DAGSTUHL
REPORTS

Volume 3, Issue 12, December 2013

Dagstuhl Reports, Vol. 3, Issue 12 ISSN 2192-5283

ISSN 2192-5283

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

Publication date
March, 2014

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

Aims and Scope
The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.
In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:
- an executive summary of the seminar program and the fundamental results,
- an overview of the talks given during the seminar (summarized as talk abstracts), and
- summaries from working groups (if applicable).

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

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

Digital Object Identifier: 10.4230/DagRep.3.12.i
www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 13491

Computational Mass Spectrometry

Edited by

Ruedi Aebersold[1], Oliver Kohlbacher[2], and Olga Vitek[3]

1 **ETH Zürich / University of Zürich, CH,** aebersold@imsb.biol.ethz.ch
2 **Universität Tübingen, DE,** oliver.kohlbacher@uni-tuebingen.de
3 **Purdue University, US,** ovitek@stat.purdue.edu

Abstract

The last decade has brought tremendous technological advances in mass spectrometry, which in turn have enabled new applications of mass spectrometry in the life sciences. Proteomics, metabolomics, lipidomics, glycomics and related fields have gotten a massive boost, which also resulted in vastly increased amount of data produced and increased complexity of these data sets. An efficient and accurate analysis of these data sets has become the key bottleneck in the field. The seminar 'Computational Mass Spectrometry' brought together scientist from mass spetrometry and bioinformatics, from industry and academia to discuss the state of the art in computational mass spectrometry. The participants discussed a number of current topics, for example new and upcoming technologies, the challenges posed by new types of experiments, the challenges of the growing data volume ('big data'), or challenges for education in several working groups.

The seminar reviewed the state of the art in computational mass spectrometry and summarized the upcoming challenges. The seminar also led to the creation of structures to support the computational mass spectrometry community (the formation of an ISCB Community of Interest and a HUPO subgroup on computational mass spectrometry). This community will also carry on with some of the efforts initiated during the seminar, in particular with the establishment of a computational mass spectrometry curriculum that was drafted in Dagstuhl.

Seminar 1.–6. December, 2013 – www.dagstuhl.de/13491
1998 ACM Subject Classification J.3 Life and medical science
Keywords and phrases computational mass spectrometry, proteomics, metabolomics, bioinformatics
Digital Object Identifier 10.4230/DagRep.3.12.1

1 Executive Summary

All participants of Dagstuhl Seminar 13491

Motivation

Mass Spectrometry (MS) is an analytical technique of immense versatility. Detection of explosives at airports, urine tests for doping in sports, tests for cancer biomarkers in a clinic – all these rely on mass spectrometry as the key analytical technique. During the last decade, technological advances have resulted in a flood of mass spectrometric data (high-resolution mass spectrometry, mass spectrometry coupled to high-performance liquid chromatography – HPLC-MS). The publication of the first human genome in 2001 was a key even that enabled the explosive development of proteomics, which led to the conception of the Human

Proteome Project in 2010. Today, mass spectrometric techniques are an indispensible tool in the life sciences. Their development, however, is more and more hampered by the lack of computational tools for the analysis of the data. Modern instrumentation can easily produce data sets of hundreds of gigabytes from an individual sample. Most experimental groups are no longer able to deal with both the amount and the inherent complexity of these data. Computer science has the necessary tools to address these problems. It is thus necessary to intensify collaboration between the three key communities involved: life scientists applying MS; analytical chemists and engineers developing the instruments; computer scientists, bioinformaticians and statisticians developing algorithms and software for data analysis.

Goals

The seminar 'Computational Mass Spectrometry' is a follow-up seminar to the successful Dagstuhl seminars on 'Computational Proteomics (05471 and 08101). The different title was chosen to reflect the growing scope of computational mass spectrometry: from proteomics to metabolomic, lipidomics, and glycomics.

The goal of the seminar was thus to assess the state of the art for the field of computational mass spectrometry as a whole and to identify the challenges the field will be facing for the years to come. To this end we put together a list of participants covering both computational and experimental aspects of mass spectrometry from industry and academia from around the world. The result of these discussion should then be summarized in a joint status paper.

Results

The seminar was very productive and led to a number of tangible outcomes summarized below.

The Big Challenges

Not unexpectedly, it turned out to be difficult to identify the *big challenges* of the coming years and views on this differed quite a bit. After lengthy discussions, we were able to categorize the challenges. We are currently in the process of finalizing the draft of a paper on these challenges for computational mass spectrometry, which is supposed to be submitted by end of March 2014. The paper is a joint work of all the participants and will document the current state of the field. The challenges identified were the following:

Challenges of computational and statistical interpretation of mass spectra

- *Identification*
  Identification of analytes is still a challenge. In proteomics, the identification of post-translational modifications and of different proteoforms pose problems. Also the identification of non-tryptic peptides (peptidomics, MHC ligands) are interesting problems. Estimation of false-discovery rates based on target-decoy approaches has been criticized, but there is still a distinct lack of established alternatives. With the increasing interest in small-molecule mass spectrometry, the identification of metabolites, glycans, and lipids is increasingly becoming an issue and the algorithmic support for this is currently still lacking.
- *Quantification*
  Quantification faces challenges due to the – still-growing – diversity of experimental methods for analyte quantifications that necessitate a permanent development of new

computational approaches. There are also more fundamental, statistical problems, for example, inferring the absence of an analyte based on the absence of a signal. Quantification is also expected to contribute to the understanding of protein complexes and their stoichiometry.

Challenges arising from new experimental frontiers

- *Data-independent acquisition*
  The recent developments of data-independent acquisition techniques resulted in a set of entirely new computational challenges due to the different structure of the underlying data.
- *Imaging*
  Imaging mass spectrometry has become mature on the experimental side. The analysis of spatially resolved MS data, however, poses entirely new problems for computational mass spectromtry with increased complexity and data volume.
- *Single-cell mass spectrometry*
  Multi-parameter single cell mass spectrometry enables the characterization of rare and heterogeneous cell populations and prevents the typical averaging across a whole tissue/cell population. The key challenge will be the development of new computational tools able to define biologically meaningful cell types and then model the dynamic behaviour of the biological processes.
- *Top-down proteomics*
  Despite its obvious advantages of top-down approaches for functional proteomics, isoform identification and related topics, the approach suffers from unmet challenges on the computational side. Methods for mass spectrum deconvolution need to be improved and algorithms for the identification of multiple PTM sites are required.

Challenges of extracting maximal information from datasets

- *Democratization of data*
  Public availability of large datasets enables novel types of studies in computational mass spectrometry (data mining). The standardized deposition is and reliable repositories handling this data is still a major problem that needs to be addressed.
- *Integration of MS data with different technologies*
  Increasingly, computational biologists face data from multiple omics technologies. Integrating data from computational mass spectrometry across omics levels (genomics with transcriptomics, transcriptomics with proteomics, proteomics with metabolomics) poses a difficult data integration challenge, but will be essential for a more comprehensive view of the biological systems under study.
- *Visualization of heterogeneous data sets*
  The amount, structure and complexity of large-scale mass spectometric data turns out to be a challenging issue. While some end-users of theses methods tend to be interested in a final, aggregated result of a complex data analysis pipeline, it is often essential to analyze the data conveniently down to the raw spectra. Tools navigating these data sets on all levels are currently not yet available.

Community Building

It was felt among participants that computational mass spectrometry is lacking a structured community. Researchers in computational mass spectrometry come from diverse backgrounds: statistics, computer science, analytical sciences, biology, or medicine. Traditionally they are thus organized in different scientific societies, for example the International Society on Computational Biology (ISCB), the American Society of Mass Spectrometry (ASMS), the Human Proteome Organization (HUPO), the Metabolomics Society, and of course various national societies. Many participants attend both computational and experimental conferences in the area of mass spectrometry organized by these different organizations. Participants suggested to form subgroups for computational mass spectrometry in different societies. At the same time, in order to avoid duplication of structures and efforts, it was planned to share these subgroups across the different societies and establish joint chairs of these groups, organize joint workshops, and coordinate educational activities.

After the Dagstuhl seminar we contacted ISCB and HUPO to discuss the formation of these subgroups. After intensive discussion with the societies, HUPO and ISCB both agreed to this plan. A HUPO subgroup CompMS on computational mass spectrometry was formed. In parallel, ISCB agreed to form a *Community of Special Interest* (CoSI) CompMS. Both subgroups share a joint structure. A joint steering committee (Steering Committee (Oliver Kohlbacher, Olga Vitek, Shoba Ranganathan, Henning Hermjakob, and Ruedi Aebersold) has been established to guide both groups through their formation period. The groups have set up a joint mailing list, a website, and are currently planning initial kick-off meetings as satellite workshops to ISMB 2014 (Boston) and HUPO 2014 (Madrid).

Teaching Initiative

Recognizing the great need for educational materials for various audiences (bioinformaticians, biologists, computer scientists) some participants initiated an initiative to put these materials together as online courses. Discussions of this initiative have come quite far. It is currently planned to come up with a core curriculum for mass spectrometry. This core curriculum will be open for discussion within the computational mass spectrometry community. After the contents of the core curriculum has been established, tutorial papers will be solicited for the various modules of the curriculum. These papers will refer to each other, will use a coherent vocabulary and notation and will appear as a paper collection online in PLoS Computational Biology (edited by Theodore Alexandrov). Additional materials will be included, for example, online courses and lecture videos. An initial tutorial workshop is currently in planning to kickstart the further development of the curriculum.

Reviewing Guidelines

A working group (see Section 4.5) discussed the problems that computational papers face in the reviewing process. The main driver for this discussion was expediting the review process, specifically in terms of reducing the number of review cycles. It is worth noting that the Journal of Proteome Research (JPR), published by the American Chemical Society (ACS), presents a special case since this journal is the only one in the field that does not have a regular mechanism for the reviewers to see the comments of the other reviewers and the corresponding responses of the authors after each round of review. The proposal initially on the table was to share all reviews among reviewers and invite comments and changes before the first editorial decision is made for the first round of review. This system is, for instance, already in place via EasyChair (software used for the RECOMB meetings, but not for proteomics journals). After discussion, it was decided that it would clearly be beneficial if the JPR distributed all reviews among reviewers after each stage of revision. But it was felt that it would only be necessary to collect comments and feedback from the reviewers (based on sending them all reviews) before the editor reached an initial decision in cases where there was substantive disagreement among reviewers on critical points. These ad hoc communications can be handled in a semi-manual way within the existing manuscript management systems used by the proteomics journals, with the added benefit of maintaining an audit trail for the process. The reviewing guidelines developed by the participants in Dagstuhl are currently being discussed by with the editorial boards of different journals (currently J. Proteome Res and Mol. Cell. Prot.).

2 Table of Contents

3 Structure of the Seminar

The seminar was structured in an unusual way in so far as we omitted introductory talks by participants on purpose. While the organizers had prepared some material on the most obvious topics, we started the seminar in a very open manner and solicited proposals for break-out groups. The participants then voted on these topics.

3.1 Planned Topics

The topics initially planned to be covered were:

- **Identification from MS/MS spectra.** Although these topics received a lot of attention from the computational community in the past, many aspects of spectral identification are still in need of computational solutions. Open problems include the identification of high-quality spectra that are not identified by conventional methods, identification of metabolites, identification of post-translational modifications, MS/MS spectra from a mixture of sources, proteogenomics and metagenomics, metaproteomics, cross-linking data. These problems require advanced algorithmic solutions, in particular efficient search algorithms to deal with the combinatorial nature of the search space that needs to be explored during database searches.
- **Quantification.** There is currently a great diversity of sample preparation workflows and of mass spectrometry-based quantitative techniques, and these produce quantitative measurements with diverse properties and structure. Although statistical models and algorithms for analysis increasingly appear, there is currently no consensus on the appropriate analysis of these data. Open issues include the choice of normalization, handling of missing data, features of low quality and features with uncertain identity, and treatments of interferences (e.g., due to post-translational modifications).
- **Downstream interpretation of the identified and quantified proteins.** Many biological investigations require interpretations of mass spectra beyond the lists of identified and quantified proteins. Relatively little work has been done on the downstream interpretation of proteomic experiments so far. Open problems in biological applications include integration of data from heterogeneous sources (e.g., from transcriptomic, metabolomic and interactomic experiments) and with functional annotations, and dealing with novel data structures such as spectral images. Open problems in biomedical applications include the development of diagnostic and prognostic rules from the quantitative protein profiles, integration of the protein predictors with the clinical records, and assessment of the predictive ability. Solutions involve statistical inference and machine learning techniques.
- **Experimental design.** The choice of experimental design is critical for the success of all proteomic experiments, however many practitioners do not approach this choice from the computational and statistical point of view. The issues in need of computational development are the choice of type and number of the biological samples, and allocation of the experimental resources in space and time, guiding the choice of labeling strategies in label-based experiments, and the choice of target proteins, peptides and transitions in targeted experiments. Solutions to these problems involve approaches from statistical modeling and stochastic optimization.
- **Data management.** A major bottleneck in the analysis of proteomic experiments is the volume and complexity of the generated data. Although a lot of progress has been recently achieved in developing proteome-centric formats, there is an urgent need for the infrastructure for integrating measurements across experiments and data types. Most

currently available data repositories are genome-centric, and development of proteome-centric repositories is a complex but important task. Open issues in this field are related to databases and data mining.
- **Software engineering.** Even the best algorithms are of a limited practical use if they are not accompanied with the biologist-friendly software. The complexity of the analytical tasks creates new opportunities for development of comprehensive but modular pipelines, and standardization of analytical workflows.
- **Training of interdisciplinary experts.** The current and future computational approaches can rarely be used blindly, and should often be adapted to the experiment at hand. There is a strong need for training of interdisciplinary experts, and a Dagstuhl seminar can help identify the opportunity for such training.

3.2 Seminar Schedule

After a brief introduction the participants decided to form working groups to discuss and assess the state of the art as well as upcoming challenges for various topics. These working groups formed partially *ad hoc* and partially based on suggestion by previous working groups. The schedule of the seminar is shown below. Brief abstracts describing the conclusions of the individual working groups are reproduced in Section 6.

- **Monday**
  - Welcome and introductions
  - Formation of working groups (WGs)
  - WG 'New instrumentation challenges, quantitative proteomics, and data-independent acquisition'
  - WG 'Multiple Omics, Integration, and Metabolomics'
  - WG 'Democratization of proteomics data and big data'
- **Tuesday**
  - WG 'Teaching and Outreach'
  - WG 'Data-independent acquisition'
  - WG 'Guidelines for sustainable software and reviewing of computational MS' papers
  - WG 'Single-cell technologies and imaging'
  - WG 'Biological applications'
- **Wednesday**
  - Discussion of a first version of the challenges paper
  - Continuing discussions in the various working groups
  - Outing The usual Dagstuhl outing went to Trier, where the participants had a guided tour of the city, visited the Weihnachtsmarkt, and finally a wine tasting at one of the local wineries.
- **Thursday**
  - WG 'Non-DIA Quantification'
  - WG 'Algorithmic issues and big data in mass spectrometry'
  - WG 'Statistical methods and experimental design'
  - WG 'Categorization of challenges'
- **Friday**
  - Final discussion of the status paper, distribution of work
  - Discussions on the formation of a computational mass spectrometry community

Figure 1 Some impressions from the seminar and the outing in Trier (photos: Oliver Kohlbacher).

4 Working Groups

The various working groups formed and re-formed throughout the whole seminar. Each group reported on its results at the beginning of each day and several working groups came up with proposals for the formation of other groups. While some groups (e.g., data-independent acquisition) were active throughout the seminar, others were active for shorter periods of time only.

4.1 New Technologies and Data-independent Acquisition

Much of the discussion centered on whether data-independent acquisition (DIA) could replace data-driven acquisition (DDA) for identification purposes, or whether it was best suited as a targeted proteomics approach. It was felt by the group that at the moment, DIA is already a great alternative to SRM approaches, and indeed much of the published work is using DIA in this context. This does not mean that all the issues in using DIA as an alternative to SRM have been resolved, and signal processing is still difficult. A better framework to integrate data from different experiments (technical, biological replicates, etc.) is still needed.

Sample complexity and the type of experiments performed may drive the selection of using DIA for identification, SRM-like quantification, or both. The outcome of these discussions is that we should be thinking about different types of workflows as separate, and try to work out examples of the kind of computational challenges that are posed by the different types of experimental designs. Examples of such experimental samples to be worked out are clinical samples and other samples of high complexity, affinity purified samples and samples enriched for PTMs for which peptide level information is needed. Dataset size (meaning whether you are only analyzing a few samples, or hundreds of them) may also bring us to consider different computational approaches. This should help framing the next small group discussions.

We also briefly discussed how to discover, score and quantify PTMs from DIA data, but this is certainly a point that should be further discussed. An ongoing discussion is how starting from a peptide and looking for its presence (or evidence of absence) in a sample was different (conceptually and in terms of FDR calculation) from starting from the spectra (which could be virtual spectra) and performing a database search.

Identified issues are:

- signal processing issues (dynamic range estimates)?
- effects of complexity on identification and quantification?
- how to handle datasets of different sizes?
- how to score PTMs?
- what are the limitations due to current instrumentation and which of them will be overcome in the next generation(s) of instrumentation?

