, "Checking correctness of TypeScript interfaces for JavaScript libraries," in Proc. of the 2014 ACM Int'l Conf. on Object Oriented Programming Systems Languages & Applications (OOPSLA'14), pp. 1–16, ACM, 2014.
URL http://dx.doi.org/10.1145/2660193.2660215

The TypeScript programming language adds optional types to JavaScript, with support for interaction with existing JavaScript libraries via interface declarations. Such declarations have been written for hundreds of libraries, but they can be difficult to write and often contain errors, which may affect the type checking and misguide code completion for the application code in IDEs.

We present a pragmatic approach to check correctness of TypeScript declaration files with respect to JavaScript library implementations. The key idea in our algorithm is that

many declaration errors can be detected by an analysis of the library initialization state combined with a light-weight static analysis of library function code.

Our experimental results demonstrate the effectiveness of the approach: it has found 142 errors in the declaration files of 10 libraries, with an analysis time of a few minutes per library and with a low number of false positives. Our analysis of how programmers use library interface declarations furthermore reveals some practical limitations of the TypeScript type system.

5.8 Analyzing JavaScript Web Applications in the Wild (Mostly) Statically

Sukyoung Ryu (KAIST – Daejeon, KR)

Analyzing real-world JavaScript web applications is a challenging task. On top of understanding the semantics of JavaScript, it requires modeling of web documents, platform objects, and interactions between them. Not only JavaScript itself but also its usage patterns are extremely dynamic. Most of web applications load JavaScript code dynamically, which makes pure static analysis approaches inapplicable. We present our attempts to analyze JavaScript web applications in the wild mostly statically using various approaches to analyze libraries.

6 Overview of Talks: Contracts

6.1 Membranes as Ownership Boundaries

Tom Van Cutsem (Alcatel-Lucent Bell Labs – Antwerp, BE)

We discuss the similarities and differences between membranes and higher-order contracts, give a brief overview of proxies in JS (which are the basic building block for membranes) and then show how membranes can be used to express the use cases typically expressed using ownership type systems.

6.2 TreatJS: Higher-Order Contracts for JavaScript

Matthias Keil (Universität Freiburg, DE)

Joint work of Keil, Matthias; Thiemann, Peter
URL http://www2.informatik.uni-freiburg.de/~keilr/talks/talk_dagstuhl2014-treatjs.pdf

TreatJS is a language embedded, dynamic, higher-order contract system for JavaScript. Beyond the standard abstractions for building higher-order contracts (base, function, and object contracts), TreatJS' novel contribution is its support for boolean combinations of

contracts and for the creation of parameterized contracts, which are the building blocks for dependent contracts and more generally run-time generated contracts.

TreatJS is implemented using JavaScript proxies to guarantee full interposition for contracts and it exploits JavaScript's reflective features to run contracts in a sandbox environment. This sandbox guarantees that contracts do not interfere with normal program execution. It also facilitates that all aspects of a contract are specified using the full JavaScript language. No source code transformation or change in the JavaScript run-time system is required.

TreatJS including sandboxing, is formalized and the impact of contracts on execution speed is evaluated in terms of the Google Octane benchmark.

6.3 Contracts for Domain-Specific Languages in Ruby

Jeffrey Foster (University of Maryland, US)

Joint work of Foster, Jeffrey; Strickland, T. Stephen; Ren, Bree

This talk concerns object-oriented embedded DSLs, which are popular in the Ruby community but have received little attention in the research literature. Ruby DSLs implement language keywords as implicit method calls to self; language structure is enforced by adjusting which object is bound to self in different scopes. We propose RDL, a new contract checking system that can enforce contracts on the structure of Ruby DSLs, attributing blame appropriately. We describe RDL and RDLInfer, a tool that infers RDL contracts for existing Ruby DSLs.

7 Overview of Talks: Languages

7.1 HOP: A Multi-tier Language For Web Applications

Tamara Rezk (INRIA Sophia-Antipolis, FR)

We present HOP a multi-tier language to write web applications. We propose a small-step operational semantics to support formal reasoning in HOP. The semantics covers both server side and client side computations, as well as their interactions, and includes creation of web services, distributed client-server communications, concurrent evaluation of service requests at server side, elaboration of HTML documents, DOM operations, evaluation of script nodes in HTML documents and actions from HTML pages at client side.

7.2 Perl: The Ugly Parts

Matthew Might (University of Utah, US)

Let there be no mistake: Perl is extremely useful. Every programmer needs Perl in their arsenal. Thanks to many implicit behaviors, some complex programs can be specified with alarming brevity. Perl excels at extracting and transforming data. But, Perl is as dangerous as it is ugly. This talk looks at the ugly.

7.3 So, What About Lua?

Roberto Ierusalimschy (Pontifical University – Rio de Janeiro, BR)

Lua is a programming language developed at the Catholic University in Rio de Janeiro that came to be the leading scripting language in video games. Lua is also used extensively in embedded devices, such as set-top boxes and TVs, and other applications like Adobe Lightroom and Wikipedia. This talk presents a quick overview of some unconventional aspects of the language.

7.4 Regular Expression Parsing

Bjorn Bugge Grathwohl (University of Copenhagen – DK)

Joint work of Henglein, Fritz and Terp-Rasmussen, Ulrik

Regular expressions (REs) are usually interpreted as languages. For many programming tasks, this is an inadequate interpretation, as it only provides the programmer with a means for testing language membership. Facilities for submatch extraction in tools such as sed and Perl-style REs have been developed to let programmers do data extraction and manipulation with REs. However, the submatch extraction approach is severely limited in its expressibility, as it only allows for a fixed number of submatches, independent of the input size.

Instead, we interpret REs as types. Testing language membership is replaced by a parsing problem: Given an RE E and string s, produce the value (parse tree) in the type T(E) whose flattening is s. With this interpretation, data extraction and manipulation can be performed by writing functional programs that operate on the data types represented by the REs.

We present two automata-based algorithms producing the greedy leftmost parse tree: The two-pass algorithm requires one pass over the input data and an extra pass over an auxiliary data structure; the streaming algorithm implements an optimally streaming parser, in the sense that as soon as the input read so far determines a prefix of all possible parse trees, this prefix is output. This is guaranteed given a PSPACE-complete analysis of the automaton, which can be performed independently of any input strings. However, we conjecture that for "realistic", non-pathological, REs, this analysis is not needed.

7.5 HTML5 Parser Specification and Automated Test Generation

Yasuhiko Minamide (University of Tsukuba, JP)

Joint work of Minamide, Yasuhiko; Mori, Shunsuke
Main reference Y. Minamide, S. Mori, "Reachability Analysis of the HTML5 Parser Specification and Its Application to Compatibility Testing," in Proc. the 18th Int'l Symp. on Formal Methods (FM'12), LNCS, Vol. 7436, pp. 293–307, 2012.
URL http://dx.doi.org/10.1007/978-3-642-32759-9_26

The HTML5 specification includes the detailed specification of the parsing algorithm for HTML5 documents, including error handling. We develop a reachability analyzer for the parsing specification of HTML5 and automatically generate HTML documents to test compatibilities of Web browsers. The set of HTML documents are extracted using our reachability analysis of the statements in the specification. In our preliminary experiments, we generated 353 HTML documents automatically from a subset of the specification and found several compatibility problems by supplying them to Web browsers.

7.6 AmbientTalk: a scripting language for mobile phones

Tom Van Cutsem (Alcatel-Lucent Bell Labs – Antwerp, BE)

We introduce the AmbientTalk programming language, which was designed to script collaborative distributed applications on mobile phones. We give an overview of the language's features and historical roots. We discuss how AmbientTalk is embedded on the JVM, with particular attention to maintaining concurrency invariants.

7.7 Glue Languages

Arjun Guha (University of Massachusetts – Amherst, US)

Joint work of Guha, Arjun; Gupta, Nimish

Puppet is a configuration management system used by thousands of organizations to manage thousands of machines. It is designed to automate tasks such as application configuration, service orchestration, VM provisioning, and more. The heart of Puppet is a declarative domain specific language that, to a first approximation, specifies a collection of resources (e. g., packages, user accounts, files, etc.) to install and the dependencies between them.

Although Puppet performs some static checking, there are many opportunities for errors to occur in Puppet configurations. These errors are very difficult to detect and debug. Even if a configuration is itself bug-free, when a machine is upgraded to a new configuration, it is easy for the machine state and its specified configuration in Puppet to be inconsistent.

8 Overview of Talks: Security

8.1 Information Flow Control in WebKit's JavaScript Bytecode

Christian Hammer (Universität des Saarlandes, DE)

Joint work of Bichhawat, Abhishek; Rajani, Vineet; Garg, Deepak; Hammer, Christian
Main reference A. Bichhawat, V. Rajani, D. Garg, C. Hammer, "Information Flow Control in WebKit's JavaScript Bytecode," in Proc. of the 3rd Int'l Conf. on Principles of Security and Trust (POST'14), LNCS, Vol. 8414, pp. 159–178, Springer, 2014.
URL http://dx.doi.org/10.1007/978-3-642-54792-8_9

Websites today routinely combine JavaScript from multiple sources, both trusted and untrusted. Hence, JavaScript security is of paramount importance. A specific interesting problem is information flow control (IFC) for JavaScript. In this paper, we develop, formalize and implement a dynamic IFC mechanism for the JavaScript engine of a production Web browser (specifically, Safari's WebKit engine). Our IFC mechanism works at the level of JavaScript bytecode and hence leverages years of industrial effort on optimizing both the source to bytecode compiler and the bytecode interpreter. We track both explicit and implicit flows and observe only moderate overhead. Working with bytecode results in new challenges including the extensive use of unstructured control flow in bytecode (which complicates lowering of program context taints), unstructured exceptions (which complicate the matter further) and the need to make IFC analysis permissive. We explain how we address these challenges, formally model the JavaScript bytecode semantics and our instrumentation, prove the standard property of termination-insensitive non-interference, and present experimental results on an optimized prototype.

8.2 Hybrid Information Flow monitoring against Web tracking

Thomas Jensen (INRIA Bretagne Atlantique – Rennes, FR)

Joint work of Bielova, Nataliia; Besson, Frederic; Jensen, Thomas

Motivated by the problem of stateless web tracking (fingerprinting), we propose a novel approach to hybrid information flow monitoring by tracking the knowledge about secret variables using logical formulae. This knowledge representation helps to compare and improve precision of hybrid information flow monitors.

We define a generic hybrid monitor parametrised by a static analysis and derive sufficient conditions on the static analysis for soundness and relative precision of hybrid monitors.

We instantiate the generic monitor with a combined static constant and dependency analysis. Several other hybrid monitors including those based on well-known hybrid techniques for information flow control are formalised as instances of our generic hybrid monitor. These monitors are organised into a hierarchy that establishes their relative precision. The whole framework is accompanied by a formalisation of the theory in the Coq proof assistant.

8.3 Intrusion Detection by Control Flow Analysis

Arjun Guha (University of Massachusetts – Amherst, US)

Joint work of Guha, Arjun; Krishnamurthi, Shriram; Jim, Trevor
Main reference A. Guha, S. Krishnamurthi, T. Jim, "Using Static Analysis for Ajax Intrusion Detection," in Proc. of the 18th Int'l Conf. on World Wide Web (WWW'09), pp. 561–570, ACM, 2009.
URL http://dx.doi.org/10.1145/1526709.1526785

We present a static control-flow analysis for JavaScript pro- grams running in a web browser. Our analysis tackles numerous challenges posed by modern web applications including asynchronous communication, frameworks, and dynamic code generation. We use our analysis to extract a model of expected client behavior as seen from the server, and build an intrusion-prevention proxy for the server: the proxy intercepts client requests and disables those that do not meet the expected behavior. We insert random asynchronous requests to foil mimicry attacks. Finally, we evaluate our technique against several real applications and show that it protects against an attack in a widely-used web application.

8.4 Multiple Facets for Dynamic Information Flow

Cormac Flanagan (University of California – Santa Cruz, US)

Joint work of Flanagan, Cormac; Austin, Thomas H.
Main reference T. H. Austin, C. Flanagan, "Multiple facets for dynamic information flow," in Proc. of the 39th Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages (POPL'12), pp. 165–178, ACM, 2012; pre-print available from author's webpage.
URL http://dx.doi.org/10.1145/2103656.2103677
URL http://users.soe.ucsc.edu/~cormac/papers/popl12b.pdf

JavaScript has become a central technology of the web, but it is also the source of many security problems, including cross-site scripting attacks and malicious advertising code. Central to these problems is the fact that code from untrusted sources runs with full privileges. We implement information flow controls in Firefox to help prevent violations of data confidentiality and integrity. Most previous information flow techniques have primarily relied on either static type systems, which are a poor fit for JavaScript, or on dynamic analyses that sometimes get stuck due to problematic implicit flows, even in situations where the target web application correctly satisfies the desired security policy. We introduce faceted values, a new mechanism for providing information flow security in a dynamic manner that overcomes these limitations. Taking inspiration from secure multi-execution, we use faceted values to simultaneously and efficiently simulate multiple executions for different security levels, thus providing non-interference with minimal overhead, and without the reliance on the stuck executions of prior dynamic approaches.

8.5 Shill: shell scripting with least authority

Christos Dimoulas (Harvard University, US)

Joint work of Moore, Scott; Dimoulas, Christos; King, Dan; Chong, Stephen

The Principle of Least Authority suggests that software should be executed with no more authority than it requires to accomplish its task. Current security tools make it difficult to apply this principle: they either require significant modifications to applications or do not facilitate reasoning about combining untrustworthy components.

We propose Shill, a secure shell scripting language. Shill scripts enable compositional reasoning about security through declarative security policies that limit the effects of script execution, including the effects of programs invoked by the script. These security policies are a form of documentation for consumers of Shill scripts, and are enforced by the Shill execution environment.

We have implemented a prototype of Shill for FreeBSD. Our evaluation indicates that Shill is a practical and useful system security tool, and can provide fine-grained security guarantees.

8.6 Hybrid Information Flow Analysis for JavaScript

Tamara Rezk (INRIA Sophia-Antipolis, FR)

We propose a novel type system for securing information flow in JavaScript that takes into account the defining features of the language, such as prototypical inheritance, extensible objects, and constructs that check the existence of object properties. The type system infers a set of assertions under which a program can be securely accepted and instruments it so as to dynamically check whether these assertions hold. By deferring rejection to run-time, the hybrid version can typecheck secure programs that purely static type systems cannot accept.

8.7 A Collection of Real World (JavaScript) Security Problems:

Achim D. Brucker (SAP Research – Karlsruhe, DE)

URL http://www.brucker.ch/bibliography/abstract/talk-brucker-js-challenges-2014.en.html

JavaScript is gaining more and more popularity as an implementation language for various applications types such as Web applications (client-side), mobile applications, or server-side applications.