4.2 Multiple Omics, Integration, and Metabolomics

For a full understanding of biological complexity and functionality, system-wide identifications and quantification of all relevant players in a cell would be highly desirable. However, the various entities (genes, transcripts, proteins, metabolites, etc ...) all require different technological approaches for their assessment and quantification. Consequently, different research communities and scientific approaches have developed around each of these. “Multi-omics” aims to bridge and re-connect these different worlds, in order to take advantage of synergies and complementaries. For example, concomitant/parallel measurements of transcripts and proteins might help in removing false-positive hits, and can reveal which transcriptional events/isoforms gave rise to mature proteins and are hence perhaps more relevant functionally. Likewise, environmental proteomics experiments often require DNA sequencing to be conducted in parallel from the same sample, in order to construct matched ORF databases for peptide identifications. Most importantly, mechanistic and quantitative modeling of functional processes inside cells often requires precise quantifications of players of various types. What are currently the biggest obstacles for multi-omics integration? Mainly, common ground needs to be established with respect to nomenclature, statistics and reference datasets. Which representation can encompass all players? Which entities will be the “central” ones, where the others are attached? Rather than “linear” genome browsers, networks are likely to be an apropriate representation. In addition, novel software tools need to be developed for visualization purposes and dissemination, and algorithms developed for scoring datasets jointly, not separately. Most importantly, however, since few scientists are experts on multiple dimensions of -omics, renewed efforts need to go into providing meta-sites resource directories on available software, pipelines, tools and procedures.

4.3 Democratization of Proteomics Data and Big Data

The session on the democratization of proteomics data and big data discussed several topics related to overarching issues around computational proteomics. The focus was on teaching and outreach materials, logistics, quality control, peer reviewing and ethics and privacy issues. Two additional working groups were proposed as a result of this session, one on drafting a curriculum for teaching computational mass spectrometry and the other on drafting guidelines and manuscript types for computational contributions in proteomics journals.

4.4 Teaching and Outreach

The participants of the seminar felt that at the moment there is a lack of teaching material to bring interested students and experimentalists quickly into the field of computational proteomics. In addition it was felt that this is also detrimental for the external perception of the field. The group thought it would be beneficial to come up with a curriculum for an online course which:

- allows people from various backgrounds to enter the field
- explains the main concepts as opposed to operating certain programs
- is set up as a backbone of main lectures complemented by some prerequisite courses to homogenize assumed knowledge in biochemistry or mathematics and by sidetrack courses that go in-depth (e.g., details of HPLC or how to solve a specific algorithmic problem efficiently)

In the discussion it was decided to discuss the content of such a curriculum (draft is in the seminar wiki). In order to extend this, the plan was to identify potential funding souces for an experienced postdoc as an editor of this course (homogenize nomenclature, provide templates, technically implement it, approach contributors and reviewers, etc.). This effort should be coordinated within the scientific societies, for example with existing efforts at HUPO (or elsewhere if they are presently not known). The group produced a first draft of the core topics of such a curriculum (see Appendix A).

4.5 Guidelines for Sustainable Software and for Reviewing of Computational MS papers

The session on guidelines for sustainable software and reviewing of computational mass spectrometry papers focused on the specific topic of improving the guidelines (for authors as well as for reviewers) that apply to computational contributions to proteomics journals. An important aspect of the discussion concerned the ability to provide a place for all possible contributions, including highly innovative but immature approaches, highly mature but incrementally innovative tools, and both highly innovative as well as mature contributions. This resulted in three manuscript types, matched to already existing manuscript categories for logistic convenience at the journals, and clarity resulting from the already defined scope: Brief Communication, Application/Technical Note, and Research Article. Furthermore, a request was made to introduce a new subtype for two of these types, that would bear the 'stamp of approval' resulting from passing the checklist. The ability of journals to accommodate this request is to be investigated, and the specific title they would carry is to be decided.

4.6 Single-cell Technologies and Imaging

Understanding molecular characteristics in dynamic and heterogeneous cell populations on the levels of single-cell and cell populations is an essential challenge of biology. Mass cytometry is an emerging technique for single-cell analysis allowing analysis of up to tens of molecules simultaneously either in suspension or directly on tissue. Imaging MS is a maturing technique for spatially-resolved mass spectrometry analysis directly on tissue. Both techniques provide tools for molecular imaging and allow for unprecedented insight into biological organization of biological processes on the levels of individual cells and cell populations and paving the way to spatial pathway analysis. We outlined the challenges which are faced by these young and promising techniques.

4.7 Biological Applications

After discussing some definitions (who are our 'customers' – example biologist name 'Steve', who are the 'computational mass spectrometrists'), current challenges on the biological side were identified:

1. Communication about the biological question
   This is a challenge for us and Steve. He has to take time to educate us, we'll end up very experienced biologists. This is as important for the data collectors as the data processors. Challenges for bioinformatician: communication with Steve, establishing end point, educating future Steves.
2. Triaging
   Will it ever work? Is there a better method? Therefore we have to have a good idea of alternative technologies.
3. Experimental design
   How can you design the critical experiment to give a clear answer? We have to manage Steve's expectation – this is a challenge in itself. Define bioinformatics pipeline and endpoint to determine whether the design is sufficient and the experiment is feasible.
4. Knowing and selling our unique selling points
   - The read out of translational control and protein degradation: (a) is a protein translated, if so, how much of it and (b) is a protein isoform translated

  - Subcellular/organelle (complementary methods would be: immunocytochemistry, confocal microscopy)
  - post translational modifications/truncations
  - protein interactions

These biological applications result in several grand challenges for computational MS:

- System wide probing of fluxes
- Dynamics of everything (multiple layers)
- Coverage, comprehensive measurement of the proteome and isoformome
- Combinatorial nature of PTMs – going beyond the single PTM
- Epitope mapping – MHC ligands demand de novo identification

4.8 Non DIA-quantification

Quantitation is fundamental to characterizing biological material with mass spectrometry. Despite major advances made over the past 15 years, particularly with respect to proteomics, significant obstacles remain that inhibit our ability to comprehensively and unambiguously evaluate our data, and share them with others. We describe here five critical computational challenges that underly all quantitative techniques. Solving them will promote wider dissemination of high-confidence results, and deeper biological understanding.

4.9 Algorithmic Issues and Big Data in MS

One key issue arises from big data in computational mass spectrometry: finding structure in the big data sets. In proteomics, a key statistical challenge is still the determination of FDRs beyond the target-decoy approach. Also the inclusion of structural data as part of protein inference is a big challenge. Identification of proteins can be improved by inclusion of genomic variants, algorithms to deal with PTMs, and annotation problems. New algorithms are also required to analyze non-shotgun data and top-down data. Finally, the visualization of all levels of the data poses interesting problems.

4.10 Statistical Issues and Experimental Design

Statistics is key for reproducible research. It is central to experimental design, data processing and downstream analysis. In some applications, there are available statistical workflows that can still be improved, but are not perceived as a grand statistical challenges, for example, confidence in peptide identification in a DDA-type workflow. There are other areas where even basic tools do not exist, and the need for these is more pressing (e.g., new types of experimental workflows are in need of solutions). Furthermore, perhaps one of the most important challenges is the communication and training of good statistical principles.

A A draft curriculum for computational mass spectrometry

- Intro, Goals, of (Computational) Proteomics and Metabolomics
  - What is the proteome, metabolome, transcriptome
  - What is it good for? (Protein ID, Quantitation for pathway analysis, Biomarker analysis, maybe concrete examples, e.g. protein interaction)
  - What skills do you need?
- Basic Biochemistry (PR)
  - DNA, Proteins, PTMs
  - Lipids, Metabolites
  - Degradation
- Analytical Chemistry
  - Precision, accuracy
  - Dynamic range
  - LOD/LOQ (noise)
  - Profiling vs. Targeted
- Statistics
  - Distributions
  - p-value
- Needed Computational concepts
  - Algorithms, run time, space requirements
  - Machine learning
- Mass spectrometry
  - LC/GC/CE
  - MS (precurscor selection)
  - MS/MS (ion types, ion ladders
  - Resolution, accuracy
  - Isotopes (de-isotoping, deconvolution)
  - Fragmentation patterns (CID, ECD, peptides vs. lipids, glycans, etc..)
  - MS of biomolecules
- Peptide ID methods
  - DB Search
    - perfect spectra, ion ladders
    - what can wrong: noise, missing ions, PTMs
  - DeNovo (DeNovo vs DB search)
  - Spectral libraries
  - Common mistakes
- Protein inference
  - Coverage, shared peptides, one-hit wonders
  - standard inference approaches
- Quantification
  - different strategies (labeled, unlabeled (in-vivo, in-vitro))
  - absolute vs relative
  - main algorithmic concepts (alignment, model fitting, peak integration)
- Validation/Error control
  - FDR, decoy DB
  - Quality control
- Interpretation of results

  - Pathways enrichment
  - How to extract biology of the data
- Data management/DBs
  - Repositories
  - Standard formats (mzML, etc.)
  - Metadata
- Strategies
  - Targeted vs non targeted
  - Top-Down, Bottom-up
- Experimental design (ST)
  - technical, biological replicate
  - variability, number of samples
  - multiple testing, pooling

Participants

- Rudolf Aebersold
ETH Zürich, CH
- Theodore Alexandrov
Universität Bremen, DE
- Dario Amodei
Stanford University, US
- Bernd Bodenmiller
Universität Zürich, CH
- Sebastian Böcker
Universität Jena, DE
- Karsten Boldt
Univ.-Klinikum Tübingen, DE
- Daniel R. Boutz
University of Texas – Austin, US
- Julia Burkhart
ISAS – Dortmund, DE
- Manfred Claassen
ETH Zürich, CH
- John Cottrell
Matrix Science Ltd. – London, GB
- Eric Deutsch
Institute for Systems Biology – Seattle, US
- Joshua Elias
Stanford University, US
- David Fenyö
New York University, US
- Anne-Claude Gingras
University of Toronto, CA
- Henning Hermjakob
EBI – Cambridge, GB
- Lukas Käll
KTH Royal Institute of Technology, SE
- Sangtae Kim
Pacific Northwest Nat'l Lab., US
- Oliver Kohlbacher
Universität Tübingen, DE
- Theresa Kristl
Universität Salzburg, AT
- Bernhard Küster
TU München, DE
- Henry Lam
The Hong Kong Univ. of Science & Technology, HK
- Wolf D. Lehmann
DKFZ – Heidelberg, DE
- Kathryn Lilley
University of Cambridge, GB
- Michal Linial
The Hebrew University of Jerusalem, IL
- Mike MacCoss
University of Washington – Seattle, US
- Brendan MacLean
University of Washington – Seattle, US
- Alexander Makarov
Thermo Fisher GmbH – Bremen, DE
- Lennart Martens
Ghent University, BE
- Sara Nasso
ETH Zürich, CH
- Alexey Nesvizhskii
University of Michigan – Ann Arbor, US
- Steffen Neumann
IPB – Halle, DE
- William Stafford Noble
University of Washington – Seattle, US
- Paola Picotti
ETH Zürich, CH
- Knut Reinert
FU Berlin, DE
- Bernhard Renard
RKI – Berlin, DE
- Hannes Röst
ETH Zürich, CH
- Stephen Tate
AB SCIEX – Concord, CA
- Andreas Tholey
Universität Kiel, DE
- Henning Urlaub
MPI für Biophys. Chemie – Göttingen, DE
- Olga Vitek
Purdue University, US
- Christian von Mering
Universität Zürich, CH
- Susan T. Weintraub
The University of Texas Health Science Center, US
- Witold E. Wolski
ETH Zürich, CH
- René Zahedi
ISAS – Dortmund, DE

Report from Dagstuhl Seminar 13492

Geosensor Networks: Bridging Algorithms and Applications

Edited by

Matt Duckham[1], Stefan Dulman[2], Jörg-Rüdiger Sack[3], and Monika Sester[4]

1 **The University of Melbourne, AU,** `mduckham@unimelb.edu.au`
2 **TU Delft, NL,** `s.o.dulman@tudelft.nl`
3 **Carleton University, Ottawa, CA,** `sack@scs.carleton.ca`
4 **Leibniz Universität Hannover, DE,** `monika.sester@ikg.uni-hannover.de`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13492 "Geosensor Networks: Bridging Algorithms and Applications." New geosensor networks technologies have the potential to revolutionize the way we monitor and interact with the world around us. The objective of the seminar was to move closer to realizing this potential, by better connecting theoretical advances with practical applications and education. The Seminar ran from 1–6 December 2013, and brought together 21 participants from around the world, representing wide variety of disciplinary backgrounds and expertise connected with geosensor networks. While these discussions are continuing to develop and bear fruit, this report summarizes the results of the discussions held at the seminar.

Seminar 1.–6. December, 2013 – www.dagstuhl.de/13492

1998 ACM Subject Classification C.2.1 Network Architecture and Design, E.1 Data Structures, H.2.8 Database Applications, I.2.9 Robotics, I.2.11 Distributed Artificial Intelligence, K.4.1 Public Policy Issues, K.4.2 Social Issues, K.6.4 System Management

Keywords and phrases amorphous computing, decentralized spatial computing, distributed algorithms, location privacy, organic computing, self-organization, sensor/actuator networks, situation awareness, smart materials, spatial analysis

Digital Object Identifier 10.4230/DagRep.3.12.17

1 Executive Summary

Matt Duckham
Stefan Dulman
Jörg-Rüdiger Sack
Monika Sester

The aims of Dagstuhl Seminar 13492, "Geosensor Networks: Bridging Algorithms and Applications," were to advance research into, and application of geosensor networks by enhancing interdisciplinary and cross-domain collaboration. The premise of the Seminar was that the potential for useful, practical applications of geosensor networks (wireless sensor networks tasked with monitoring changes in geographic space) are being held back by the enormous diversity of applications and expertise connected with different facets of geosensor networks. The result today is many niche solutions to specific problems, where what is needed are a few general solutions to broader problems.

More specifically, the diversity of concepts, approaches, and tools used in connection with geosensor networks is inhibiting more rapid and fruitful research and development. Examples of this diversity include:

- *Ontologies, representations, and models*: The diversity of uses of geographic information leads inexorably to diversity in the different ontologies, representations, and models commonly used to conceptualize that information.
- *Algorithms and data structures*: The plethora of models, algorithms, data structures, and architectures that exist in the literature are frequently incompatible, founded on divergent assumptions and inconsistent approaches.
- *Applications*: While the potential applications of geosensor networks are legion, today we lack agreement on a set of applications that together encompass and illustrate the bulk of issues faced by all applications.
- *Benchmarks, tools, and technologies*: Perhaps more than any other single issue, a lack of consensus on core benchmark data sets, problems, simulation systems, and software tools inhibits convergence in research and application.

Participants

To begin to address this diversity, and bridge the gap between theory and application, the Seminar participants represented a broad spectrum of disciplines, including computer science, geographic information science, computational geometry, statistics, artificial intelligence, pervasive computing. The seminar had strong groundings in previous Dagstuhl Seminars, including Seminars 10491 and 12512 on Representation, Analysis and Visualization of Moving Objects, and Seminar 06361 on Computing Media and Languages for Space-Oriented Computation. However, we were very pleased that the Seminar also attracted a significant proportion of newer Dagstuhl attendees: more than half the attendees had attended at most one Dagstuhl seminar before, with around one quarter of Seminar participants attending their first ever Dagstuhl.

Bringing together this diversity of backgrounds, expertise, and experiences was central to the core aim of building bridges between related fields, and was central to the success of the Seminar.

Format

The seminar was structured around three complementary perspectives: models and algorithms; benchmarking and applications; and teaching and curricula. The objective of the models and algorithms perspective was to survey, catalog, and compare the ontologies, representations, algorithms, and data structures that are fundamental to computing in geosensor networks. Through the benchmarks and application perspective, the seminar aimed to improve comparability and compatibility in models and algorithms, as well as connect existing models and algorithms more directly to practical uses. Focal applications included emergency response, intelligent transportation, smart materials, and environmental monitoring. As a capstone, the teaching and curricula perspective aims to distill and collate the collected expertise in models, algorithms, benchmarks, and applications into a coherent body of knowledge: a "library" of core concepts and techniques for computation with and application of geosensor networks.

The seminar focused less on presenting individual lectures than on achieving its objectives above through collaborative discussions and activities. The organizers invited three speakers with diverse backgrounds to give longer talks (40 minutes) and lead the subsequent discussions. The three speakers were René Doursat (CNRS, Paris, discussing organic computing), Thomas Kirste (Universität Rostock, discussing situational awareness and intention recognition using sensed data), and Edzer Pebesma (Universität Münster, on spatial data analysis with sensor data). As well as providing an introduction to the breadth and depth of ideas related to the field, the speakers were able inspire the participants and spark many subsequent discussions.

The majority of the seminar then focused on workshop-style discussion and break-out groups. In this way the seminar aimed to elicit answers to the question of what are the essential elements of computing with geosensor network. The aim was be to advance the field through consensus on priorities as well as providing opportunities for new innovations to emerge from new collaborations. The working groups' discussions and conclusions are summarized in this report (Section 5). Broadly, the working groups' focuses included spatial computing (e.g., self-organization and smart materials); applications of sensor networks (e.g., developing countries and big data); social issues (e.g., privacy); education (e.g., teaching resources and curriculum); and data and benchmarking.

However, even though the primary focus was on discussion and collaboration, the program still allowed time for short focus talks from participants (up to 10 minutes for senior researchers, or up to 15 minutes for junior researchers—researchers were able to self-select as to whether they regarded themselves as junior or senior). A summary of the focus talks given by participants is also contained in this report. All the speakers were asked to address one of the three Seminar perspectives (models and algorithms; benchmarking and applications; and teaching and curricula), as can be seen from section 4.

Outcomes

The participants were highly satisfied with the quality of the seminar. Many and diverse research results were presented during the Seminar, surveyed in the following sections. As with many Dagstuhl Seminars, the new collaborations and results of those collaborations are ongoing. However, amongst the key findings and ongoing collaborations, we highlight:

- Considerable progress has been made in recent years in the areas of (decentralized) spatial computing. This includes advances in the bottom-up design of distributed and decentralized algorithms. However, in contrast top-down aggregate programming techniques offer an important advantage over more conventional decentralized programming, in that they are substantially less complex for developers to use. Whatever future advances in this area may hold, the Seminar participants were agreed in that decentralization is a means, but never an ends. A focus on the behavior of a distributed geosensor network as a whole, rather than the rules required to generate that behavior, should always be the focus. In this respect, the focus on emergent behaviors found in spatial computing would seem ideally suited.
- One particular area of progress at the seminar was in teaching and curriculum. Despite the wide range of expertise and academic backgrounds of the participants, the Seminar exposed the considerable commonality and agreement around the fundamental concepts behind geosensor networks. This convergence was evident in the recent development of tools (such as the Proto aggregate programming language) and the publication of text books on the subject of geosensor networks.

- In contrast, one area of particular difficulty was in benchmarking and data sets. The availability of data sets is unquestionably increasing, as evidenced by several different data sets that were made available at, and through the preparatory work by participants in advance of the Seminar. However, the wide diversity of requirements for data sets across different applications continues to defy standardization or convergence on a small set of benchmarks. Issues such as validation and ground truthing; requirements for massive data sets with thousands or millions of sensors; metadata and provenance; and privacy issues were all various inhibitors to the development of a small set of benchmark data sets.
- Finally, the Seminar highlighted the numerous practical, social, and environmental challenges that still remain to truly bridging the gap between theoretical algorithms and practical applications, such as cost and deployment strategies, privacy, and environmental pollution. Although these issues remain largely unsolved longer-term research problems, they are already the explicit focus of several new collaborations that have directly resulted from this Seminar.

2 Table of Contents

3 Invited Talks

3.1 Architectured self-organized systems: Toward the best of both worlds by "morphogenetic engineering"

René Doursat (CNRS, FR)

Engineering is torn between an attitude of strong design and dreams of autonomous devices. It wants full mastery of its artifacts, but also wishes these artifacts were much more adaptive or "intelligent." Meanwhile, the escalation in system size and complexity has rendered the tradition of rigid top-down planning and implementation in every detail unsustainable. In this context, natural complex systems (large sets of elements interacting locally and behaving collectively) can constitute a powerful source of inspiration and help create a new generation of "self-x" systems, these properties being mostly absent from classical engineering.

This talk showed other avenues of bio-inspired design stressing the importance and benefits of a genuine *self-organization in architectured systems*——as exemplified by the growth of multicellular organisms or the nests of social insects. A new field of research was presented, "morphogenetic engineering" (ME; http://doursat.free.fr/morpheng.html), which explores the artificial design of complex morphologies that can reproducibly arise without centralized or externalized control. Potential applications range from swarm robotics to distributed software, techno-social networks and geosensor networks to synthetic biology. What they have in common is a myriad of hardware/software/bioware/geoware agents that can be programmed to dynamically build structures on the sole basis of peer-to-peer communication and local computation.

3.2 Symbolic behavior models for intention recognition in situation-aware assistance

Thomas Kirste (Universität Rostock, DE)

One objective of situation aware assistance is to support persons that need to perform structured activities by providing guidance and automation. In general, this requires the estimation of the actual activity sequence and the underlying intentions, from noisy and ambiguous sensor data. Unfortunately, obtaining the training data required for building probabilistic inference systems that employ sequential state estimation is often very expensive. Therefore, it is of interest to use prior knowledge on the causal structure of action sequences for building a system model. Symbolic causal models are one option for doing this, for instance based on the STRIPS paradigm used in the domain of symbolic planning. This talk outlined the basic mechanisms for attaching a probabilistic semantics to such a symbolic model and discussed the applicability of this approach to activity recognition problems of realistic complexity. The results obtained on the reconstruction of everyday activities from wearable accelerometer data (specifically, an instrumental activity of daily living) suggest that such problems can indeed be successfully tackled by symbolic modeling methods.

3.3 Sensor networks from a data analytic perspective

Edzer Pebesma (University of Münster, DE)

This talk surveyed a number of distinct practical use cases over the past 20 years connected with sensor networks and analysis of the data they produce.

In the first use case [1] we analyze data from groundwater quality monitoring networks, which are sampled yearly to assess the availability and development of long-term quality water resources. We tried to map spatial variability of groundwater quality and its dynamics, and found that for several components (e.g. nitrate) either the noise was too high, or the Nyquist frequency was not met and predictions were highly inaccurate, even when predicting average concentrations for 4km $\times$ 4km blocks. By only presenting results in terms of 95% prediction intervals, we avoided the misinterpretation of highly uncertain predicted values.

In the second use case [2], the goal was to evaluate, optimize or design from scratch a monitoring network that should detect nuclear outbreaks from various possible origins in an as short as possible time, on a country-wise or region-based basis. This led to a spatial design problem, and by representing space as a regular grid on a fine-enough resolution, to a discrete optimization problem. The implicit random field to be sampled was generated by deterministic plume models, covering a relevant range of sources and weather conditions.

In the third use case [3], we evaluated rural background air quality data collected at a European scale, and assessed whether for the interpolation of yearly aggregated values one should model and interpolate the daily values directly, or rather aggregate station time series to yearly aggregates and interpolate those.

The fourth use case addresses ambient variables and car properties (e.g. temperature, fuel consumption) measured from cars, and collected on a data sharing site called envirocar.org[1]. Understanding data from mobile sensors, interpreting and analyzing them poses new problems, not addressed in mainstream spatial statistics literature. Analyzing such data also leads to the problem which spatial predictions (interpolations) and aggregations are meaningful, and which are not [4]: we can interpolate car engine temperatures, but doing this for a location and time where there is no car is meaningless. Averaging over several cars may also be not meaningful.

Finally, the talk addressed how collaborative software development takes place in the R project [5], where sharing software and analysis scripts has become the rule, rather than the exception. This opens up solutions to the problem of reproducing scientific results, as far as they concern computational aspects, as well as benchmarking computational procedures.

References

1 Pebesma, E.J. and J.W. de Kwaadsteniet. Mapping groundwater quality in the Netherlands. *Journal of Hydrology*, 200:364–386, 1997.
2 Helle, K.B. and E.J. Pebesma. Stationary sampling designs based on plume simulations. In Jorge Mateu and Werner G. Müller (eds.) *Spatio-temporal Design: Advances in Efficient Data Acquisition*, Chapter 14, 2012.
3 Gräler, B., L. Gerharz, and E.J. Pebesma. Spatio-temporal analysis and interpolation of PM10 measurements in Europe. ETC/ACM Technical Paper 2011/10; Released: 2012/01/30.

[1] See http://www.envirocar.org/ and http://rpubs.com/edzer/envirocar2

4 Stasch, C., S. Scheider, E.J. Pebesma, and W. Kuhn. Meaningful spatial prediction and aggregation. *Environmental Modelling & Software*, 51:149—165, 2014.
5 Bivand, R.S., E.J. Pebesma, and V. Goméz-Rubio *Applied Spatial Data Analysis with R*, Second edition. Springer, NY, 2013.

4 Focus Talks

Theme: Models and Algorithms

4.1 Data mining techniques in sensor networks: Summarization, interpolation, and surveillance

Annalisa Appice (University of Bari, IT)

Joint work of Annalisa Appice, Anna Ciampi; Fabio Fumarola; Donato Malerba
Main reference A. Ciampi, D. Malerba, F. Fumarola, A. Appice, " Data Mining Techniques in Sensor Networks: Summarization, Interpolation and Surveillance," SpringerBriefs in Computer Science, 105p., ISBN 978-1-4471-5454-9, Springer, 2014.
URL http://www.springer.com/computer/database+management+%26+information+retrieval/book/978-1-4471-5453-2

Sensor networks consist of distributed devices, which monitor an environment by collecting data (light, temperature, humidity, ...). Each node in a sensor network can be imagined as a small computer, equipped with the basic capacity to sense, process and act. Sensors act in dynamic environments, often under adverse conditions.

Typical applications of sensor networks include monitoring, tracking, and controlling. Some of the specific applications are photovoltaic plant controlling, habitat monitoring, traffic monitoring, ecological surveillance. In these applications, a sensor network is scattered in a (possibly large) region where it is meant to collect data through its sensor nodes.

While the technical problems associated with sensor networks have reached a certain stability, managing sensor data brings numerous computational challenges in the context of data collection, storage, and mining. In particular, learning from data produced from a sensor network poses several issues: sensors are distributed; they produce a continuous flow of data, eventually at high speeds; they act in dynamic, time-changing environments; the number of sensors can be very large and dynamic. These issues require the design of efficient techniques for processing data produced by sensor networks. These algorithms need to be executed in one step of the data, since typically it is not always possible to store the entire data set, because of storage and other constraints.

The focus of this talk was to provide an idea of data mining techniques in sensor networks. We have taken special care to illustrate the impact of data mining in several network applications by addressing common problems, such as data summarization, interpolation, and surveillance. We propose a clustering technique to summarize data and permit the storage and querying of this amount of data, produced by a sensor network in a server with limited memory. Clustering is performed by accounting for both spatial and temporal information of sensor data. This permits the appropriate trade-off between size and accuracy of summarized data. Data are processed in windows. Trend clusters are discovered as a summary of each window. They are clusters of georeferenced data, which vary according to a similar trend along the time horizon of the window [2]. We can use trend clusters to interpolate missing data. The estimation phase is performed by using the *inverse distance weighting approach* [1]. Finally, we can also use trend cluster for fault detection in photovoltaic plants [3].

References

1 A. Appice, A. Ciampi, D. Malerba, and P. Guccione. Using trend clusters for spatiotemporal interpolation of missing data in a sensor network. *Journal of Spatial Information Science*, 6(1):119–153, 2013.

2 A. Appice, A. Ciampi, and D. Malerba. Summarizing numeric spatial data streams by trend cluster discovery. *Data Mining and Knowledge Discovery* 2013, in press.

3 A. Ciampi, A. Appice, D. Malerba, and A. Muolo. An intelligent system for real time fault detection in PV plants. *Smart Innovation, Systems and Technologies*, 12:235–244, 2012.

4.2 Programming distributed algorithms using computational fields

Jacob Beal (BBN Technologies, Cambridge MA)

A geosensor network can typically be viewed as a spatial computer: a collection of devices distributed through space such that the difficulty of moving information between devices is strongly dependent on the distance between them, and the "functional goals" of the system are generally defined in terms of the system's spatial structure.

It is often useful to use decentralized approaches in implementing such networks, particularly when network infrastructure is unavailable or unreliable. Pragmatically, however, the difficulty of implementing complex and reliable distributed algorithms has formed a major barrier to using such techniques.

Aggregate programming techniques can lower this barrier, by allowing engineers to program a geosensor network as though it were centralized, and then automatically compile that program into an adaptive distributed algorithm implementing the equivalent.

We have found much success in using a continuous space abstraction, the amorphous medium, which views a geosensor network as a discrete sampling of the continuous space that is being monitored. Field calculus provides a space-time universal basis set for aggregate programming under this model, which enables a number of forms of implicit safe composition, adaptivity, and scalability. We have elaborated this model into an implementation as the Proto programming language (http://proto.bbn.com), and applied this to programming a broad range of platforms, including sensor networks, swarm robotics, agent-based simulations, and engineered biological cells.

4.3 Smart materials: From sensor networks to computational meta-materials

Nikolaus Correll (University of Colorado, Boulder CO)

Joint work of Apice, Annalisa; Beal, Jacob; Correll, Nikolaus; Dulman, Stefan; Doursat, René; Papp, Zoltan; Sack, Jörg-Rüdiger; Sester, Monika

Materials can become truly "smart" by combining properties of the carrier polymer with sensing, actuation, and computational, networked logic. Depending on the complexity of the underlying distributed algorithms, such materials can be purely reactive like the human skin, implement distributed processing and actuation like the retina or colon, respectively, or even become conscious like the human brain. Research in smart materials and their application will be driven by the desire for multi-functionality of existing materials and surfaces. For example, a vinyl floor can easily be extended by pressure sensors and colored lights, enabling applications such as emergency routing, indoor navigation, occupancy mapping or playful activities using comparably simple, gradient-based distributed algorithms. Here, the additional cost for sensing, actuation, and computation might become marginal given the resulting multi-functionality. Similarly, enhancing fabrics by sensing, actuation and computation might enable a new generation of garments that can not only monitor, but affect posture and physiological parameters of their wearer. Other functional materials might have the ability to shape-change, monitor structural health and self-repair, or even assemble and disassemble, allowing repurposing of an object's functionality and its enclosing space. Scalability and robustness requirements of such materials strongly motivate the use of fully distributed, self-organizing algorithms, many of which have begun to be studied by the swarm robotics, sensor network, and computational geometry communities. In addition to challenges in material science engineering and manufacturing, truly smart materials will require composing the existing suite of distributed algorithms into aggregates of increasing complexity that run on larger numbers of computational nodes with minimalist capabilities.

4.4 Organic building evacuation support system: A vision

Sabrina Merkel (KIT: Karlsruhe Institute of Technology, DE)

A rapidly growing world population and increasingly dense settlements demand ever-larger and more complex buildings from architects and engineers. However, current evacuation support systems in these buildings are comparatively outdated. Their analogous and static character makes evacuation support easily overlooked in panic and prevents route guidance from being adaptive to the current evacuation situation, such as recognizing and avoiding congestions. The Organic Building Evacuation Support System (OBESS) introduced in this talk is a concept for solving these issues by using mobile devices to guide people to safe exits. The observer/controller architecture from the research area of organic computing was introduced. It was shown how this architecture can be applied to mobile devices in order to achieve adaptive and self-organized localization and evacuation path planning that is robust and controllable at the same time. Furthermore, OBESS provides for a central control unit, which is meant to perform building-specific optimization in order to adapt the system to the building's individual requirements. One example for such an optimization approach is the improvement of the positions of anchor nodes in the building, which are used for localization of the mobile devices. In this talk, an evolutionary algorithm developed for this purpose was described. Moreover, four fitness evaluation criteria are introduced which are tested in experiments. The presented research shows that optimizing the coverage of anchor nodes is not enough to yield good localization results and that the localization algorithm, as well as the applied performance metric, has crucial impact on optimal anchor placements. Apart from the presented research, a variety of research topics that arise in this area are proposed in this talk, which have to be addressed in order to achieve a long overdue changeover in evacuation support of modern buildings.

Theme: Applications and Benchmarking

4.5 Geosensor networks applications

Steve Liang

This session discussed the potential for a "killer app" in geosensor networks. The session started by identifying two real-world applications:

1. monitoring wild mammal populations and migration in Northern Alberta [1] (more specifically, monitoring woodland caribous and wolves); and
2. pipeline safety monitoring.

Both applications present similar challenges, including the remote sites, power constraints, extreme weather, and potentially high maintenance costs, etc.

Domain knowledge plays critical roles for each both applications. For example, domain knowledge is required to determine the parameters to be observed and the installation location of the sensor nodes. Deploying large-scale geosensor networks in real-world remains a challenging task.

Other potential killer applications might include "smart space garbage": how might decentralized geosensor networks be used to track debris and detritus in the space?

References

1 Corey J.A. Effects of petroleum exploration on woodland caribou in northeastern Alberta. *The Journal of Wildlife Management*, 61(4):1127–1133, 1997.

4.6 Runtime reconfiguration in sensor/actuator networks

Zoltan Papp (TNO, The Hague, NL)

In order to address today's societal challenges (e.g. environmental management, intelligent mobility, safety/security, sustainable urban living, etc.) monitoring and control of large-scale, complex, dynamical systems are fundamental technologies. These monitoring and control systems realize sophisticated signal processing algorithms (state estimation, pattern recognition, optimal control, decision making, etc.) in distributed configurations and many times are deployed in difficult to access, hostile environments. Under these circumstances failures, changes in operational conditions and changes in user needs are not exceptions but should be considered as nominal operation. Runtime adaptation capabilities (covering self-organization, self-optimization) have to be realized to extend the operational envelope of the system.