We outline a few security challenges that need to be prevented in such applications and, thus, for which there is a demand for analysis methods that help to detect them during during development.

9 Lightning Talks

9.1 Reasoning about membranes using separation logic

Gareth Smith (Imperial College – UK)

URL http://www.dagstuhl.de/mat/Files/14/14271/14271.SmithGareth.Other.pdf

We propose an extension to separation logic which would make it possible to statically prove security properties of an implementation of a *membrane* program.

9.2 Complexity Analysis of Regular Expression Matching Based on Backtracking

Yasuhiko Minamide (University of Tsukuba, JP)

Joint work of Sugiyama Satoshi; Minamide, Yasuhiko
Main reference S. Sugiyama, Y. Minamide, "Checking Time Linearity of Regular Expression Matching Based on Backtracking," to appear in IPSJ Transactions on Programming.

Regular expression matching is implemented with backtracking in most programming languages. Its time complexity is exponential on the length of a string in worst case. This high complexity causes significant problems in practice. It causes DoS vulnerabilities in server-side applications. It may also affect the result of matching in some implementation with a limit on the number steps in matching, e. g. PCRE. We present a decision procedure to check whether for a given regular expression matching based on backtracking runs in linear time.

9.3 PHPEnkoder: a Wordpress Plugin

Michael Greenberg (Princeton University, US)

URL http://wordpress.org/plugins/php-enkoder/

PHPEnkoder encodes mailto: links and e-mail addresses with JavaScript to stifle webcrawlers. It works by automatically turning plaintext e-mails into (enkoded) links.

Interesting facts:

- Wordpress plugins are installed by being placed in a directory; the files are run at the top level.
- Wordpress plugins are automatically released by tagging in subversion.
- PHPEnkoder parses the page with regular expressions, since Wordpress 'hooks' don't give PHPEnkoder an AST to process, just text.
- Wordpress has an extremely stable API.

For more on this plugin, see http://www.weaselhat.com/phpenkoder/.

9.4 SAST for JavaScript: A Brief Overview of Commercial Tools

Achim D. Brucker (SAP Research – Karlsruhe, DE)

URL http://www.brucker.ch/bibliography/abstract/talk-brucker-sast-js-2014.en.html

Static application security testing (SAST) is a widely used technique that helps to find security vulnerabilities in program code at an early stage in the software development life-cycle. Since a few years, JavaScript is gaining more and more popularity as an implementation language for large applications. Consequently, there is a demand for SAST tools that support JavaScript.

We report briefly on our method for evaluating SAST tools for JavaScript as well as summarize the results of our analysis.

References

1 Achim D. Brucker and Uwe Sodan. Deploying static application security testing on a large scale. In Stefan Katzenbeisser, Volkmar Lotz, and Edgar Weippl, editors, *GI Sicherheit 2014*, volume 228 of *Lecture Notes in Informatics*, pages 91–101. GI, March 2014.

10 Breakout Sessions

In addition to the contributed talks, the seminar had four breakout sessions focussing on cross-cutting issues deemed important by the participants.

10.1 Contracts and Blame

Cormac Flanagan

We discussed some of the counter-intuitive ways in which contracts can fail in systems with multiple modules, and the ways in which blame may be assigned in a manner that may not point at the component that is truly at fault.

10.2 On the Role of Soundness

Matthew Might, Jeffrey Foster

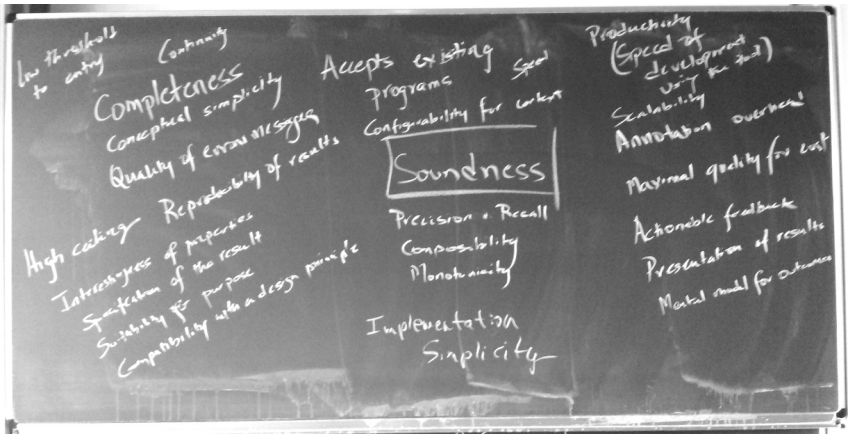
We debated the merits and importance of soundness of tools and analyses for scripting languages. On the one hand, while soundness is essential for relying upon the results of the analysis, on the other, some constructs may be pathologically hard to analyze soundly and even unsound tools may provide extremely invaluable feedback to the developer.

10.3 Metrics for Programming Tools

Krishnamurthi, Shriram; Politz, Joe Gibbs

URL https://drive.google.com/file/d/0B32bNEogmncORS1sN0YtaXZ3V1k/edit?usp=sharing

We gathered metrics for measuring the utility of programming language tools (focused on scripting language applications), prompted by considering alternatives and complements to soundness. See sketch on the blackboard below.



10.4 JavaScript Analysis and Intermediate Representation

Thomas Jensen (INRIA Bretagne Atlantique – Rennes, FR)

Joint work of Jensen, Thomas; Sridharan, Manu

Two issues were discussed:

- how to share models of libraries,
- can we come up with a common intermediate representation for JS analyzers.

The overall goal is to support a re-usable, shared effort. Modeling libraries is not very publishable, hence the need for a collective effort. Another issue is that different kind of models are needed, depending on the analysis. Nevertheless, it was deemed worth to have a common starting point. Models could be written in JS or in an IR or in a formalism that allows integrating elements of abstract domains. One point of view was that it would be valuable to have models satisfying that everything is translatable to the IR, so that different library models can co-exist.

Concerning the IR, several points were discussed:

- Should it accommodate `pre/post` annotations to model libraries?
- Should it be executable (could enable re-injecting into JS to do dynamic analysis)? There is a certain amount of common structure in existing IR so why not just pick one of those.

Some shortcomings were discussed: WALA: not serializable, which is necessary, S5 : should be OK, can be ANF-ed and CPS-ed, MSR IR: has existing formats but prepared to do a clean slate Two different kind of formats were identified: a CFG or something close

to the AST. Perhaps there is a need for a series of IR that end in the common format but maximum two seems reasonable to standardize. The discussion ended with a presentation of a proposal for a common IR. The current version can be found at the URL above.