The design of runtime adaptable systems poses new challenges both for the signal processing and for the system architecture aspects. The presentation surveyed the approaches for automatic runtime adaptation and identifies the design and implementation challenges. A model-based methodology was introduced, which greatly simplifies the development of these complex artifacts and thus enables the successful introduction of the runtime adaptation concept in various of application domains. The distinguishing features of the targeted systems are serious resource constraints (e.g., processing capabilities, communication, energy supply) and the presence of demanding non-functional requirements, such as timeliness, robustness, lifetime and, the capability of handling system "evolution." The "practical aspects" the runtime reconfiguration were also addressed (representation, reasoning and developer support). The talk concluded with identifying a number of open issues, such as resource aware and distributed algorithms for optimization, decision making and control, measurement of "costs" and "usefulness," guaranteeing pre-defined closed-loop behavior.

4.7 Live geography: Fusing data from technical and human sensors

Bernd Resch (University of Heidelberg, DE)

The predicted rise of geo-sensor webs has not taken place as quickly as estimated, which prevents many research efforts from being carried out due to lacking real-time base data. Additionally, we are currently witnessing the rapid emergence of user-generated data in a variety of geo-social networks, the mobile phone network, or micro-blogs. This human-generated data can potentially complement sensor measurements to a significant degree.

In terms of user-centered sensing approaches, three concepts can be distinguished:

1. "People as sensors" defines a measurement model, in which measurements are not only taken by calibrated hardware sensors, but in which also humans can contribute their subjective "measurements," such as their individual sensations, current perceptions, or personal observations.
2. "Collective sensing" tries not to exploit a single person's measurements and data, but analyzes aggregated anonymized data coming from collective sources, such as Twitter, Flickr, or the mobile phone network.
3. "Citizen science" stands for a human-based approach to science where citizens contribute semi-expert knowledge to specific research topics.

Particular challenges in fusing data from human and technical sensors include standardization (on data, service and method levels), data assimilation (resolution, aggregation, etc.), the combination of methods from geoinformatics and computer linguistics (to extract information from user-generated data), quality assurance (both for technical and human sensors), and the consideration of privacy issues (data ownership, storage, optimum aggregation levels, etc.).

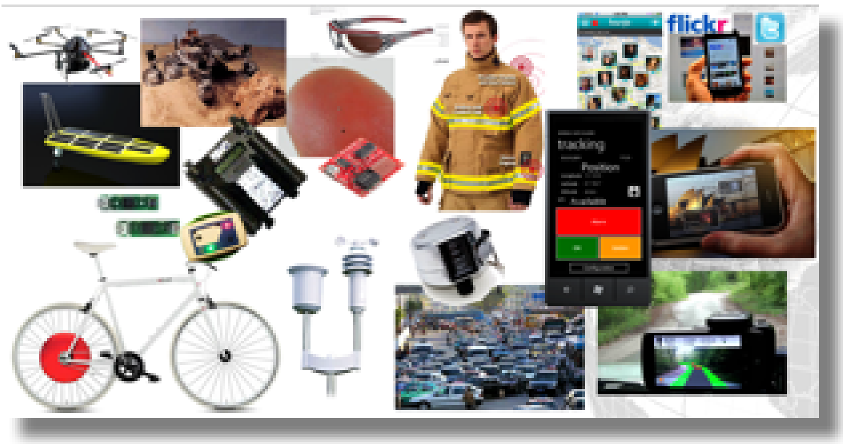

Figure 1 Live geography: Diverse sources of sensor data.

References

1 Resch, B. People as sensors and collective sensing—contextual observations complementing geo-sensor network measurements. In J. Krisp *Advances in Location-Based Services*, Springer, Berlin. pp. 391–406, 2013.

2 Resch, B., A. Zipf, P. Breuss-Schneeweis, E. Beinat, and M. Boher. Towards the live city—paving the way to real-time urbanism. *International Journal on Advances in Intelligent Systems*, 5(3&4): 470–482, 2012.

4.8 Environmental testbeds for dependable sensor networks

Kay Rômer (TU Graz, AT)

Sensor networks provide a substrate to realize applications in several domains of utmost importance for our society, including surveillance of critical infrastructures, smart cities, smart grids, and smart healthcare. However, many of these applications are only possible if sensor networks provide dependable performance. Application-specific guarantees on network performance parameters such as data delivery reliability and latency must be given for all system operation conditions. Failure to meet these requirements at all times may lead to reduced user satisfaction, increased costs, or to critical system failures. Unfortunately, existing sensor network technologies mostly follow a best-effort approach and do not offer guaranteed performance.

The major hurdle to providing dependable sensor is that their operation is deeply affected by their surrounding environment. Environmental properties such as electromagnetic (EM) radiation, ambient temperature, and humidity have significant impact on achievable network performance. Sensor network communication has to deal with interference from other communication networks such as WiFi, Bluetooth, RFID and from other systems such as microwave ovens or engine ignition systems. The resulting interference may lead to message loss, which in turn leads to an increase in latency and energy consumption. Environmental temperature also deeply affects the operation of WSAN. We have shown in previous work that temperature variations in a deployment may lead to failing transmissions during hot periods. Not only are these environmental conditions hard to predict for a given deployment site, they also may largely vary from one deployment site to another, thus hindering scalable deployment of IoT applications as every new deployment site requires costly customization. For example, climatic conditions and the use of frequency bands varies heavily across cities and countries.

In order to design sensor networks that can provide certain performance guarantees despite changing environmental conditions, there is a need for testbeds with realistic environmental effects, where protocols and applications can be run on real sensor network hardware under repeatable and realistic environmental conditions. In this talk, two such testbeds were presented—TempLab and JamLab—where user-defined temperature and interference conditions can be created. In TempLab, sensor nodes are equipped with infrared heating lamps that can be controlled via wireless dimmers to create an accurate temperature profile that varies over time and space. In JamLab, a subset of the sensor nodes in the testbed is used to record and playback interference patterns with high accuracy without the need for additional hardware.

Theme: Teaching and Curricula

4.9 Education materials: Lessons learned and opportunities

Susanne Bleisch (The University of Melbourne, AU)

GITTA (Geographic Information Technology Training Alliance, http://www.gitta.info) was initially funded by the Swiss Virtual Campus Initiative to create comprehensive and in-depth online teaching materials in geographic information science and technology. While the original eLearning hype pushed the hopes of saving money it was soon realized that this is difficult to achieve and many of the early eLearning projects disappeared after funding run out. GITTA was able to survive by opening up its content under a creative commons license and transferring from the original project consortium to a supporting association that welcomes new partners. Additionally, the project employs the eLesson Markup Language eLML (http://www.elml.ch) to achieve Learning Management System independence and output format flexibility. The XML-based language eLML supports content creation by offering the ECLASS structure. From these eLearning projects was learned that the discussion between the different project partners about learning objectives and lesson contents is very valuable. However, each teacher also has his or her own style and does not necessarily like to use predefined teaching materials. Additionally, it is often difficult to fit a series of longer learning units such as lessons into the available time slots or the lesson levels or contents do not correspond with the students' prior knowledge. For the usage of the case studies it is crucial that the employed data sets are available.

From a user survey conducted in 2010 it is known that the GITTA materials are actively used for teaching and learning. However, it is also known that quite often the materials are used as a source of inspiration or single illustrations or animations from the GITTA materials are used in lectures. Based on this knowledge it seems natural to propose that the focus of creating educational materials in the area of geosensor networks should take on the form of small and flexible learning objects or resources. Learning objects are more complete but small units that are ideally structured (for example according to the eLMLs ECLASS principle), aligned with learning objectives and described through metadata (for example the IMS Learning Resource Meta-data Specification, http://www.imsglobal.org/metadata/). Even smaller are learning resources, which may consist of single illustrations, simulations, questions, etc. To improve discoverability, learning resources should ideally also be aligned with learning objectives and described through metadata. A repository of learning objects or resources allows using them flexible in a range of settings such as face-to-face lectures, flipped classroom or potentially mobile learning. Each user has the freedom to choose and to incorporate into his or her teaching what seems to fit with fewer difficulties regarding teaching style, level, or time allocations.

4.10 Education and decentralized spatial computing

Matt Duckham (The University of Melbourne, AU)

Main reference M. Duckham, "Decentralized Spatial Computing: Foundations of Geosensor Networks," 320p., ISBN 978-3-642-30852-9, Springer, 2013.
URL http://link.springer.com/book/10.1007/978-3-642-30853-6
URL http://ambientspatial.net/book

Learning about decentralized spatial computing requires the development of specific domain skills and knowledge across a number of important areas, including computing and distributed systems; spatial information and analysis; and algorithm design and analysis. In addition to these specialized topics, decentralized spatial computing links well to several general skills for students of computer science, engineering, and geographic information science. Specifically, teaching the design and testing of decentralized spatial algorithms can support the development transferable skills including:

1. logical thinking and analysis, requiring careful and structured problem solving and system engineering, in the design of the algorithms;
2. constructive critical thinking and self-criticism, necessary in the adversarial analysis of algorithms; and
3. validating conclusions through the construction of objective and statistical experimental evidence to support the properties of an algorithm.

The information on http://ambientspatial.net/book to complement the book [1] contains many examples of algorithm and practical code to support teaching and learning.

References

1 Matt Duckham *Decentralized Spatial Computing: Foundations of Geosensor Networks*. Springer, Berlin. 2013.

5 Working Groups

5.1 Smart Materials

Stefan Dulman, Jacob Beal, Rene Doursat, Kay Romer, and Nikolaus Correll

The breakout session started by identifying examples of smart materials and listing related work examples, including:

- strong objects (tables);
- biological materials (synthetic cells; bacteria produced materials);
- intelligent fabric;
- smart facades;
- room dividers; and
- health related things, prosthetics, smart chair, etc.

Recent funding programs have showcased many more examples, such as DARPA WarriorWeb (e.g., smart suits with accelerometers and sensors; smart skin; competitors to exoskeleton; shirts with microphones, that locate where sound comes from; and applications to geriatrics, which identify when some someone is falling or losing their balance).

This led to a refinement of the definition of smart materials from the perspective of our community. In short, features of smart materials include:

- autonomous components;
- physical coupling of devices;
- morphological computation;
- a combination of the analog and digital computing;
- a focus on "programmable";
- distinct from existing approaches like body area networks by sheer numbers, amorphous characteristics, imprecision, and price.

In turn, some of the problems facing smart materials include: What can the engineering provide (properties of materials)? What can be manufactured with what cost? and What can we program (programming emergence , a focus in this workshop).

As a result of the brainstorming, the group explored a specific example of smart materials, arriving at the concept of a programmable fabric equipped with a set of sensors (stress and shear, temperature, distance, medical, audio, physiology sensors); actuators (vibration, stiffness, torque, force, heat, light, biochemical, thermal); and outputs (pose information = somatosensory computing, pose support, mecano-transduction).

Applications for this programmable fabric include:

- mechanical actuation, e.g., posture manipulation;
- monitoring of sleeping, e.g., detecting scratching and preventing it.
- posture correction;
- sport training, e.g., teaching/observing/correcting moves;
- fall prevention, e.g., detect loss of equilibrium, geriatric support;
- mechanical actuation, e.g., tactile signaling;
- haptic feedback and virtual reality;
- feeling i-materials, e.g., signal strengths for access point deployment or radiation detection.

Finally, in order to create a prototype for the programmable fabric, the following building blocks would be needed on top if the material science engineering:

- distributed operating systems (viral programming, maintenance, debugging, error detection);
- several programming layers (start from the aggregates and end up with a medic's user interface);
- a library of adaptive computations (needed to raise the level of programming abstraction, e.g., doctors should be able to program);
- modeling in simulations (connect to tools used in material science, such as abaqus);
- centralized control (centralized calibration of sensors, self-assessment, safety and certification); and
- identifying who is in control (e.g., shirt manufacturer, doctor, user).

5.2 Self-organization and geosensor networks

Rene Doursat, Stefan Dulman, Jacob Beal, Kay Romer, Zoltan Papp, Sabrina Merkel, and Nikolaus Correll

Questions posed in this group included: What makes the topic of self-organization interesting? How does self-organization fit the field of geosensor networks? Would it be possible to import ideas and tools from other domains?

Self-organization can be defined as a collection of objects that form and maintain a structure. Temporal efficiency needs to be part of the definition. Clear requirements are needed before deciding that a distributed approach is to be preferred. Typically, there exist trade-offs between functionality,spatial constraints, and resources (communication).

Real systems need teal-time predictability. Combining control systems with emergence must lead to a system collectively predictable. The decentralized features of a practical system need to "be hidden under the hood." Users are not interested in the centralized/decentralized system design approach. Instead, users care about efficiency, reliability, robustness, and modularity.

For example, wilderness monitoring is an interesting application for distributed systems (wild fires, precision agriculture, ocean/environment monitoring). A second application is field infrastructure (emergency and disaster monitoring, and the developing word). Smart materials is a third application area, bringing in the question of high density nodes. The switch from physical to wireless would be a loss of information density. These scenarios involve dynamic/volatile architectures (nodes/links come and go) but one wants to see a stable functionality and have clear guarantees.

However, it was agreed that self-organization should be used only if there is a clear incentive to do so. The effort must not outweigh the benefits.

5.3 Benchmarking and data challenge

Matt Duckham, Annalisa Appice, Edzer Pebesma, and Monika Sester

The development of benchmark data sets and an associated data challenge in the area of geosensor networks would serve two purposes:

1. Increasing comparability and reproducibility of research results, allowing many researchers to test algorithms and analysis on the same data set.
2. Helping with outreach and enlarging the community, showcasing relevant problems.

Geosensor network data is typically low-accuracy, high-resolution data. In the benchmarking and data challenge session, we identified certain key characteristics of an idea benchmark data set and associated data challenge:

- simple accessible data formats (e.g., csv files, database tables);
- raw, unprocessed sensor readings;
- real in preference to simulated data;
- data about both static and mobile sensor platforms;
- context descriptions and ground truth information (e.g., events that are known to have occurred in the region/period covered by the data).

A data challenge would then need to identify and make accessible suitable data, develop applications and questions for the challenge, and help provide a forum for collaboration and communication of results (for example via a workshop, a challenge associated with a major conference, or a Dagstuhl seminar).

One of the planned activities following on from this seminar is the development of a data challenge. For example, http://www.marinetraffic.com/ makes accessible shipping AIS data which might be collated for a geosensor network data challenge, such as:

- based on historical movement, forecast vessels in a specific region or port in the future;
- detect anomalies, such as suspicious movements or AIS spoofing.

5.4 GSN4D: Geosensor networks for development: Applications in smart grid and e-health

Hedda Schmidtke, Edzer Pebesma, Daniel Fitzner, Thomas Kirste, Sabrina Merkel, and Martijn Warnier

Geosensor networks (GSN) have a number of applications for facilitating international development programs. International development is a rewarding field of application that contributes to saving human lives and secure international stability and security. This breakout session focused on two particular application areas: smart grid and e-health. We discussed how standard application scenarios for the developed world match to scenarios for development. In the smart grid area, standard GSN application scenarios include:

- How can GSN be used to provide energy from renewable sources during peak times and control energy flow? and
- How can end-users be made aware of their energy usage, so as to incentivize desirable behavior of consumers?

In a developing world scenario, similar research questions are relevant. But here, avoiding peaks in demand is crucial to avoid power outages and damage to equipment. Inclusion of renewable energy sources in this scenario facilitates economic development and increased life quality for citizens. Many of the practical challenges that hinder innovation in energy systems in the developed world are less of a concern in developing countries. Utilities in developing countries may be more open towards integrating smart energy systems, as these systems may contribute to improved reliability of services and protection of assets. Also, end-consumers in the developing world may be more likely to wish to understand how they can have a positive influence on the operation of the grid.

There is a wide area of opportunities for GSN research in this application scenario. Communication protocols and distributed processing can be studied well with a smart grid application scenario. Sensor nodes can be attached to power infrastructure for monitoring and controlling the energy flow. Moreover, the energy grid can provide power to sensors alleviating constraints arising from battery life concerns in other geosensor network applications. Ad-hoc networks and self-organizing network structures are a focus of GSN research that is particularly interesting in a developing world scenario. But mobile communications protocols, such as GSM, are also viable alternatives for implementing communication for GSN research with a focus outside of the communications area. All research scenarios in the area of GSN for smart grid applications, such as decentralized processing of information, and GSN for monitoring, error prediction, and localization of errors carry over to a developing world scenario. Some differences exist in the economic perspective on smart grids, since consumers in the developing world are in general more willing to adopt the new technologies without economic incentives, as a more reliable grid is an incentive per se.

However, consumers in both scenarios need to be made aware of energy consumption and grid operations with easy-to-use interfaces. Another question is how smart appliances or smart plugs can be economically sensible options in a developing world scenario. Interesting options would be to integrate smart plug functionality into plugs for spike protection, or to integrate smart meters into pre-paid meters regularly installed into households.

Among the algorithms and models used in the domain are market simulation and prediction models and algorithms, e.g. multi-agent systems and game theory, as well as network models to simulate the grid structure as a spatially embedded network, and to model the energy flow. Special distributed processing simulations and algorithms are needed to test smart grid GSNs, integrating the wide range of interdisciplinary perspectives on the problem: from the perspective of the electrical engineer to the economist and social science perspective.

A similarly exciting and worthwhile application domain in the larger area of applications for international development is e-health. GSN in this domain arise as crowd sensing or human sensing networks. A large proportion of the population in developing countries owns a mobile phone. Successful applications for the case of Rwanda, e.g., are the RapidSMS implementation for maternal and child health[2] and the mUbuzima (Kinyarwanda for m-health) project for monitoring parameters for the Milennium Development Goals[2] set by the UN. RapidSMS is used for monitoring pregnancies in remote villages and also for intervention in case of complications. In case of an emergency, the system can provide advice and automatically requests an ambulance to bring the mother to the nearest hospital. Like in the case of smart grid applications, applications in e-health require an interdisciplinary approach with GSN research integrating medical and/or pharmaceutical perspectives.

[2] http://www.unmillenniumproject.org/goals/

References

1 Krogh, B.H. and H.R. Schmidtke. Electrical power in Africa: Challenges and opportunities. *IEEE Smart Grid Newsletter*, 12 2012.

2 Ngabo, F., J. Nguimfack, F. Nwaigwe, C. Mugeni, D. Muhoza, D.R. Wilson, J. Kalach, R. Gakuba, C. Karema, and A. Binagwaho. Designing and implementing an innovative SMS-based alert system (RapidSMS-MCH) to monitor pregnancy and reduce maternal and child deaths in Rwanda. *The Pan African medical journal*, 13, 2012.

5.5 Data integration and big data from sensors

Annalisa Appice, Susanne Bleisch, Allison Kealy, Thomas Kirste, Steve Liang, Hedda R. Schmidtke, and Monika Sester

Geosensor networks (GSN) produce large amounts of data. In order to make data usable a number of questions have to be addressed. Two key challenges are scalability and integration of heterogeneous data. An important goal is the discovery of hidden knowledge in data through queries, browsing, or exploration of data. Technologies to access this hidden knowledge are machine learning and logical reasoning methodologies or combinations of these. To consider an example, assume a suspicious boat has been reported: how can a GSN be used to determine what has happened? In this case information may be available from sensors deployed in the water, remote sensing, boat AIS providing ID, position, and velocity, plus material from social networking platforms, such as twitter and Flickr. Spatial and temporal indexing play a key role to limit the search space, assuming that locality implies causal relation between events.

5.6 Privacy issues for geosensor networks

Bernd Resch, Edzer Pebesma, Matt Duckham, Jörg-Rüdiger Sack, and Martijn Warnier

Geosensor networks collect data. Typically a lot of different data is collected, some of which might be privacy-sensitive because it can be linked back to individuals. For this reason engineers working in the field of geosensor networks should be aware of the privacy issues involved.

Ultimately, when designing geosensor networks, the privacy of individuals related with the network should be respected. It is not always clear who's privacy should be protected, because data can be linked back to individuals in many ways, including combining geosensor data with other networks. Several issues are of importance here:

- End users should be able to balance privacy needs with performance measures, i.e., by offering more detailed information about oneself, the offered service might perform better.
- Decentralized architectures inherently have the potential to allow more fine grained control/choice for end-users about the type of information they share and the granularity of that information. However, decentralized solutions do not guarantee more privacy by default.

- Several forms of data obfuscation can help the privacy of end users. For example, inaccurate time stamps in data might help privacy of individuals. This comes at the cost of less reliable—and probably overall less useful—data.
- Privacy awareness of end-users remains an issue, though applications such as Snapchat (http://snapchat.com) might raise the privacy awareness of users.
- Privacy transparency needs further exploration. For most applications it is not clear how they process data (and who they share it with), making most applications opaque from a privacy perspective.

We concluded that guaranteeing privacy is a complex and difficult issue, with no general solutions for the all cases, but with some possibilities for application- or context-specific solutions.

5.7 Teaching geosensor networks

Warnier, Martijn

This group discussed some of the issues related to teaching geo sensor networks. Besides the (obvious) conclusion that a book such as [1] is a nice starting point for this (especially with the integration of a programming language like NetLogo), showing students how to think "from the sensor" and stresses decentralized design, the group also identified the importance of several other issues. These include showing (and letting students play with) real hardware. Playing with hardware should be stimulated at the beginning of an (introductory) course, so that students become familiar with issues such as energy constraints and get a feeling for networks. For example, it might be possible to let students design their own routing protocols, which are then peer-reviewed by the students and then implemented in NetLogo. This stimulates them to think for themselves and let them appreciate the complexities involved. A competitive element might also stimulate students (e.g., Who can build the best sensor network?). In turn, this might require a teaching platform and/or real data.

References

1 Matt Duckham *Decentralized Spatial Computing: Foundations of Geosensor Networks*. Springer, Berlin. 2013.

6 Open Problems

Matt Duckham, Jacob Beal, René Doursat, Thomas Kirste, Steve Liang, Zoltan Papp, Edzer Pebesma, Hedda Schmidtke, and Monika Sester

Finally, an open problems session yielded the following range of open problems:

JB Can the Nyqvist Theorem be generalized to universal space-time computation?
ZP Can we guarantee that we find a solution for self-organization within a particular time constraint?

EP How do I generalize dense attributed trajectories? Cartographic generalization operates only on the spatial projection, not space-time.
TK Is it possible to define SVD (single value decomposition) for high dimensional categorical state space using a generative definition?
JB What API will make it easy for sensor network engineers to use aggregate programming?
SL How do I discover the sensors fit for my application and search results need to be fast, fresh, complete, relevant?
MS Is there a space-time search engine (i.e., a GIR engine that also accounts for time)?
RD/HS Don't we have to give up on (some of) our needs for assurance/validation/feasibility of implementation in exchange for qualitative leap in functionality/benefits strong self organization? And if yes, why aren't we doing it? This may alternatively be decomposed into two questions. What is the best trade off between certainty and predictability in engineering self-organizing algorithms? and 2. How to use a machine for brainstorming, computer aided creativity?

7 Future Visions

René Doursat

This closing session invited all participants to describe their "dreams" or long-term motivations in their research on decentralized sensor networks. What did they imagine (or would they like to see) happening in several years or decades from now? What are to them the ultimate endeavors of this cross-disciplinary field? Additionally, what are the potential dangers or abuses that could also emerge from these new technologies?

The answers were extremely diverse, illustrating the wide scope of application domains and techniques. A useful classification that emerged from this session relies on the spatial scale of the systems:

System scales (from smallest to largest)

- in the body (< 1mm): "swarms" of nanotech particles (medication delivery, targeted cures);
- on the body (1–10cm): sensor/actuator "skin" or "suit," wearable computing;
- on furniture, on a wall (1m): "touch-table," life-size videoconferencing;
- across a home (5–10m): assisted living, accident detection, ambient intelligence;
- in a building (10–100m): automated regulation of utilities, emergency evacuation;
- in a crowd: opportunistic networking via human/cellphone carriers;
- across a city (1–10km): traffic control and optimization (lights, traffic jams, detours);
- across a social network: Wiki-like mobile collaboration;
- across a region, territory (10–100km): forest fires, agriculture, ecology of fauna/flora;
- in a population: disaster preparedness, epidemic, emergency response; and
- in the ocean or atmosphere (100–1000km): currents, climate, environment, pollution.

A few of these potential applications were discussed further. Here follows a summary, which attempts to categorize them into major domains and topics (with many crossovers, naturally):

Body, health, and assisted living

- automated health assessment by navigating through the environment, for example, detecting a cold leads to medication or a recommendation to visit the pharmacy.
- smart clothes combining sensing and actuating, for example, a sore muscle or headache is detected leading to a proposed massage; or a baby's respiration or a senior's gait/balance are monitored.
- automated physical therapy, for example, pose recognition and posture manipulation; or rehabilitation, and regaining patterns of motion.
- body implants with sensors, for example "smart" hip bone prosthesis.
- enabling a better connection between senior people and their entourage, for example alerting a medical center in case of emergency or simply being able to say hello to family displayed on the wall.

Human crowds, social networks, populations

The main idea in this category is to create an "opportunistic" peer-to-peer network that relies on the human carriers, essentially via their PDAs: cell phones, smart watches, or some other specific device (such as a bracelet given out at a concert).

- *entertainment*: a crowd at an event (sports, concert, demonstration), for example, PDAs emit colored light signals—seen from above, the whole crowd creates waves, patterns or entire videos on a giant "amorphous" display.
- *security*: law enforcement, emergency response, for example, counting how many people there are or guiding the crowd to an evacuation route, via PDAs showing them the way.
- *information, knowledge*: Wiki-like collaboration MANETS (mobile ad-hoc networks), for example, the empowerment of anyone to innovate and contribute to project (in a "prosumer" fashion), such as "mobile journalism."
- *disaster management*: information to support responses to earthquakes, floods, nuclear accidents, epidemics, etc.

The main challenges here include the needs for interoperability between people's contributions; and dealing with privacy problems (and/or adapting to changes in expectations and approaches to privacy).

Smart materials, mechanical/civil engineering, aeronautics

Many of the visions centered around new materials and smart paints or coatings. A myriad of immobile tiny sensor-actuator particles, densely strewn on a surface (stuck in place), collectively performing "amorphous computing" formed the infrastructure for these visions, including:

- at home, for example a sensitive carpet, touch-walls, touch-tables, etc.
- engineering structures, such as self-monitoring bridges and aircraft fuselages.
- beyond assessing, actuating and optimizing a structure, for example modifying an aircraft wings' profile during flight ("smart/soft aeronautics").
- a myriad of mobile tiny sensor/actuator particles ("smart dust"), which either passively glide along with the winds or currents, or are self-propelled and can steer themselves, for example in environmental applications (forests, rivers, crops, etc.).

For the latter applications in particular, there was the potential risk of unwanted "nano-pollution." Creating a degradable paint or dust sensor particles might avoid this accumulating pollution problem. Degradation might happen spontaneously, or upon a global termination signal, using abiotic (organic chemistry, biomimicry) or biotic (leveraging the self-organizing power of bacteria, synthetic biology) means.

Theoretical and technical challenges

Altogether, what is really needed is not just smart "sensing," but a whole sensing-computing-actuating cycle in each element.

A major question raised was whether there was any advantage in using a decentralized algorithm as opposed to a centralized algorithm. The question was posed: "Which applications warrant decentralization?" Conversely, which applications can be implemented in a classical way, i.e., by a computing method that has complete knowledge of all the data available in the system?

- *Answer 1*: applications where space plays an important role in restricting the flow of information, so that there are spatial constraints to the movement of information.
- *Answer 2*: where there is a need to optimize resources (primarily energy), even if their could technically be a central computing device, i.e., we want to do more with less resources.

It was also argued that, although important, this question actually should not concern the programmer. Decentralization, if any, should be "transparent" or "under the hood." Ideally, there would be a "compiler" able to transform any central algorithm into a set of rules distributed over a myriad of agents. Going even further, such systems should allow anyone, not just skilled programmers, to innovate by using and transforming the sensing information. Users should not only be consumers, but also "natural" programmers who could instantly create new applications.

Next, was the lower-level issue of how decentralized algorithms could be implemented. How do elements collaborate in a peer-to-peer fashion and make decisions collectively? Typically, spatial structures, such as gradients of "hop counters," are an important fundamental mechanism of self-organization.

Finally, a methodological and epistemological question was raised. With all these future network computing systems, is it not the case that traditional engineering requirements for assurance of results, validation/proof/checks, and feasibility of implementation may need to be relaxed in favor of a step-change in functionality provided by genuine self-organization? In other words, it may be that "organic"-like systems, where function is the emergence of a great collective of small elements, will never be fully predictable or controllable. Then, it is not a matter of guaranteeing a service, but rather setting an acceptable level of probability of success. We must also be prepared for that.

Participants

- Annalisa Appice
University of Bari, IT
- Jacob Beal
BBN Tech. – Cambridge, US
- Susanne Bleisch
The University of Melbourne, AU
- Nikolaus Correll
Univ. of Colorado – Boulder, US
- René Doursat
CNRS – Paris, FR
- Matt Duckham
The University of Melbourne, AU
- Stefan Dulman
TU Delft, NL
- Daniel Fitzner
Leibniz Univ. Hannover, DE
- Allison Kealy
The University of Melbourne, AU
- Thomas Kirste
Universität Rostock, DE
- Steve Liang
University of Calgary, CA
- Marco Mamei
University of Modena, IT
- Sabrina Merkel
KIT – Karlsruhe Institute of Technology, DE
- Zoltan Papp
TNO – The Hague, NL
- Edzer Pebesma
Universität Münster, DE
- Bernd Resch
Universität Heidelberg, DE
- Kay Römer
TU Graz, AT
- Jörg-Rüdiger Sack
Carleton Univ. – Ottawa, CA
- Hedda R. Schmidtke
Carnegie Mellon University – Moffet Field, US
- Monika Sester
Leibniz Univ. Hannover, DE
- Martijn Warnier
TU Delft, NL

Report from Dagstuhl Seminar 13502

Approaches and Applications of Inductive Programming

Edited by
Sumit Gulwani[1], Emanuel Kitzelmann[2], and Ute Schmid[3]

1 **Microsoft – Redmond, US,** sumitg@microsoft.com
2 **Universität Duisburg – Essen, DE,** ekitzelmann@gmail.com
3 **Universität Bamberg, DE,** ute.schmid@uni-bamberg.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13502 "Approaches and Applications of Inductive Programming". After a short introduction to inductive programming research, an overview of the talks and the outcomes of discussion groups is given.

Seminar 8.–11. December, 2013 – www.dagstuhl.de/13502
1998 ACM Subject Classification I.2.2 Automatic Programming, Program Synthesis
Keywords and phrases inductive program synthesis, end-user programming, universal artificial intelligence, constraint programming, probabilistic programming, cognitive modeling
Digital Object Identifier 10.4230/DagRep.3.12.43
Edited in cooperation with Umair Z. Ahmed

1 Executive Summary

Sumit Gulwani
Emanuel Kitzelmann
Ute Schmid

Inductive programming (IP) research addresses the problem of learning programs from incomplete specifications, such as input/output examples, traces, or constraints. In general, program synthesis is a topic of interest to researchers in artificial intelligence as well as in programming research since the 1960s [2]. On the one hand, this research aims at relieving programmers from the tedious task of explicit coding on the other hand it helps to uncover the complex cognitive processes involved in programming as a special domain of complex problem solving. From the beginning, there were two main directions of research – deductive knowledge based approaches and inductive machine learning based approaches. Due to the progress in machine learning, over the last decades the inductive approach currently seems to be the more promising.

Researchers working on the topic of IP are distributed over different communities, especially inductive logic programming (ILP) [12, 6], evolutionary programming [13], functional programming [15, 5, 10], grammar inference [1], and programming languages and verification [7]. Furthermore, domain specific IP techniques are developed for end-user programming [4, 9] and in the context of intelligent tutoring in the domain of programming [8]. In cognitive science, researchers concerned with general principles of human inductive reasoning have constructed computer models for inductive generalization which also have some relation to IP [3, 16].

In general, approaches can be classified by (1) the strategy of program construction which can be example-driven or generate-and-test driven; (2) the implicit or explicit restriction bias which can be Horn clauses, functional programs, or domain specific languages possibly with further constraints given as meta-interpreters, templates or program schemes; (3) the possibility to consider background knowledge.

IP research had its first boost in the 1970s in the context of learning Lisp programs from examples. Due to only limited progress, this direction of research decayed and in the 1990s was newly addressed in the context of ILP and evolutionary programming. Again, after first promising results, disappointment set in [14, 11]. However, over the last years, a new revival in IP research can be observed in different communities and promising results, for example in the domain of enduser programming, give rise to new expectations.

Therefore, in the Dagstuhl Seminar AAIP we brought together researchers from these different communities as well as researchers of related fields. The possibility to discuss and evaluate approaches from different perspectives helped to (a) gaining better insights in general mechanisms underlying inductive programming algorithms, (b) identifying commonalities between induction algorithms and empirical knowledge about cognitive characteristics of the induction of complex rules, and (c) open up new areas for applications for inductive programming in enduser programming, support tools for example driven programming, and architectures for cognitive systems.

The presentations covered several aspects of inductive programming and were grouped in the topic sessions

- Inductive Programming Systems and Algorithms (with an introductory talk by Stephen Muggleton),
- Enduser Programming (with an introductory talk by Sumit Gulwani),
- Intelligent Tutoring and Grading,
- Cognitive Aspects of Induction (with an introductory talk by José Hernández-Orallo),
- Combining Inductive Programming with Declarative Programming and with Other Approaches to Program Synthesis (with an introductory talk by Luc de Raedt).

In an initial discussion round three focus topics were identified and further discussed in working groups

- Comparing Inductive Logic and Inductive Functional Programming as well as other Approaches to Program Synthesis,
- Potential New Areas of Applications and Challenges for Inductive Programming,
- Benchmarks and Metrics.

Concluding Remarks and Future Plans

In the final panel discussion the results of the seminar as well as future plans were identified. Participants stated that they learned a lot about different inductive programming techniques and tools to try. The general opinion was that it was very inspiring to have researchers from different backgrounds. To facilitate mutual understanding it was proposed to give introductory lectures, define the vocabulary of the different groups, collect a reading list, and identify common benchmark problems.

To progress in establishing inductive programming as a specific area of research it was proposed to write a Wikipedia page, and to collect introductory literature from the different areas covered in the seminar. Furthermore, plans for joint publications and joint grant proposals were made.