Participants

- Achim D. Brucker
SAP Research – Karlsruhe, DE
- Niels Bjoern Bugge Grathwohl
University of Copenhagen, DK
- Ravi Chugh
University of California – San Diego, US
- Arlen Cox
Univ. of Colorado – Boulder, US
- Christos Dimoulas
Harvard University, US
- Julian Dolby
IBM TJ Watson Research Center – Hawthorne, US
- Matthias Felleisen
Northeastern University – Boston, US
- Daniele Filaretti
Imperial College London, GB
- Cormac Flanagan
University of California – Santa Cruz, US
- Jeffrey Foster
University of Maryland – College Park, US
- Ronald Garcia
University of British Columbia – Vancouver, CA
- Philippa Gardner
Imperial College London, GB
- Michael Greenberg
Princeton University, US
- Arjun Guha
University of Massachusetts – Amherst, US
- Shu-Yu Guo
MOZILLA – Mountain View, US
- Christian Hammer
Universität des Saarlandes, DE
- Fritz Henglein
University of Copenhagen, DK
- Roberto Ierusalimschy
PUC – Rio de Janeiro, BR
- Thomas Jensen
INRIA Bretagne Atlantique – Rennes, FR
- Ranjit Jhala
University of California – San Diego, US
- Matthias Keil
Universität Freiburg, DE
- Shriram Krishnamurthi
Brown University, US
- Benjamin Lerner
Brown University, US
- Benjamin Livshits
Microsoft Res. – Redmond, US
- Sergio Maffeis
Imperial College London, GB
- Matt Might
University of Utah, US
- Yasuhiko Minamide
University of Tsukuba, JP
- Anders Møller
Aarhus University, DK
- Joe Gibbs Politz
Brown University, US
- Ulrik Terp Rasmussen
University of Copenhagen, DK
- Tamara Rezk
INRIA Sophia Antipolis – Méditerranée, FR
- Tiark Rompf
EPFL – Lausanne, CH
- Sukyoung Ryu
KAIST – Daejeon, KR
- Alan Schmitt
INRIA Bretagne Atlantique – Rennes, FR
- Jeremy G. Siek
Univ. of Colorado – Boulder, US
- Gareth Smith
Imperial College London, GB
- Manu Sridharan
Samsung Research, US
- Éric Tanter
University of Chile, CL
- Peter Thiemann
Universität Freiburg, DE
- Sam Tobin-Hochstadt
Indiana University – Bloomington, US
- Tom Van Cutsem
Alcatel-Lucent Bell Labs – Antwerp, BE
- David Van Horn
University of Maryland – College Park, US
- Panagiotis Vekris
University of California – San Diego, US
- Ben Wiedermann
Harvey Mudd College – Claremont, US
- Kwangkeun Yi
Seoul National University, KR

Report from Dagstuhl Perspectives Workshop 14272

Exploring Interdisciplinary Grand Challenges in ICT Design to Support Proactive Health and Wellbeing

Edited by

m. c. schraefel[1] and Elizabeth F. Churchill[2]

1 **University of Southampton, UK,** mc@ecs.soton.ac.uk
2 **eBay Research Labs, San Jose, CA, US,** churchill@acm.org

Abstract

There have been significant successes in ICT in eHealth. Examples include deploying mobile devices to improve drug adherence, designing Internet services to extend human expert contact, and developing devices and services that encourage engagement in proactive healthcare activities. From an infrastructure perspective, better supply-chain management has reduced healthcare and patient support costs.

However, we believe that even greater benefits for improved Quality of Life (QoL) can be realized by broadening the eHealth agenda. We advocate moving upstream from medical intervention and healthcare for those already diagnosed as "il" to the design of sociotechnical technologies and systems aimed at fostering Proactive Health and Wellbeing. While not focused on medical health issues specifically, proactive strategies for wellbeing are key to long term health and thus to the reduction of healthcare needs and costs. Through support for lifestyle adjustments to focus more firmly on proactive strategies, we will no doubt achieve reductions in the number of people who become ill in the first place. This will, in turn, reduce the costs of healthcare support at individual, group and societal levels. Good examples are preventable lifestyle conditions such as obesity and heart disease.

Two major challenges are clear, each of which has a number of sub-challenges.

Our first challenge is to map key issues that are tractable in the short, medium and long term. To this end, an interdisciplinary group of researchers from academia and industry, with expertise in sports science, neurology, cardiology, computer science, psychology and sociology met to create a road map for research challenges around developing interactive technologies to support this proactive health and wellbeing agenda. This gathering of research leaders was the Perspectives Workshop on Interdisciplinary Grand Challenges in ICT Design to Support Proactive Health and Wellbeing. Here, we posed the question: What are the key Human Computer Interaction and Computer Science research challenges that need to be addressed for us to support more effective proactive health and wellbeing practices in the long term? We derived five key challenges which we propose to be the foundations for a new research area, "Wellth Sciences": 1) Developing Effective Methodologies, Measures and Metrics for Understanding Proactive Health and Wellbeing; 2) Understanding Motivation and Sensemaking with regard to experiential aspects of a proactive engagement with wellbeing and health; 3) Rethinking Design Practices; 4) Creating New Frameworks and Models; and 5) Rethinking the Phenomenology and Epistemology of "Health". These challenge areas are detailed in the following report, along with landmarks for success at 1, 5 and 10 year periods.

The second major challenge is to foster a dedicated, multi-disciplinary research community focused on these issues. The Perspectives Workshop gave us the first step forward toward addressing this challenge. We offer a proposal for ongoing connection and collaboration between those assembled for the workshop, and for inviting others to address the research areas identified.

Perspectives Workshop June 29 to July 2, 2014 – http://www.dagstuhl.de/14272
1998 ACM Subject Classification H.1.2 User/Machine systems, H.5 Information Interfaces and Presentation

Exploring Interdisciplinary Grand Challenges in ICT Design to Support Proactive Health and Wellbeing, *Dagstuhl Reports*, Vol. 4, Issue 6, pp. 108–123
Editors: m. c. schraefel and Elizabeth F. Churchill

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

Keywords and phrases Proactive Health, Proactive Wellbeing, eHealth, Wellth, Wellth Creation, Quality of Life, societal impact, interactive design, health, wellness, Computer Science, Human Computer Interaction
Digital Object Identifier 10.4230/DagRep.4.6.108
Edited in cooperation with Adrian Friday

1 Executive Summary

m. c. schraefel
Elizabeth F. Churchill

To date poor health costs billions annually, negatively impacting our nations' GDPs. Costs include provision of healthcare for acute and chronic physical and mental conditions and reductions in productivity resulting from absences from work due to sickness.

Much hope has been placed in the deployment of networked information and communications technologies (ICTs) to improve the health of citizens, engage them in proactive healthcare strategies, and thus reduce the likelihood of illness in the first place. Part of the promise is that ICTs in the form of personal, commercial and infrastructural/ governmental platforms may be deployed ubiquitously, pervasively and more cost-effectively than one-to-one human care.

This perspective draws primarily from advances in mHealth and eHealth in the medical community. The focus in these domains, however, is to see health as a medical condition, focussing on tracking and management of patient records, support for doctor-patient interaction, and technologies for regimen adherence and therapy management. In our view, an excellent complement to this perspective is a focus on Proactive Health and Wellbeing, where the concept of health is broadened from being the absence or management of a medical condition or conditions to include a personal engagement with and understanding of wellbeing. ICT has, so far, delivered less success in this arena[4].

The Perspectives Workshop on Exploring Interdisciplinary Grand Challenges in ICT Design to Support Proactive Health and Wellbeing was convened to engage with these issues. We invited scholars to focus on Proactive Health and to elicit what key challenges we need to address in ICT that, if we were to put in concerted and coordinated effort as a community, would have demonstrable effect. We invited reflection on the promise of ICT in contributing to global health, GDP and wellbeing. Our participants have come from various areas in computer science, principally Human Computer Interaction, Data Science and Information Studies, both from industry and academia. We also had participation from psychology, sociology, sports science, medicine and neural science. While most participants were established research leads, we also reached out to up-and-coming, early career researchers in Computer Science and Human Computer Interaction who have a developing track record on health and wellbeing related topics. These individuals will be the future leads in this emerging field.

Over the three days of our workshop we developed 5 key challenge areas, focusing on the significance of each challenge, success at year 1, 3 and 10, as well as resources required to facilitate success. These areas correspond to data sciences, motivational modeling, design thinking, framework building, and a higher order rethinking of the space of "health":

1. Developing Effective Methodologies, Measures and Metrics for Understanding Proactive Health and Wellbeing. Small and "Big" data need to be captured, cleaned and curated to more effectively reflect hard-to-measure experiential aspects of wellbeing. Qualitative data are needed to better understand what is being captured quantitatively, and to enable a deeper understanding of the diversity of experience and to more deeply investigate what is represented in the data within the "long tail".
2. Understanding Motivation and Sensemaking. New models of motivation and sense making are needed in order to more deeply understand people's aspirations and the contingencies of their everyday lives that enable or prevent personal proactive health and wellbeing practices. A move from imposing normative models of "change" to understanding how sustained motivation and self- and other-persuasion can result in new and innovative technology-enabled programs is needed. .
3. Rethinking Design Practices. We need reflective design practices that focus on the phenomenological aspects of a design to complement designs that focus on intervention and instrumental goal achievement. This arena relates to the need for better motivational models, but addresses the ways in which our design practices mould what we create. How can we more effectively move basic science into applied science and more effective engineering?
4. Creating New Frameworks and Models. We need to develop frameworks and models that take into account unconscious as well as conscious drivers of human behavior, that better connect 'body', 'mind' and 'feeling' experiences, that address emotions as well as cognitive processing, and that acknowledge rhythms of participation and non-participation that are health-positive as well as those that are health negative. This requires a deeper engagement with psychosocial, brain and biological sciences to develop and bring into perspective more holistic frameworks and models.
5. Rethinking the Phenomenology and Epistemology of "Health". Rolling the previous areas up, one of the broader challenges directly addresses how to drive multi disciplinary thinking in regard to proactive wellbeing. A new field of enquiry at the intersection of Human Computer Interaction (HCI) and Computer Science, we need to think about how to motivate and increase engagement from researchers, from designers and engineers, from policy makers, from governmental agencies and from business leaders.

The key outcome of the workshop is an affirmation that a focus on Proactive Health and Wellbeing is both timely and socially necessary, and represents a viable area of research and development. A suite of near-term future activities have been planned and "owned" by participants to drive forward in the coming 6 months. Activities include a follow up Dagstuhl seminar, and workshops, panels, summer schools, invited publications, special issues, and the establishment of an area conference. We have also agreed to explore new ways to engage around experimental design, feedback and collaborative work. We invite potential collaborators to contact us for further discussion and to learn more about our ongoing efforts in this emerging arena of Wellth Sciences.

2 Table of Contents

3 Key Challenges Identified

All workshop participants

Five key challenge clusters were identified, each having sub-challenges that are likely to lead to targeted research proposals and the need for strategic cross-disciplinary collaboration and applications for resourcing. Our intention is to use these clustered challenge areas to begin to identify key collaborative opportunities, funding sources and opportunities for platforms for further exploration (e. g. conference workshops, sponsored seminars and working groups, etc).

Document Organization

For each challenge, we present an overview of the challenge, and offer a set of associated research challenges, and in many cases, what success at 1, 3 and 10 year points, and our requirements to enable success.

3.1 Challenge 1: Methodologies, Measures and Metrics

3.1.1 Overview

Broadly speaking there are two data related movements within the proactive health and wellbeing arena – quantitative and qualitative data from an input and output mechanism. *Quantitative data* is related to easily computable data (although not necessarily accurate at this time) in individual or aggregate statistical data form garnered from medium to large populations with the intention of communicating trends to individuals or with service providers, monitoring agencies, marketing initiatives, epidemiological and controlled comparative studies and so on. Quantitative data can be input (e. g. I swam for 20 minutes; my heart rate is 152), analyzed, and output (e. g. you walked 9,573 steps today). *Qualitative data* from an input perspective is related to the lived experiences of individuals and can cover everything from one's culture, stories, and rich multimedia/sensor experiences (e. g. a video of one's experience in a specific context). Qualitative output is broadly defined as reflecting on one's lived experiences (e. g. listening to one's favorite experiences) or viewing abstracted quantitative data (e. g. a light color that is colored based on how long one spent outside).

For Quantitative Data, statistical information is derived and manipulated to understand the relationship between behaviors, demographics or other kinds of antecedents and health outcomes. Indeed, health science traditionally relies on statistical information from large populations to understand the relationship between behaviors, demographics or other kinds of antecedents and health outcomes. This information is then used to offer prescriptive or corrective advice to achieve some goal. However, these statistical tests generally only offer reliable information when there are fairly large groupings of people that are similar in a given distribution. That is, we can offer good advice to people when they fall in the "body" of the distribution – the middle 60–80% of the population for a given variable (e. g. age, race, gender, diet, exercise regimen, body type, BMI, resting heart rate, etc.). However, this fails for those in the "tails" of the distribution, those who fall on either side of the middle clump

of people. It also fails when the distribution is more evenly distributed, or flat. This is even more complicated because most people fall in the middle of a distribution on some factors, but out in the tails on others. Thus, on person might be very similar to most other people, but respond very differently from those others in the remedies that are prescribed. Someone else might be like most people on both of these factors, but for some reason find it very easy to maintain a reasonable BMI with very little exercise. Health science has traditionally been very useful for people when they fall in the body of the distribution, but when they are in the tails, it can recommend advice that just doesn't work (or is even harmful).

This is where recent advances in data science can be very helpful. By exploring very large data sets of many people, signals in the tails become understandable. Machine learning and information visualization techniques allow us to make sense of large number of features (different factors) and how they relate to each other to result in meaningful classifications & groupings at an individual level. The classic example of this in other domains is personalization in shopping or entertainment (Amazon, Netflix, Google, etc.). There is an opportunity to innovate in similar techniques to make huge strides in how we understand wellness information on a personal level based on the aggregate data of millions of other people each providing vast amounts of individual data from new sensors and other sources. Naturally, there are serious obstacles to work through. For example, the data in this case is very sensitive and so guarding privacy is very important.

The second important format for data is personal data, as in the kinds of data available through personal fitness and body monitoring device. Examples include the Fitbit and Nike FuelBand. Much has been covered in this arena by members of the Quantified Self movement, and recent years have seen an enormous amount of investment in this space from venture capitalists and those who are keen to link these personal data 'pools' to the aforementioned large scale predictive modelling efforts.

Quantitative data are excellent for supporting decision-making around activity engagement and program change. They enable us to understand whether our health is improving by offering a baseline and or a standard against which we can measure ourselves. However, there are challenges in helping people understand what their personal data mean – as noted above individuals vary considerably and there is no one size fits all measure. It is hard to help people understand what is reasonable change over time. Health regimens are notoriously poor for maintenance when the focus is on quantitative measures of change and "improvement". We need to develop better methods for aiding individuals to make more effective use of their data. Researchers and industry have abstracted quantitative into qualitative visualizations (e. g. a flower growing on a fitbit) to assist lay people understand the impact of their everyday activities on their health, however we need a better understanding of what these visualizations should look like to assist people make better real time and post-reflective decisions. In addition, we need to develop flexible systems where people can decide how they want this feedback as they go throughout their lives – maybe during the week when one is working, they need a subtle, vibrotactile reminder that they are sitting too long, however during the weekend when activities are more varied and include other people, maybe something more ambient and attention getting is needed.