This seminar was highly productive and everybody hoped that there will be a follow-up in the near future.

References

1 D. Angluin and C. H. Smith. Inductive inference: theory and methods. *Computing Surveys*, 15(3):237–269, September 1983.

2 A. W. Biermann, G. Guiho, and Y. Kodratoff, editors. *Automatic Program Construction Techniques*. Macmillan, New York, 1984.

3 P. A. Carpenter, M. A. Just, and P. Shell. What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices test. *Psychological Review*, 97:404–431, 1990.

4 A. Cypher. EAGER: Programming repetitive tasks by example. In *Human Factors in Computing Systems, Proc. of CHI'91*, pages 33–39. ACM Press, 1991.

5 C. Ferri-Ramírez, José Hernández-Orallo, and M. José Ramírez-Quintana. Incremental learning of functional logic programs. In *FLOPS '01: Proceedings of the 5th International Symposium on Functional and Logic Programming*, pages 233–247, London, UK, 2001. Springer-Verlag.

6 P. Flener and S. Yilmaz. Inductive synthesis of recursive logic programs: Achievements and prospects. *Journal of Logic Programming*, 41(2–3):141–195, 1999.

7 Sumit Gulwani. Synthesis from examples: Interaction models and algorithms. In *SYNASC*, pages 8–14, 2012.

8 Sumit Gulwani. Example-based learning in computer-aided stem education. *To appear in Commun. ACM*, 2014.

9 Sumit Gulwani, William R. Harris, and Rishabh Singh. Spreadsheet data manipulation using examples. *Commun. ACM*, 55(8):97–105, 2012.

10 E. Kitzelmann and U. Schmid. Inductive synthesis of functional programs: An explanation based generalization approach. *Journal of Machine Learning Research*, 7(Feb):429–454, 2006.

11 Tessa Lau. Why programming-by-demonstration systems fail: Lessons learned for usable ai. *AI Magazine*, 30(4):65–67, 2009.

12 S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming, Special Issue on 10 Years of Logic Programming*, 19-20:629–679, 1994.

13 Roland Olsson. Inductive functional programming using incremental program transformation. *Artificial Intelligence*, 74(1):55–83, March 1995.

14 J.R. Quinlan and R.M. Cameron-Jones. FOIL: A midterm report. In P. B. Brazdil, editor, *Proceedings of the ECML'93*, number 667 in LNCS, pages 3–20. Springer, 1993.

15 U. Schmid and F. Wysotzki. Induction of recursive program schemes. In *Proc. 10th European Conference on Machine Learning (ECML-98)*, volume 1398 of *LNAI*, pages 214–225. Springer, 1998.

16 Ute Schmid and Emanuel Kitzelmann. Inductive rule learning on the knowledge level. *Cognitive Systems Research*, 12(3):237–248, 2011.

2 Table of Contents

3 Overview of Talks on: Inductive Programming Systems and Algorithms

3.1 Predicate invention and learning of recursive logic programs

Stephen Muggleton (Imperial College London, UK)

Joint work of Lin, Dianhuan; Pahlavi, Niels; Tamaddoni-Nezhad, Alireza
URL http://ilp.doc.ic.ac.uk/metagolD/

Inductive Logic Programming (ILP) is the sub-area of Machine Learning concerned with inductive inference of logic programs. Since logic programs can be used to encode arbitrary computer programs, ILP is a highly flexible form of Machine Learning, which has allowed it to be successfully applied in a number of complex areas. However, despite this fact, state-of-the-art ILP systems such as FOIL, Golem and Progol are unable to effectively learn representations such as regular and context-free grammars from example sequences. Such tasks require the automated introduction of new recursively defined non-terminals into the description language. Within ILP this is referred to as predicate invention. Early ILP systems achieved limited forms of predicate invention. However, this approach was abandoned since it was unclear how to bound the complexity of the search involved. A recent review of ILP emphasised an urgent need for renewed attention to Predicate Invention in order to broaden the applications of ILP. Recent work has substantially generalised this idea to support learning of regular and context-free grammars as well as higher-order dyadic datalog programs. The key innovation was the introduction of a Prolog meta-interpreter driven by a set of higher-order Datalog definite clauses. Instantiation of the predicate symbols of these higher-order clauses is achieved using a form of ground abduction. The Meta-interpreter is used to prove the set of examples. If any sub-proof fails then a rule is formed by abduction, allowing the proof to be completed. A set of abduced rules which allows all the positive examples and none of the negative examples to be proved comprises a valid hypothesis. Completeness of the inductive process is ensured by the use of a complete form of ground abduction. The proof space of the meta-interpreter is several orders of magnitude more efficient than the refinement-graph search of the state-of-the-art ILP systems.

3.2 Cumulative Learning by Refinement and Automated Abstraction

Robert J. Henderson (Imperial College London, GB)

Main reference R. J. Henderson, "Cumulative Learning in the Lambda Calculus," PhD thesis, Imperial College London, 2014.

I describe a new approach to 'cumulative learning', in which an inductive inference system automatically acquires knowledge necessary for solving harder problems through experience of solving easier ones. I have implemented this approach in an inductive functional programming system called RUFINSKY, which employs an alternating two-phase policy in order to solve a sequence of related but successively more difficult learning problems. In the first phase, a hypothesis is learned to fit training data using a guided search technique called refinement which has been adapted from inductive logic programming. In the second phase, new elements of background knowledge are abstracted from syntactic patterns in hypotheses by anti-unification of subterms and a form of inverse beta-reduction. The new background knowledge has the effect of shifting RUFINSKY's inductive bias, allowing it to automatically adapt to a problem domain. Cumulative learning techniques are promising as a route towards practical AIXI-style Artificial General Intelligence (AGI). Taking into account new literature on the potential existential risks to humanity posed by AGI, I discuss whether research into cumulative learning should or should not be considered ethical.

3.3 Example-driven Inductive Functional Programming with IGOR2

Emanuel Kitzelmann (Universität Duisburg – Essen, DE)

Main reference E. Kitzelmann, "A Combined Analytical and Search-based Approach for the Inductive Synthesis of Functional Programs," KI – Künstliche Intelligenz, 25(2):179–182, 2011; see also corresponding PhD Thesis, Faculty of Information Systems and Applied Computer Sciences, University of Bamberg, 2010.
URL http://dx.doi.org/10.1007/s13218-010-0071-x
URL http://opus4.kobv.de/opus4-bamberg/frontdoor/index/index/docId/250

IGOR2, the inductive functional programming (IFP) system presented in this talk, learns recursive functional programs from incomplete specifications such as input-output examples or computation traces. The main contribution of IGOR2 is a combination of techniques from generate-and-test systems with techniques from analytical approaches. The goal is to gain a better trade-off between expressivity of the hypothesis language and efficiency of the synthesis procedure. Therefore, IGOR2 searches in program spaces but computes candidate programs directly from the specification. IGOR2 has been successfully applied in different domains such as list processing and strategy learning.

3.4 gErl: an Inductive Programming System with user-defined operators

Fernando Martínez-Plumed (Polytechnic University of Valencia, ES)

Joint work of Martínez-Plumed, Fernando; Ferri, Cesar; Hernández-Orallo, José; Ramirez-Quintana, Maria-José
Main reference F. Martínez-Plumed, C. Ferri, J. Hernández-Orallo, M.-J. Ramérez-Quintana, "On the definition of a general learning system with user-defined operators," arXiv:1311.4235v1 [cs.LG] , 2013
URL http://arxiv.org/abs/1311.4235v1

GErl was born as an advocacy of a more general framework for machine learning: a general rule-based learning system where operators can be defined and customised according to the problem, data representation and the way the information should be navigated. Since changing operators affect how the search space needs to be explored, heuristics are learnt as a result of a decision process based on reinforcement learning where each action is defined as a choice of operator and rule guided by an optimality criteria (based on coverage and simplicity) which feed a rewarding module. States and actions are abstracted as tuples of features in a Q matrix from which a supervised model is learnt. Erlang is used to represent theories and examples in an understandable way: examples as equations, patterns as rules, models as sets of rules, and, to defining operators: gErl provides some meta-level facilities called meta-operators which allow the user to define well-known generalisation and specialisation operators in Machine Learning.

3.5 MagicHaskeller on the Web: Inductive Functional Programming System for Casual Programming

Susumu Katayama (University of Miyazaki, JP)

MagicHaskeller on the Web is a web-based tool for inductive functional programming in Haskell. Its main focus is on offhandedness, and its users can use it in a similar way to using a web search engine. In this talk, I first demonstrated its usage. Then, I explained its implementation by shared memoization table. I concluded my presentation by discussing possible future work, including application of the algorithm to other languages.

3.6 Project of IDRE I&D oriented to implement arbitrary DSLs by learning and develop applications by demonstrating examples of their execution represented on implemented DSL

Alexey Grigoryev (National Nuclear Research University MEPhI, RU)

Joint work of Grigoryev, Alexey; Sergievsky, George M.
Main reference G M. Sergievsky, "Concept of Inductive Programming Supporting Anthropomorphic Information Technology," Journal of Computer and Systems Sciences International, 50(1):38–50, 2011.
URL http://dx.doi.org/10.1134/S1064230711010163

Inductive synthesis usually consists of two stages: 1) front-end transformation which builds a non-recursive program using the implementation language. This programs executes calculations specified by examples (usually is done by a program chosen by ad hoc developer). 2) Back-end transformation which properly implements inductive synthesis and has is a recursive program as the result of it. The main property of this program is that its unfolding-transformations allow to get a representation containing a non-recursive program (as stated above). In IDRE I&D a front-end transformation is implemented by a program which is synthesized by learning. It brings a possibility to represent instances on arbitrary DSLs specified by user in learning mode. Back-end transformation supports in increment mode, controlling structure predicate synthesis and implementation language based on non-deterministic model of calculations. This brings possibility to develop arbitrary applications by demonstrating examples of their execution represented on implemented DSLs.

4 Overview of Talks on: Enduser Programming

4.1 Flash Fill: An Excel 2013 feature

Sumit Gulwani (Microsoft Research – Redmond, US)

Joint work of Gulwani, Sumit; Singh, Rishabh; Zorn, Ben
Main reference S. Gulwani, W. Harris, R. Singh, "Spreadsheet data manipulation using examples," Comm. of the ACM, 55(8):97–105, 2012.
URL http://dx.doi.org/10.1145/2240236.2240260
URL http://research.microsoft.com/en-us/um/people/sumitg/flashfill.html

Flash Fill is a new feature in Excel 2013 for automating string transformations by examples. In this talk, I will demo Flash Fill and talk about the underlying algorithms. I will also describe some extensions to Flash Fill that enable semantic string transformations, number transformations, and table layout transformations.

References

1 Sumit Gulwani. Automating string processing in spreadsheets using input-output examples. In *POPL*, 2011.
2 Sumit Gulwani, William Harris, and Rishabh Singh. Spreadsheet data manipulation using examples. *Communications of the ACM*, Aug 2012.

4.2 Empowering Users with Data

Benjamin Zorn (Microsoft Research – Redmond, US)

Spreadsheets are valuable because they allow the free-form aggregation of data from arbitrary sources. Users can then manipulate that data using built-in tools, including filtering, aggregation, and visualization. Unfortunately the free- form nature of spreadsheets often prevents relational queries from being applied to the data, limiting its value. We propose a new language, Flare, and a new synthesis algorithm, FlashRelate, that allow users to extract relational data from semi-structured data using examples. Flare programs combine constraints on the contents of cells expressed as traditional regular expressions with spatial constraints that express how cells are related on the grid. Just as regular expressions result in a sequence of matching strings, Flare programs produce a set of tuples containing the contents of cells matching the specified constraints. Our FlashRelate synthesis algorithm can automatically generate Flare programs from a few positive and negative examples of tuples in the desired output table. FlashRelate searches over a space of possible Flare programs and iteratively builds up a solution tree, selecting the next highest ranked constraint (similar to Kruskal's spanning tree algorithm). If the resulting tree does not satisfy the negative examples, the algorithm backtracks. We describe the algorithm and evaluate its effectiveness in extracting structured data from real semi-structured spreadsheets.

4.3 Test-Driven Synthesis

Daniel Perelman (University of Washington – Seattle, US)

Joint work of Perelman, Daniel; Gulwani, Sumit; Grossman, Dan; Provost, Peter

Programming-by-example technologies allow an end-user to create simple programs to automate their workflows merely by providing input/output examples, but existing systems are specific to the domain-specific language (DSL) they were designed for. We present a new programming-by-example language LaSy which is parameterized by an arbitrary DSL that may contain conditionals and loops and therefore able to synthesize programs in any domain. In LaSy, the user provides a sequence of increasingly sophisticated input/output examples along with an expert-written DSL definition. LaSy is powered by our novel test-driven synthesis methodology which performs program synthesis iteratively, consuming a sequence of input/output examples one at a time, and DSL-based synthesis which efficiently synthesizes programs within a given DSL. We present applications of our synthesis methodology to end-user programming with LaSy programs for transformations over strings, XML, and table layouts. We compare our synthesizer on these applications to state-of-the-art DSL-specific synthesizers as well to the general purpose synthesizer Sketch.

5 Overview of Talks on: Intelligent Tutoring and Grading

5.1 Autograder: Automated Feedback Generation for Programming Problems

Rishabh Singh (MIT – Cambridge, US)

Joint work of Singh, Rishabh; Gulwani, Sumit; Solar-Lezama, Armando
Main reference R. Singh, S. Gulwani, A. Solar-Lezama, "Automated Feedback Generation for Introductory Programming Assignments," in SIGPLAN Notices, 48(6):15–26, 2013.
URL http://dx.doi.org/10.1145/2491956.2462195

In this talk, I will present the Autograder tool for providing automated feedback for introductory programming problems. Using a reference implementation and an error model, the tool automatically derives minimal corrections to student's incorrect solutions, providing them with a measure of exactly how incorrect a given solution was, as well as feedback about what they did wrong. Our results on thousands of student attempts from edX 6.00x course show that relatively simple error models can correct a significant fraction of all incorrect submissions. Towards the end of the talk, I will also present some of the recent work and many interesting open research challenges.

5.2 Automated Grading for DFA Constructions

Dileep Kini (University of Illinois – Urbana Champaign, US)

Joint work of Alur, Rajeev; D'Antoni, Loris; Gulwani, Sumit; Kini, Dileep; Viswanathan, Mahesh
Main reference R. Alur, L. D'Antoni, S. Gulwani, D. Kini, M. Viswanathan, "Automated Grading of DFA Constructions," in Proc. of the 23rd Int'l Joint Conf. on Artificial Intelligence (IJCAI'13), pp. 1976–1982, AAAI, 2013; available as pre-print from the author's webpage.
URL http://ijcai.org/papers13/Papers/IJCAI13-292.pdf
URL http://www.cis.upenn.edu/~alur/Ijcai13.pdf

One challenge in making online education more effective is to develop automatic grading software that can provide meaningful feedback. This paper provides a solution to automatic grading of the standard computation-theory problem that asks a student to construct a deterministic finite automaton (DFA) from the given description of its language. We focus on how to assign partial grades for incorrect answers. Each student's answer is compared to the correct DFA using a hybrid of three techniques devised to capture different classes of errors. First, in an attempt to catch syntactic mistakes, we compute the edit distance between the two DFA descriptions. Second, we consider the entropy of the symmetric difference of the languages of the two DFAs, and compute a score that estimates the fraction of the number of strings on which the student answer is wrong. Our third technique is aimed at capturing mistakes in reading of the problem description. For this purpose, we consider a description language MOSEL, which adds syntactic sugar to the classical Monadic Second Order Logic, and allows defining regular languages in a concise and natural way. We provide algorithms, along with optimizations, for transforming MOSEL descriptions into DFAs and vice- versa. These allow us to compute the syntactic edit distance of the incorrect answer from the correct one in terms of their logical representations. We report an experimental study that evaluates hundreds of answers submitted by (real) students by comparing grades/feedback computed by our tool with human graders. Our conclusion is that the tool is able to assign partial grades in a meaningful way, and should be preferred over the human graders for both scalability and consistency.

5.3 Automatically Generating Problems and Solutions for Natural Deduction

Umair Zafrulla Ahmed (Indian Institute of Technology – Kanpur, IN)

Joint work of Ahmed, Umair Zafrulla; Gulwani, Sumit; Karkare, Amey
Main reference U. Z. Ahmed, S. Gulwani, A. Karkare, "Automatically generating problems and solutions for natural deduction," in Proc. of the 23rd Int'l Joint Conf. on Artificial Intelligence (IJCAI'13), pp. 1968–1975, AAAI, 2013.
URL http://ijcai.org/papers13/Papers/IJCAI13-291.pdf

In this talk, I will present our recent work on automatically generating problems and solutions for Natural Deduction proofs. Natural deduction, which is a method for establishing validity of propositional type arguments, helps develop important reasoning skills and is thus a key ingredient in a course on introductory logic. We present two core components, namely solution generation and practice problem generation, for enabling computer-aided education for this important subject domain. The key enabling technology is use of an offline-computed data-structure called Universal Proof Graph (UPG) that encodes all possible applications of inference rules over all small propositions abstracted using their bitvector-based truth-table representation. This allows an efficient forward search for solution generation. More interestingly, this allows generating fresh practice problems that have given solution characteristics by performing a backward search in UPG. We obtained around 300 natural deduction problems from various textbooks. Our solution generation procedure can solve many more problems than the traditional forward- chaining based procedure, while our problem generation procedure can efficiently generate several variants with desired characteristics.