In addition to abstraction, qualitative data can take the form of narrations and lived experiences. The qualitative data that help us understand and interpret the quantitative datasets are essential for developing understanding. The most successful fitness and wellbeing applications and services are those that are social, where people are able to account and narrate their activities. These are not successful, however, when people are made to be accountable to an imposed regimen of improvement. They are successful when they are focus

on experience rather than outcome. In addition, designers can gain a rich understanding of one's life through these qualitative narratives such as recent work that has been done investigating youtube videos.

Quantitative data are about monitoring, and are easy to generate. Qualitative experiences are somewhat easier to collect, however they are harder to design for and are difficult to measure. We will return to this point in outlining another challenge, below.

Within the space of data capture, we also have a particular opportunity to consider the long tail in new ways.

3.1.2 Key Research Challenges

- The more automated quantitative and qualitative data collection methods become, the less we burden users, however they may also be less likely to think about the data and reflect on it. How do we design low burden, but high reflection health and wellness tools that empower people to take action in their everyday lives?
- Today's health and wellness technology is largely still created for majority groups (typically middle to high socioeconomic, knowledgeable and/or enthusiastic about technology with access to relevant information resources, and often male). How can we integrate more diversity into the stakeholder design space (e. g. children, people with low literacy and/or numeracy and fewer economic and other resources)?
- Social interactions are a key component in motivating people to maintain health lifestyles – once we diversify the stakeholder design space, how should the research community address the power relationships within these social interactions (e. g. child – parent; patient-healthcare provider)?
- Quantitative data is relatively easy to collect (e. g. accelerometers), however difficult to visualize. How should we effectively visualize quantitative data for a diverse population user group?
- Qualitative data is easy to collect, however difficult to process and reflect on (in large quantities). For collection, how can we make it easy to collect while maintaining others privacy expectations? For reflection, how can we make it easy to reflect on large sets of qualitative and derive appropriate results (e. g. no confirmation biases)?

3.2 Challenge 2: Understanding Motivation and Sensemaking

The individual is a critical agent in proactive health. While ICTs are enablers for successful participation, the decision to participate rests with the individual. Thus, we need to understand how to motivate the individual to participate in his or her health. Since *proactive health* requires a life-long commitment, the time dimension is critical in addressing motivation, as we know that people's commitment will wax and wane over time. Thus, we need improved models of how motivation operates over *it's lifecycle*, that is, how to initially involve people in proactive health, how to maintain their commitment and how to enable them to return to proactive health activities should their commitment wane.

The large body of literature that has investigated motivation is a testament to the complexity of the topic. Increasingly researchers are adopting ecological theories (e. g. the work of Uri Bronfenbrenner on Ecological Systems Theory) to organize the interdependency of individual determinants of behavior (psychodynamic factors) with those derived from social (e. g. emotional and pragmatic support within the family), organizational levels and

cultural levels (e. g. socioeconomic factors such as poverty, social opportunity) to understand how to better support an individual to make positive behavioral decisions.

We propose to augment existing theoretical approaches to motivation with a heavily data-driven approach to understand the operation of motivation across the lifecycle with regards to the individual's involvement in proactive health activities. The goal of this work stream is to devise a method for using behavioral data collected in the course of normal activities to refine initial estimates of the "motivational equation" for an individual that were defined on the basis of theoretical constructs. Some basic research questions include the following. Is it possible to consider normal activities in the course of an individual's life to be small "experiments" that provide data to be used to refine the set of "proactive health experiences" initially offered to an individual based on a categorization guided by theories of motivation? What kind of infrastructure might be needed to collect data to support such data-driven refinements of proactive health experiences? What data is useful for making initial theory-based classifications of people? What results on in-the-wild experiments would be informative for updating the scripting of future proactive health experiences?

3.3 Challenge 3: Rethinking Design Practices

One challenge facing designers of health and wellness systems is which goals they should support or encourage and how heavy-handed they should be in their promotion of them. Fit4Life [3], an elegant paper by researchers at Cornell, describes a future in which people succeed at creating an engaging system that monitors and provides feedback/instructions on what to do to achieve an optimal state of health and fitness, as determined by the application's designers or health professionals. This future, however, is dystopian. Wearers of the system are shamed into compliance, and it becomes clear that the system allows little to no room for other forms or definitions of wellness or well-being.

Researchers and consumer device manufacturers have been rapidly developing the capabilities described in the Fit4Life paper. These include persistent, ubiquitous sensing capabilities for a variety of biomarkers and activity indicators, techniques that can identify patterns in this data, and a variety of real-time feedback techniques to promote healthier behaviors. If taken to their logical extreme, current trends in personal informatics may approach or even achieve Fit4Life.

We argue, however, we should strive not to achieve this future. Over-optimizing for certain health outcomes or focusing overly on specific numbers may come at the expensive of individual autonomy or other forms of wellbeing.

3.3.1 Key Research Challenge

Our community should identify, embrace, and further develop design strategies that help balance these tradeoffs for a specific situation. While we generally argue for design of systems that respect people's wishes and autonomy, we can quickly identify situations when more persuasive systems, or even coercive systems, may be appropriate (consider, for example, someone who is anorexic, or parents encouraging their child to eat vegetables). We also agree that people should not always be so focused on their health and wellness that they are unable to enjoy a delicious meal or simply just be in a moment without worrying about the consequences of each small decision, but we also agree that some amount of mindfulness and reflection is appropriate or necessary. There are also many questions about whether and how

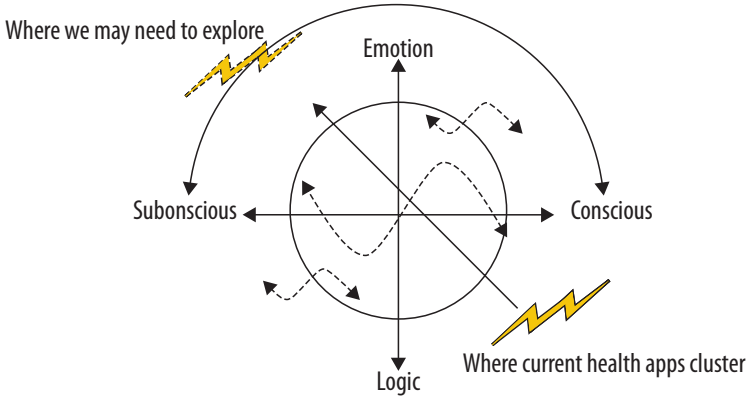


Figure 1 Design dimensions for proactive wellbeing.

to engage people who are not motivated to change their health and wellness behavior, but for whom health providers or others in society generally believe changes would be beneficial.

Determining guidelines for ethical design and deployment of proactive health and wellness technologies, or even how one even goes about identifying such guidelines, is a grand challenge for the next decade. It is imperative that our communities begin work on this before we create systems that create a future that most people would not want, and that we revisit guidelines as new systems reach the market and technological advances make new interactions possible.

3.3.2 Success markers in 1, 3, and 5 years

As an **immediate goal**, our research community should identify one or more measures of quality of life that go beyond specific health outcomes and be considering them for use in future studies. Many quality of life scales exist; it is outside of our current expertise to identify those that are most suitable or changes that would make them more appropriate.

Similarly, our community should adopt and promote the development of design strategies that help designers and policy makers identify a full range of individuals' motivations, goals, and priorities, and design to be inclusive and respectful of those. In particular, we think that health and wellness intervention designers would benefit from more exposure to and training in Value Sensitive Design [1].

3.4 Challenge 4: Creating New Frameworks and Models

Of central import is the emerging design space of proactive wellbeing; a first attempt to outline this design space is shown in Figure 1.

Our thesis is that behaviour change interventions typically target the conscious self, appealing to us as rational beings: it's good for us, therefore we will of course integrate wellbeing into our lives. Only we often don't. Wellbeing is often not prioritised, i. e. is subsumed and subjugated by pressures of work and home, or constraints of the infrastructure of our everyday lives. Consequently, those needing wellbeing the most are least likely to action towards it. A marked social inequality in wellbeing at work also exists, i. e. those with the lowest paid and most strenuous jobs are least likely to call for actions towards wellbeing.

Our horizontal axis takes the focus of our designs from the conscious and rational, which is still an important target, towards the unconscious. This is partly motivated by technologies that speak to the unconscious brain, e. g. training aides while sleeping; but is also in our deeply rooted autonomous actions: our primitive brain is trying to gain energy quickly and optimise and store energy in case we need it (fight or flight). In order to understand these domains, we must take our conscious and subconscious perception into account. Some foods have an unconscious appeal (the burger is always more tempting than the carrot) – how can we start to design to support making wellbeing less conscious and rational? How can the technology be used to nudge us towards healthier behavior?

The vertical axis appeals to our senses and emotions. Humans inherently make choices that are instantaneously rewarding: dopamine release is pleasant, rather than rewarding in the long-term. We are envisioning the future where we incorporate all the human senses; how we perceive our world is dependent on how we perceive our sensory stimulus and its interplay with our emotions and memories. Can we design technologies that filter or augment our perception of the world (AR, BCI) to make wellbeing more appealing at an emotional level. We might for example, offer stimulation and rewards, e. g. that "back of neck feeling", to reward health promoting behaviours. The nascent argument being that we are emotional creatures, and that our hearts will rule our rational heads. Along this axis we would include social technologies that promote and share positive reinforcement to make proactive wellbeing more normatively acceptable, and to gain support from others.

By understanding, appealing or even manipulating our subconscious reasoning and perception, we might find new ways to encourage wellbeing in our lives. However, broadening out from the individual, we must also recognise that we are not typically unconstrained or free to act in our rational or even irrational choices. We act in a context constrained by rules, relationships, policy and infrastructures of both of our home and work lives, and of the wider social norms and expectations of our peer groups and of society.

This opens up in important 'intentional design' concept 'proactive wellbeing by design': in which policies that promote wellbeing become embedded in the socio-technical systems, tools and technologies that surround us and support our lives. Applying this concept to conventional workplace systems, we use the example of a company meeting strategy in which the calendaring tool deliberately suggests 'walking meetings' or books rooms that are *deliberately less convenient* to encourage exercise. All this done in a fashion that seems appealing and meaningful to the employees rather than annoying. Similarly, a route planner might leave a transit system early to give you with a short walk at the end of your route, while still ensuring you arrive on time. Here, we might think of 'deliberate inconvenience' that promote new experiences related to wellbeing in order to break down preconceptions and gain new competencies [2].

We also note the trends toward 'quantified self' technologies for reflecting on and motivating exercise (e. g. step counters, smart watches, and so on). While such technologies clearly focus on benefit to the individual, we believe a significant opportunity space exists for exploiting this information collectively 'en masse': beyond the quantified self (as individual), we might think of the quantified workforce (as collective), i. e. how can we leverage aggregate measures of the self in and beyond the self to help make strong empirical cases on the positive impact of wellbeing technologies and strategies to powerful actors controlling such infrastructures (e. g. employers, town planners, politicians). These data and new tools to analyse and visualise it, could help start powerful changes to the environments, facilities, and policies that surround us, and thus enable practices that reshape workplace and social norms towards proactive wellbeing.

3.5 Challenge 4, Part 2: Scenarios for Exploration

For each individual, long-term health and wellbeing depends greatly on maintaining healthy behaviors and reducing habitual unhealthy ones. It seems appropriate to take a step back when designing health behavior change technology and aim to really understand the daily contexts before we commit to programs of proactive health intervention. For example, a 'go-to-the-gym' app on your smartphone does not make sense if you live in an area where there is no gym around. But what might make sense in this circumstance is if you knew that your elderly neighbor would be happy if you would walk her dog twice a week. How would one be able to integrate highly contextual personal and interpersonal understandings in relation to the more quotidian aspects of our life. How do we identify and seamlessly mesh the many specifics of context in order that one might develop systems that take advantage of such contextual understandings? We believe it is essential that we get people involved in proactive health design. We would like to believe that people could democratically engage in the developments of systems that might have an impact on their own life. We would also like people to explore the notion of tools for self-design tools in respect to proactive health and personal behavioural adaptation. We would also like to look at the social dynamics of such situations - who gets empowered, who can get impact, who says what intervention is right? In aiming for adaptation and adoption of proactive health behavior change there needs to be a degree of understanding that comes directly from the user in order that any change can be maintained and sustained. Therefore, we would argue that there needs to be flexibility in proactive systems and that such flexibility needs to be transparent, intelligible, and a subject of ongoing conversation.

3.5.1 Key Research Challenges

- Our first specific challenge in designing health behavior change technology is to understand the daily contexts of people's lives before we commit to building specific interventions. For example, a go-to-gym-app does not make sense if you live in an area where there is no gym around but your elderly neighbor would be happy if you walk her dog twice a week.
- With specific regard to the role of technology infrastructure that would be required to support behavior change applications we seek to create and/or engage with venues and platforms for creation and accretion of knowledge and motivational resources around nudging people to be healthier in their daily lives.
  - We can see other examples of such platforms in open source software projects and repositories, Massive open online course, and Wikipedia to name a few.
  - The domain of application is specific behavior change to mitigate long-term healthcare costs due to correctable behaviors, e. g. reduce preventable health inhibitor such as workplace conditions that contribute to lost productivity in the workplace (e. g. joint and back pain and other strength and posture related conditions) and increased healthcare cost in the long term because of emergent diseases (e. g. diabetes, deterioration of the spine)
  - We envision creating or participating in one or more marketplaces or other economic mechanisms for those involved in content production and management and in service provision related specific proactive healthcare interventions and contexts. i. e. Host specific behavior change applications that arise from and/or integrate to the platform
  - We further seek to empower interested stakeholders to be involved in self-design tools for health behavior change. This goal arises from working in the margins of official Healthcare systems

3.6 Challenge 5: Rethinking the Phenomenology and Epistemology of "Health", A Meta Challenge for Computer Science and Wellth Science – framing a new discipline

We are inspired by Ben Shneiderman's Science 2.0 that shows great science can come from exploring real and practical problems [5]. Our health, wellbeing, quality of life are all in the space of such real problems. This document has presented a suite of key challenges that require Computer Science expertise to help solve.

Within Computer Science

Many computer scientists will immediately see opportunities from these research questions to help advance our knowledge into delivering support from infrastructure to interaction to have social benefit. This immediacy is truly exciting: it gives us real footholds to make progress, and offers opportunities for applications from domains that may not have considered they have anything to offer these domains.