5.4 Personalized Mathematical Word Problem Generation

Oleksandr Polozov (University of Washington – Seattle, US)

Word problems are an established technique for teaching mathematical mathematical modeling skills in elementary and middle school education. However, the effectiveness of word problems widely varies among students. A large fraction of students finds word problems unconnected to real life, artificial, and uninteresting. Most students find them much more difficult than the corresponding symbolic representations. To account for these opinions, an ideal textbook should consist of a individually crafted progression of unique word problems, that form a personalized plot.

We propose a novel technique for automatic generation of personalized word problems. In our system, word problems are generated procedurally using answer-set programming from general specification. The specification includes tutor requirements (mathematical features, problem class) and student requirements (personalization, plot characters, setting). Our system generates a narrative plot, a mathematical representation, and a natural language description according to the provided specification. It makes use of a rich language of plot elements that can be parametrized by a narrative setting (fantasy world, science fiction, etc.). We are currently investigating the connection of our plot language with FrameNet, the database of semantic knowledge elements.

6 Overview of Talks on: Cognitive Aspects of Induction

6.1 Small but deep. What can we learn from inductive programming?

José Hernández-Orallo (Polytechnic University of Valencia, ES)

Main reference J. Hernández-Orallo, "Deep Knowledge: Inductive Programming as an Answer," Manuscript, 2013.
URL http://users.dsic.upv.es/~flip/papers/deepknowledge2013.pdf

Inductive programming has focussed on problems where data are not necessarily big, but representation and patterns may be deep (including recursion and complex structures). In this context, we will discuss what really makes some problems hard and whether this difficulty is related to what humans consider hard. We will highlight the relevance of background knowledge in this difficulty and how this has influence on a preference of inferring small hypotheses that are added incrementally. When dealing with the techniques to acquire, maintain, revise and use this knowledge, we argue that symbolic approaches (featuring powerful construction, abstraction and/or higher-order features) have several advantages over non-symbolic approaches, especially when knowledge becomes complex. Also, inductive programming hypotheses (in contrast to many other machine learning paradigms) are usually related to the solutions that humans would find for the same problem, as the constructs that are given as background knowledge are explicit and shared by users and the inductive programming system. This makes inductive programming a very appropriate paradigm for addressing and better understanding many challenging problems humans can solve but machines are still struggling with. Some important issues for the discussion will be the relevance of pattern intelligibility, and the concept of scalability in terms of incrementality, learning to learn, constructive induction, bias, etc.

6.2 A Cognitive Model Approach to Solve IQ-Test Problems

Marco Ragni (Universität Freiburg))

Current computational approaches outperform humans for most reasoning problem classes. Nevertheless, humans do perform better for some specific problem classes, for instance when only imprecise information is available or "insight" is a necessary precondition. Typically, in such domains not all information is given, therefore, functions and operators must often first be identified. By imitating human approaches it is possible to develop artificial intelligence (AI) systems that can deal with problems in such domains (e.g., IQ-test problems). In my talk I will elaborate on two specific domains: number series and IQ-tests. Results and limitations of the approaches are discussed.

6.3 Applying IGOR to Cognitive Problems

Ute Schmid (Universität Bamberg, DE)

Joint work of Schmid, Ute; Kitzelmann, Emanuel
Main reference U. Schmid, E. Kitzelmann, "Inductive rule learning on the knowledge level," Cognitive Systems Research, 12(3–4):237–248, 2011.
URL http://dx.doi.org/10.1016/j.cogsys.2010.12.002

IGOR is an inductive programming system based on an analytical, example-driven technique for generalization. IGOR learns recursive functional programs (in Maude or Haskell) from a small set of positive examples. Main features of IGOR are the possibility to rely on background knowledge, automated invention of sub-functions, guaranteed extensional corrections wrt. the input/output examples and guaranteed termination of the induced programs. Main restrictions of IGOR are that the given input/output examples need to be correct and to cover the first k instances over the input data type. While induction typically is very fast (due to the analytical approach), IGOR can run into memory problems when background knowledge is provided due to its generalization strategy.

Aside from typical applications in the context of automated program construction, we investigated how IGOR can learn generalized rules in cognitive domains. Taking the cognitive perspective, IGOR addresses the acquisition of constructive rules. For example, it can generalize a solution strategy for the Tower of Hanoi from an example solution trace for an Hanoi problem with three discs. Further applications we investigated are from typical planning domains such as blocksworld, learning relations such as the transitivity of “isa”, simple natural language grammars, and – most recently – rules to continue number series. While application of IGOR to examples from such cognitive domains is straight-forward, there is no general mechanism to provide the necessary examples in the necessary representation. That is, we need to explore how IGOR can be embedded in a system which generates suitable example experience.

6.4 Learning Analogies

Tarek R. Besold (Universität Osnabrück, DE)

Joint work of Besold, Tarek R.; Jäkel, Frank

Analogical reasoning is a core capacity of human cognition. In an analogy superficially dissimilar domains of knowledge are regarded as similar with respect to their relational structure. Discovering this structure allows us to transfer knowledge from one domain to another.

Over the last decades, a significant body of research in cognitive AI and cognitive science studied how this capacity can be modeled in computational terms. Most systems are focusing on the process of analogical mapping between given domain theories, i.e., on establishing correspondences between elements of the source and the target domain of the analogy, and on the following transfer of knowledge from source to target.

We want to combine the abstract mechanism of analogy-making with learning capacities from IP/ILP. Our aim is to develop a two-level model for cross-domain analogies. Starting out from independent sets of observations from several domains, we want to learn the most likely governing base theory within each domain by means of IP/ILP. In parallel, we are also trying to establish an overall cross-domain general theory encompassing the abstract structure underlying the learned base theories. If a general theory, i.e, an analogy between domains can be established, this theory, in turn, can inform the base theories in each domain.

6.5 Towards Quantifying Program Complexity and Comprehension

Mike Hansen (Indiana University – Bloomington, US)

Joint work of Hansen, Mike; Lumsdaine, Andrew; Goldstone, Robert
Main reference M. Hansen, R. L. Goldstone, A. Lumsdaine, “What Makes Code Hard to Understand?” arXiv:1304.5257v2 [cs.SE], 2013.
URL http://arxiv.org/abs/1304.5257v2

Psychologists have studied the behavioral aspects of programming for at least 40 years. In the last few decades, multiple qualitative cognitive models of program comprehension have been proposed. These models describe important aspects of a programmer’s knowledge, and provide a framework for discussing the dimensions along which a program’s cognitive complexity may vary. In this talk, we outline work being done on a quantitative cognitive model that seeks to operationalize the shared aspects of existing qualitative models. Taking inspiration from the ACT-R cognitive architecture, our model will include active vision, behavioral, and declarative/spatial memory components. In the context of inductive programming, we propose using this model for the automated culling of generated programs. Less complex programs – i.e., those that minimize some measure of resource expenditure in the model – may be preferred by human operators.

7 Overview of Talks on: Combining Inductive Programming with Declarative Programming and with Other Approaches to Program Synthesis

7.1 Towards declarative languages for learning

Luc De Raedt (KU Leuven, BE)

One of the long standing goals of artificial intelligence and machine learning is to develop machines that can be programmed automatically. To realize this dream, researchers have investigated programming and modeling languages that support machine learning. These languages provide primitives for specifying the machine learning task of interest, that is, the training instances and constraints on the programs to be learned, and the system should automatically solve the learning task. This is an effective way to realize inductive programming.

In this context, I shall report on three different languages for learning: 1) the probabilistic (logic) programming language ProbLog [1, 2], which extends Prolog with probabilistic facts and which supports parameter estimation, 2) the modeling language MiningZinc [3], which extends the constraint programming language MiniZinc with primitives for data mining, and which allows to declaratively model (and solve) a wide variety of pattern mining problems, and 3) kLog [4], the logical and relational language for kernel-based learning, which allows users to specify logical and relational learning problems at a high level in a declarative way. I shall also discuss how these languages can be used for automatic and inductive programming.

References

1 http://dtai.cs.kuleuven.be/problog/index.html
2 D. Fierens, G. Van den Broeck, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt. Inference and learning in probabilistic logic programs using weighted Boolean formulas. arXiv preprint arXiv:1304.6810 (to appear in Theory and Practice of Logic Programming), 2013. http://arxiv.org/abs/1304.6810
3 T. Guns, A. Dries, G. Tack, S. Nijssen, and L. De Raedt. MiningZinc: A Modeling Language for Constraint-based Mining, In Proceedings of IJCAI, 2013. https://lirias.kuleuven.be/bitstream/123456789/399282/1/miningzinc_ijcai_final.pdf
4 P. Frasconi, F. Costa, L. De Raedt, K. De Grave. kLog: A Language for Logical and Relational Learning with Kernels. arXiv:1205.3981, 2012. http://arxiv.org/abs/1205.3981

7.2 Probabilistic programming and automatic programming

Iurii Perov (University of Oxford, GB)

Joint work of Perov, Iurii; Wood, Frank; Mansinghka, Vikash; Kulkarni, Tejas, Kulkarni; Tenenbaum, Joshua

Probabilistic programming has recently attracted much attention in Machine Learning and Computer Science communities. Probabilistic programming (PP) took from "traditional" programming its convenient way to define generative models as "algorithms", which in the case of PP generally contain much uncertainty. In addition to evaluation (program execution) component, probabilistic programming involves inference, so that models can converge from its prior to its posterior (given observations).

That is, in traditional programming you usually define inputs, define a precise algorithm, and computer gives you outputs. In probabilistic programming you define inputs, specify outputs (i.e. observations) or at least part of them (i.e. a train set), provide a prior model with uncertainty, and computer is supposed to get you into posterior via general- and special-purpose inference techniques (e.g. by providing you with approximate posterior distribution on latent variables). One can imagine on some models that converging from prior to posterior is a reduction of uncertainty given observations. Many Machine Learning problems could be written compactly and easily in probabilistic programming languages.

The idea of combining probabilistic programming and automatic programming (i.e. inference happens on a generative metamodel, which is written as a probabilistic program and which produces a stochastic model – a desired synthesized program) and the draft of road map for this direction will be introduced for discussion.

There is much connection with related fields, including Inductive Programming (especially with Probabilistic Inductive Logic Programming and Inductive Functional Programming), and these relations should be further explored.

7.3 Programming with Millions of Examples

Eran Yahav (Technion – Haifa, IL)

Joint work of Peleg,Hila; Mishne, Alon; Shoham, Sharon; Yang, Hongseok
Main reference H. Peleg, S. Shoham, E. Yahav, H. Yang, "Symbolic Automata for Static Specification Mining," in Proc. of the 20th Int'l Symp. on Static Analysis (SAS'13), LNCS, Vol. 7935, pp. 63–83, Springer, 2013; available as pre-print from the author's webpage.
URL http://dx.doi.org/10.1007/978-3-642-38856-9_6
URL http://www.cs.technion.ac.il/~yahave/papers/sas13-symaut.pdf

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 in a tool called PRIME and applied it to analyze and consolidate thousands of snippets per tested API.

7.4 Type Inhabitation Problem for Code Completion and Repair

Ruzica Piskac (Yale University, US)

Joint work of Gvero, Tihomir; Kuncak, Viktor; Kuraj, Ivan; Piskac, Ruzica
Main reference T. Gvero, V. Kuncak, I. Kuraj, R. Piskac, "Complete completion using types and weights," in Proc. of the 34th ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI'13), pp. 27–38, ACM, 2013.
URL http://dx.doi.org/10.1145/2491956.2462192

Developing modern software typically involves composing functionality from existing libraries. This task is difficult because libraries may expose many methods to the developer. In this talk I will describe a project called InSynth. InSynth synthesizes and suggests valid expressions of a given type at a given program point, to help developers overcome the problems described in the above scenarios. As the basis of InSynth we use type inhabitation for lambda calculus terms in long normal form. For ranking solutions we introduce a system of weights derived from a corpus of code. I will conclude with an idea how to extend this approach so that it also automatically repairs ill-typed code expressions.

7.5 Learning a Program's usage of Dynamic Data Structures from Sample Executions

David White (Universität Bamberg, DE)

Joint work of White, David; Lüttgen, Gerald
Main reference D. White, G. Lüttgen, "Identifying Dynamic Data Structures by Learning Evolving Patterns in Memory," in Proc. of the 19th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'13), LNCS, Vol. 7795, pp. 354–369, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-36742-7_25

Programs making heavy use of pointers are notoriously difficult to understand and analyze, especially when the programmer is given the freedom allowed by languages such as C. We aim to simplify such analyses by employing machine learning and pattern matching to automatically identify the pointer-based dynamic data structures used by a program. Through observing a sample execution of a program, we are able to discover and label operations responsible for manipulating dynamic data structures. The output of the approach is used for program comprehension and to partially automate contract-based verification.

7.6 SMT-based Videogame Synthesis

Sam Bayless (University of British Columbia – Vancouver, CA)

Joint work of Bayless, Noah; Bayless, Sam

In recent years there has been interest in using formal methods to do constraint-based content generation for videogames, for example using Answer-Set Programming to generate mazes. We introduce an SMT solver for directed graph reachability, and show how it can be used to efficiently generate mazes, puzzles, and dungeons in the style of traditional 2D videogames. We demonstrate an actual working videogame using this technique to generate levels online, and in real-time.

8 System Demonstrations

8.1 Storyboard Programming of Data Structure Manipulations

Rishabh Singh (MIT – Cambridge, US)

Joint work of Singh, Rishabh; Solar-Lezama, Armando
Main reference R. Singh, A. Solar-Lezama, "Synthesizing Data Structure Manipulations from Storyboards," in Proc. of the 19th ACM SIGSOFT Symp. and the 13th Europ. Conf. on Foundations of Software Engineering (ESEC/FSE'11), pp. 289–299, ACM, 2011.
URL http://dx.doi.org/10.1145/2025113.2025153

We present the Storyboard Programming framework, a new synthesis system designed to help programmers write imperative low-level data-structure manipulations. The goal of this system is to bridge the gap between the "boxes-and-arrows" diagrams that programmers often use to think about data-structure manipulation algorithms and the low-level imperative code that implements them. The system takes as input a set of partial input-output examples, as well as a description of the high-level structure of the desired solution. From this information, it is able to synthesize low-level imperative implementations in a matter of minutes.

The framework is based on a new approach for combining constraint-based synthesis and abstract-interpretation-based shape analysis. The approach works by encoding both the synthesis and the abstract interpretation problem as a constraint satisfaction problem whose solution defines the desired low-level implementation. We have used the framework to synthesize several data-structure manipulations involving linked lists and binary search trees, as well as an insertion operation into an And Inverter Graph.

9 Working Groups

9.1 Comparing Inductive Logic and Inductive Functional Programming as well as other Approaches to Program Synthesis

Stephen Muggleton (Imperial College London, UK)

Refinement graphs in ILP and IFP

Muggleton and De Raedt explained how refinement graphs are central to the theory of Inductive Logic Programming (ILP). Logic programs consist of a set of definite clauses. When formulating a hypothetical logic program, clauses can be constructed by successive refinement operations. For instance the empty clause can be refined to a clause with a head by adding an atom with all variables distinct. Body atoms and variable bindings can be added by applying further refinements. Robert Henderson discussed how in his thesis he had adapted refinement theory from ILP to Inductive Functional Programming (IFP). A key idea was successive refinement operations which added functions, function application and lambda variables to the initially unspecified functional program $\perp$.

Predicate invention

Muggleton described how predicate invention was a key idea in early work on Inverse Resolution in ILP as well as more recent work on Meta-Interpretive Learning. Predicate invention involves the introduction of auxilliary definitions to support the induction of the target predicate. For instance, when learning a definition for reversing a list it may be necessary to invent a predicate which appends two lists. The analogous IFP notion of function invention was also shown to be valuable and applicable in both Kitzelman's talk on IGOR2 and Henderson's talk on his recently completed thesis. However, according to Kitzelman it is not possible for IGOR2 to invent partition and append when learning quicksort because of incompleteness in the search. Hernandez-Orallo pointed out the problems that can be produced by generating too many auxilliary predicates, which overwhelm the search.

Abstraction methods in Formal Methods

Gulwani explained how formal methods use logic-based techniques for the specification, development and verification of software. Within Formal Methods abstraction techniques are used to constructed simplified models of a program, such as a finite state machine, in order to allow model checking to be used in verification. There was a discussion about the applicability of such abstraction methods within Inductive Programming. Henderson point out that Abstraction made learning more effective within IFP. It was agreed that the use of MetaRules in Meta-interpretive Learning is related to abstraction, though this needs further investigation to understand the relationship in more detail. Michael Hansen pointed out that multiple levels of abstraction are used by humans when carrying out programming tasks.