Some example computer science domains/challenges driven by our interaction-oriented Wellth questions include:

- *Systems and Semantics*: collecting diverse personal health data sources into a unified database, while preserving individual privacy and ensuring high data quality. Developing standards for data interchange, and metadata annotation rules to ensure compatibility.
- *User interface*: Enabling users with diverse skills to understand and manage their own personal health data. .
- *NLP/text analytics*: extracting health/wellth signals from social media streams (improved Google Flu Trends) search terms, or blog posts.
- *Big Data analytics*: Researchers, public health professionals, and others need visual and statistical tools to sift through the high volume of data, clean out erroneous values, build models, present correlations, develop causal hypotheses, refine theories, and propose practical guidelines.
- *Personal Informatics and Machine Learning*: integration of personal data sets, from calendars, to social interactions, to content creation, to physical sensor data. There are opportunities to look for patterns to help surface connections and correlations to help inform practice.
- *Information Visualization*: Visualization to provide comprehensibility of these patterns not as histograms but as answers to questions: where am I relative to temporal changes, comparisons with similar people, and geographic patterns.
- *Computed Security and Computed Policy*: We need to create dynamic policies for data that may not be stored but generated by queries on personal information.

There are terrific opportunities for the challenges in this report to drive fundamental computer science research, where we need innovative infrastructure to support trustable, safe data interactions on our behalf. Driven by the human requirements for better normal, we see such profound challenges for fundamental computer science to help optimize performance for what we can do to be useful and usable in people's hands and contexts.

With Computer Science – Gaining New Expertise

We have deep within-discipline opportunities to support these challenges as Computer Scientists and HCI researchers. To deliver on the challenges identified throughout this report, particularly in terms of understanding exactly what we may wish to begin to design, however,

we see an opportunity to develop new expertise beyond Computer Science and HCI. For instance, as largely computer scientists, few of us have the skills and knowledge necessary to design tools to support, for instance, coaching in general or coaching nutrition practice in particular. A usual approach may be to collaborate with domain experts, and design tools relying on their insights. Wellth Science and Engineering as it may become known, however, also offers opportunities to consider what expertise do we want to hold personally to be Wellth Scientists.

The knowledge challenges this domain presents about for instance how does the state of the body influence the state of the mind, we hope will become part of Wellth Science general education. Our bias in this workshop is that many of our societal challenges for which we are endeavoring to develop ICT health and wellbeing solutions exist due to a profound lack of knowledge about how our bodies (and brains as part of our bodies) perform. As a result of this knowledge gap we have a consequent lack of skills and experience on how to operationalize and sustain personal and social good practice.

We strongly recommend that computer scientists interested in designing for Wellth Creation become themselves literate about the body-brain connection. We suggest that this expertise development to be a Wellth Scientist and Engineer is a meta challenge within this domain. We need to consider the curriculum that would be optimal for such a researcher and design practitioner.

Fortunately, such domain curriculum consideration is a familiar experience for computer science, itself a domain hybrid from math. omputational linguistics draws on the formalisms of computer science machine learning and linguistics, largely based in the humanities. New international programs in support of "biologically inspired computing" has blended domain expertise in molecular biology, computation and devices. Human Computer Interaction itself brings psychology, sociology and human factors to blend into effective and efficient interactive designs to help enhance people's lives, from the workplace to the home. We suggest that given the scope of the issues around our health, wellbeing and quality of life, we need to consider the benefit of developing similar interdisciplinary Computer Science programs around Wellth Science.

In the CS/HCI space in formal Health Care, we already see examples of researchers spending time from their existing degree program or post grad research efforts, to gain considerable knowledge and expertise about medical practice to have meaningful conversations with the professionals in this space. These undertakings, however, have been largely ad hoc and individual, where that individual has often had to negotiate through degree timelines or research project support to be able to have the time to gain this level of expertise. We suggest that by being more deliberate about the kinds of knowledge students, researchers, practitioners need as part of our training to be effective in a timely way in Wellth Science, that we consider the benefit of incorporating these studies as part of a Wellth Science Design and Engineering program from the outset.

3.7 Challenge 5, Part 2: Framing a Domain Epistemology

The next one to three years will be key to shaping the concept of the Wellth Sciences. A key question throughout the workshop compared traditional HCI ways of knowing and asserting knowledge via short, small participant sample evaluations with the medical model of large randomized control trials. Into this mix we proposed the new and exciting opportunities to run large scale n=1 experiments remotely. This is an accepted model of evaluation neither

in HCI nor other human oriented sciences, and yet it seems particularly apt for exploring effect when one wants to consider practice interventions.

Similarly, new models of collaboration and of impact are needed. For example, we propose the sharing of experimental protocols before experiments are run (similar to medical sciences). This would enable scientists to gain comments on a proposed protocol and also encourage new ways of running experiments such as co-running distributed interventions and creation of new data sets that can themselves be shared.

Current models of practice in HCI in particular were critiqued. Questions were raised whether our nascent community's focus might not be better spent on live meetings for networking and sharing work in progress towards solving problems, and using journals for more complete work, rather than the current dominant computer science paradigm of main publications at conferences, where the publication is the key mechanism to demonstrate impact. The critique was to ask what kind of impact is it to have a paper accepted at a conference, when we truly want to see research having an impact not only for our co-researchers, but in the community.

4 References

1 Batya Friedman. Value-sensitive design. *interactions*, 3(6):16–23, December 1996.

2 Hiroshi Kawakami. Benefit of inconvenience for ambient interface. In *2011 IEEE/SICE International Symposium on System Integration (SII)*, pages 364–367. IEEE, December 2011.

3 Stephen Purpura, Victoria Schwanda, Kaiton Williams, William Stubler, and Phoebe Sengers. Fit4life: the design of a persuasive technology promoting healthy behavior and ideal weight. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 423–432. ACM, 2011.

4 William T. Riley, Daniel E. Rivera, Audie A. Atienza, Wendy Nilsen, Susannah M. Allison, and Robin Mermelstein. Health behavior models in the age of mobile interventions: are our theories up to the task? *Translational behavioral medicine*, 1(1):53–71, 2011.

5 Ben Shneiderman. Science 2.0. *Science*, 319(5868):1349–1350, March 2008.
DOI: 10.1126/science.1153539

Acknowledgements. We thank all the participants who attended the Perspectives Workshop. We also thank all our collaborators for their insightful comments and edits to this report.

Participants

- Lars L. Andersen
NRCWE – Copenhagen, DK
- Susanne Boll
Universität Oldenburg, DE
- Alan Chamberlain
University of Nottingham, GB
- Adrian David Cheok
City University London, GB
- Elizabeth F. Churchill
eBay Research Labs – San Jose, US
- Maria Francesca Costabile
University of Bari, IT
- Ed Cutrell
Microsoft Research India – Bangalore, IN
- Catalina Danis
IBM TJ Watson Research Center – Hawthorne, US
- Adrian Friday
Lancaster University, GB
- Katherine Isbister
New York University, US
- Wolfgang Maaß
Universität des Saarlandes, DE
- Florian Michahelles
Siemens Corp. – Berkeley, US
- Sean Munson
University of Washington – Seattle, US
- Les Nelson
Xerox PARC – Palo Alto, US
- Erika Poole
Pennsylvania State Univ., US
- Albrecht Schmidt
Universität Stuttgart, DE
- m. c. schraefel
University of Southampton, GB
- Katie A. Siek
Indiana University – Bloomington, US
- Gisela Sjogaard
University of Southern Denmark – Odense, DK
- Carlos Trenado
Uniklinikum Freiburg, DE