Search and Constraint Solving – Sat Solvers

Search is a key element of Inductive Programming which is used to uncover hypotheses which are consistent with the given examples. De Raedt explained declarative languages and modelling can be used to specify an Inductive Learning problem. This approach can be used with constraint programming and Sat solvers to provide an efficient way of carrying out machine learning tasks. In this way the constraints are presented to a solver along with the input data, which generates candidate hypotheses.

Multi-task learning

It was discussed whether it is more effective to solve several inductive programming tasks together. For example, Gulwani's FlashFill system is aimed at inducing a broad class of Inductive transformations, which include reformatting social security numbers, extracting names from email addresses and formulating acronyms such as "IBM" from a company name such as "International Business Machines". Eyal Dechter suggested that it should be possible to build common re-usable libraries which can be used multiply across such tasks.

Probabilistic reasoning versus ranking

Most Inductive Functional Programming approaches, such as Gulwani's FlashFill, rank hypotheses according to a score. Within other areas of Machine Learning, including ILP, it has become common to rank hypotheses according to use Baysian ranking based on posterior probability of the hypotheses given the examples. The relationship between informal ranking schemes and Bayesian ones was discussed. One key difference identified was the ability to use well-founded methods of Bayes' prediction with the probabilistic approaches.

Probabilistic Programs as density estimators

Yura Perov discussed the relationship between Inductive Programming and Probabilistic Programming. In place of a deterministic algorithm, probabilistic programming involves the use of a probabilistic generative model to transform the input to the output. Probabilistic Programs are useful for machine learning since they can be used to directly represent a prior of a structural space. For instance, a prior might be a latent Dirichlet allocation model, and the program can be used to implement a Gibbs sampler for density estimation over a space of solutions. Inductive Programming and Probabilistic Programming have been combined over the last decade within the areas of Statistical Relational Learning and Probabilistic inductive Logic Programming.

Development of benchmarks and data and system repositories

The group discussed the importance of developing benchmarks and repositories for comparing approaches. It was agreed that it would be difficult to find a common data format which would be applicable to all Inductive Programming systems. However, it was pointed out that in practice such datasets are transformed by experimenters into the appropriate form before application of their particular system.

Teaching materials – Joint meetings IP and ILP – Summer school

There was a discussion about the value of arranging joint meetings between the IP and ILP communities. This could be done by way of workshop co-location and/or joint Summer Schools. The advantage of the latter is that it could be used to develop teaching materials for use in undergraduate and graduate courses.

9.2 Potential New Areas of Applications and Challenges for Inductive Programming

Benjamin Zorn (Microsoft Research – Redmond, US)

Joint work of Grigoryev, Alexey; Kitzelmann, Emanuel; White, David; Yahav, Eran; Perelman, Daniel; Bayless, Sam; Zorn, Ben

The focus of the working group was to think about the classes of applications for which inductive programming would provide significant advantages over other machine learning approaches. Identifying these applications would both drive the research agenda of the community in applying the technology to important problems and also serve the basis for commercial applications of the technology. Part of our discussion was to consider why existing machine learning approaches, which have been very successful for certain classes of problems, would not also be sufficient for these inductive programming applications. In our discussion, we found that the following qualities might distinguish inductive programming solutions from other ML approaches: abstraction level of the solution (captured by the operators and data types available in a domain-specific language), performance of the learning process and solution (fewer training examples needed and the ability to apply conventional code generation techniques to the result), and readability.

In thinking generally, we made some specific observations about the class of interesting applications. We felt that many classification tasks alone were not complex enough to allow inductive programming solutions to outperform existing ML techniques. Furthermore, we felt that IP solutions to common tasks for which many traditional human-written implementations exist, such as scheduling an airline flight, were not good candidates for applications because synthesized solutions are likely to be significantly weaker than existing human-written ones. A better category of task would be one-off solutions to relatively simple problems for which an existing solution already written by a human is unlikely to exist. Examples of such tasks include extracting data, converting data formats, transforming text (as is currently done in Excel Flash Fill). These simple one-off applications have the negative aspect that it is unclear whether IP addresses the problem better than other ML techniques or not.

In thinking about IP and other ML techniques, we discussed the validity of the assumption that there are applications for which IP is a better solution. For example, could the things that are encoded in an IP solution (such as the set of operators and types in a domain-specific language) also be encoded as features in a neural net, etc. Is it possible that neural nets are equally effective as IP when the output of the neural net is a program? We discussed whether research addressing this distinction could be useful and help guide which IP applications provide the best opportunity. For example, does the existence of loops in IP solutions distinguish the class of applications for which IP is a better solution? Is there a middle-ground where some of the programming language artifacts present in IP solutions, such as types, recursion, and functional composition, are encoded into a neural net or other ML classification structure, getting the best of both approaches simultaneously.

In considering possible applications, we discussed several fruitful areas. Applications that require constraint solving, such as decision support, could be a productive area of investigation. Solutions expressed as programs have the property that they are human readable and hence can be checked for correctness and debugged as needed. Problems in this area often have a legitimate need for auditing. Hence applications related to manipulating data in spreadsheets or databases would be amenable to IP-based task synthesis solutions.

Reverse engineering of code from obfuscated sources is another possible area of application. Examples of possible areas where such an approach is needed include reverse engineering device drivers and code de-obfuscation. We all agreed that there are many common simple tasks on mobile devices or in specific applications that could be solved using IP. For example, simplifying the user interface to common mobile phone tasks (like sending a text message or checking the weather) or document handling (such as printing all files linked from a web page) would be valuable and result in relatively simple programs. Manipulating email was a particular area we agreed was both a real problem and a significant opportunity for applying IP.

We also discussed the problem of editing collections of pictures. The model where IP could be applied would be to edit one picture and then apply "similar" edits to the remaining collection of pictures, which might be quite large, saving time. A similar approach to editing presentations could be taken. Robots are another interesting source of opportunity. An approach where a robot is first trained to do a task by example and then uses IP to generalize that experience is compelling and appears to be an excellent fit for IP compared to other ML approaches. The same approach could be applied in a virtual space where an avatar is trained to do a task and then set free to repeat it indefinitely. Gold "farming" is a common task currently carried out in many virtual worlds by humans where automation could be commercially lucrative (although illegal by game rules). We also considered business process mining where many examples of traces of a process exist and a IP solution could extract structure from the trace data.

Another part of our discussion focused on challenges. One goal for identifying applications is to push the state of the art on specific important problems that need to be solved with research. We discussed several related challenges. One issue that arose was how to reliably determine in many scenarios where one training example starts and ends. Partitioning a stream of observations effectively into training instances is a difficult problem in itself. In thinking about the duality of IP versus other forms of ML, we wondered if there was a class of DSLs where there could be an automatic translation from an IP-based solution to a neural net, etc. Other challenges we discussed include the common concerns around scale: in the size of the hypothesis and the amount of background knowledge required. We also consider the problem of applying datacenter computers at scale to important tasks to be a challenge and not well investigated. This would be motivated more by having a "grand challenge" application or benchmark to drive the research like the recent efforts to push deep learning techniques using large scale application of unsupervised learning. Another important challenge is understanding the user experience aspects of such systems, specifically related to expectations about what user behavior is relevant to learning and how the resulting program artifact relates to it, especially for when the user is a non-programmer.

9.3 Benchmarks and Metrics

José Hernández-Orallo (Polytechnic University of Valencia, ES), Marco Ragni (Universität Freiburg)

Joint work of Martínez-Plumed, Fernando; Schmid, Ute; Siebers, Michael; Solar-Lezama, Armando; Hernández-Orallo, José; Katayama, Susumu

Many different inductive programming systems (including those in inductive logic programming) have been developed in the past 40 years. One of the key features of inductive programming is the use of very general and expressive languages for examples, hypotheses and background knowledge: logic programming, functional programming, higher-order, constraints, etc. Another feature is the wide range of applications: program synthesis, data manipulation, artificial intelligence, robotics, programming by demonstration, etc. As a result, it has been very difficult to compare different inductive programming systems, as they can use different languages and are used in diverse applications. This lack of comparison makes it difficult to properly evaluate the improvement and real breakthrough of new systems, and also makes it difficult to tell when a new system is performing worse (or no better) than other previous systems.

The existence of benchmarks would make it easy to test and develop new systems in inductive programming, as well as also recognising inductive programming as a distinctive discipline, in terms of the kind of problems it can solve, rather than the language representation or its applications. Consequently, the motivation of this working group is to see the possibility of elaborating benchmarks and metrics for inductive programming, arranged in a form of a repository, in order to assess the capabilities and limitations of existing and future inductive programming systems.

Regarding the language, we seemed to agree that a common representation syntax should be used for the benchmark problems in the repository. We would need to define a standard, as the ARFF file format for attribute-value data. In order to use the problems for different representations and systems, we would develop converters for some common languages, such as Haskell and Prolog, to ensure that we cover the inductive functional programming and inductive logic programming communities.

We were more precise about problem representation, and we identified that we needed to define the name of the function/predicate to be inferred, the datatypes involved and the examples. In the discussion, we clarified that a dataset does not configure a problem. Rather, from a dataset, we can do different sampling of examples and generate problems with more or less examples.

Moving from a repository to a benchmark requires further things. The first question is whether we are going to check solutions extensionally (over a test set) or intensionally (inspecting the code). In order to automate the process, it seems more reasonable to do this by separating between train and test cases and do it extensionally, as in machine learning.

In order to cover several domains, the working group suggested the following areas as a start:

- Traditional IP and ILP problems.
- AI problems, planning, robotics, etc.
- Program synthesis problems.
- Programming-By-Example problems.
- Trigonometry and other educational sources.
- IQ-like problems: number series, geometrical analogy problems (Raven's progressive matrices), Bongard Problems, etc.
- Grammatical inference problems.
- Structured prediction problems.
- Data manipulation and editing problems.

We identified several sources for these problems (such as IP and ILP repositories, but other sources, such as the program synthesis competition).

At the end of the meeting, we proposed a roadmap, starting with the problem representation, then going through a problem repository, before reaching the state of a benchmark repository or even a competition.

Participants

- Umair Zafrulla Ahmed
Indian Institute of Technology – Kanpur, IN
- Sam Bayless
University of British Columbia – Vancouver, CA
- Tarek R. Besold
Universität Osnabrück, DE
- Luc De Raedt
KU Leuven, BE
- Eyal Dechter
MIT – Cambridge, US
- Alexey Grigoryev
National Nuclear Research University MEPhI, RU
- Sumit Gulwani
Microsoft Res. – Redmond, US
- Mike Hansen
Indiana University – Bloomington, US
- Robert J. Henderson
Imperial College London, GB
- José Hernández-Orallo
Polytechnic University of Valencia, ES
- Petra Hofstedt
TU Cottbus, DE
- Frank Jäkel
Universität Osnabrück, DE
- Susumu Katayama
University of Miyazaki, JP
- Dileep Kini
University of Illinois – Urbana Champaign, US
- Emanuel Kitzelmann
Univ. Duisburg – Essen, DE
- Mark Marron
Microsoft Res. – Redmond, US
- Fernando Martínez-Plumed
Polytechnic University of Valencia, ES
- Martin Möhrmann
Universität Osnabrück, DE
- Stephen H. Muggleton
Imperial College London, GB
- Daniel Perelman
University of Washington – Seattle, US
- Iurii Perov
University of Oxford, GB
- Ruzica Piskac
Yale University, US
- Oleksandr Polozov
University of Washington – Seattle, US
- Marco Ragni
Universität Freiburg, DE
- Ute Schmid
Universität Bamberg, DE
- George M. Sergievsky
National Nuclear Research University MEPhI, RU
- Michael Siebers
Universität Bamberg, DE
- Rishabh Singh
MIT – Cambridge, US
- Armando Solar-Lezama
MIT – Cambridge, US
- Janis Voigtländer
Universität Bonn, DE
- David White
Universität Bamberg, DE
- Eran Yahav
Technion – Haifa, IL
- Benjamin Zorn
Microsoft Res. – Redmond, US

Report from Dagstuhl Seminar 13511

Software Engineering for Self-Adaptive Systems: Assurances

Edited by

Rogerio de Lemos[1], David Garlan[2], Carlo Ghezzi[3], and Holger Giese[4]

1 **University of Kent, GB,** R.Delemos@kent.ac.uk
2 **Carnegie Mellon University – Pittsburgh, US,** garlan@cs.cmu.edu
3 **Politecnico di Milano, IT,** carlo.ghezzi@polimi.it
4 **Hasso-Plattner-Institut – Potsdam, DE,** holger.giese@hpi.uni-potsdam.de

Abstract

The important concern for modern software systems is to become more cost-effective, while being versatile, flexible, resilient, dependable, energy-efficient, customisable, configurable and self-optimising when reacting to run-time changes that may occur within the system itself, its environment or requirements. One of the most promising approaches to achieving such properties is to equip software systems with self-managing capabilities using self-adaptation mechanisms. Despite recent advances in this area, one key aspect of self-adaptive systems that remains to be tackled in depth is assurances, i.e., the provision of evidence that the system satisfies its stated functional and non-functional requirements during its operation in the presence of self-adaptation. The provision of assurances for self-adaptive systems is challenging since run-time changes introduce a high degree of uncertainty during their operation. In this seminar, we discussed the problem of assurances for self-adaptive systems from four different views: criteria for assurances, composition and decomposition of assurances, feedback loop and assurances, and perpetual provisioning of assurances.

Seminar 15.–19. December, 2013 – www.dagstuhl.de/13511

1998 ACM Subject Classification D.2.10 [Software Engineering] Design, D.2.11 [Software Engineering] Software Architectures

Keywords and phrases software engineering, self-adaptive systems, assurances, criteria, feedback loop, decentralization

Digital Object Identifier 10.4230/DagRep.3.12.67

1 Executive Summary

Rogerio de Lemos
David Garlan
Carlo Ghezzi
Holger Giese

Repairing faults, or performing upgrades on different kinds of software systems have been tasks traditionally performed as a maintenance activity conducted off-line. However, as software systems become central to support everyday activities and face increasing dependability requirements, even as they have increased levels of complexity and uncertainty in their operational environments, there is a critical need to improve their resilience, optimize their performance, and at the same time, reduce their development and operational costs. This situation has led to the development of systems able to reconfigure their structure and modify their behaviour at run-time in order to improve their operation, recover from failures, and adapt to changes with little or no human intervention. These kinds of systems typically operate using an explicit representation of their own structure, behaviour and goals, and appear in the literature under different designations (e.g., self-adaptive, self-healing, self-managed, self-*, autonomic, etc.).

In spite of recent and important advances in the area, one key aspect of self-adaptive systems that poses important challenges yet to be tackled in depth is assurances: that is, providing evidence that systems satisfy their functional and non-functional requirements during operation. Specifically, assurances involve not only system dependability, but also resilience with respect to changes that may occur in the system, its environment, or its goals. The provision of assurances for self-adaptive systems, which should be done tandem with their development, operation and evolution, is difficult since run-time changes (e.g., resource variability) introduce a high degree of uncertainty that is atypical in more conventional systems.

This Dagstuhl Seminar has focused on the topic of obtaining assurances for self-adaptive software systems. Self-adaptive systems has been studied independently within different research areas of software engineering, including requirements engineering, modelling, architecture and middleware, event-based, component-based and knowledge-based systems, testing, verification and validation, as well as software maintenance and evolution [1, 2]. On the other hand, the topic of assurances for software-based systems has been widely investigated by the dependability community, in particular when considered in the context of safety-critical systems. For these types of systems there is the need to build coherent arguments showing that the system is able to comply with strict functional and non-functional requirements, which are often dictated by safety standards and general safety guidelines. The major challenge when combining self-adaptability and dependability is how to obtain assurances regarding the uncertainty of changes that may affect the system, its environment or its goals.

References

1 B. H. Cheng, H. Giese, P. Inverardi, J. Magee, and R. de Lemos. 08031 Abstracts Collection – Software Engineering for Self-Adaptive Systems. In B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, editors, *Software Engineering for Self-Adaptive Systems*, number 08031 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany.

2 R. de Lemos, H. Giese, H. Müller, and M. Shaw. 10431 Report – Software Engineering for Self-Adaptive Systems. In R. de Lemos, H. Giese, H. Müller, and M. Shaw, editors, *Software Engineering for Self-Adaptive Systems*, number 10431 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2011. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany.

2 Table of Contents

3 Key Topics on Assurances

The aim of the seminar was not so much to be comprehensive concerning the topics associated with assurances for self-adaptive software systems, but to be focused on key and challenging topics.

3.1 Composition and Decomposition of Assurances

Contributors: *J. Andersson, B. Schmerl, L. Baresi, Y. Brun, M. B. Cohen, A. Gorla, C. M. F. Rubira, T. Vogel, F. Zambonelli*

In software development, system goals are often provided assurances. An *assurance* is "a reasoned and compelling argument, supported by a body of evidence, that a system, service or organisation will operate as intended for a defined application in a defined environment" [1]. Examples of goals include enforcing certain system properties, such as safety, security, and reliability. The most widely accepted approaches for assuring suchgoals are based on producing evidence, and then arguing that a system meets its specification based on that evidence. An *assurance case* [2] is composed of subgoals, strategies, contexts, and evidence, tied together into an argument justifying that the goal will be met. For example, safety cases are *"a documented body of evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment"* [3, p. 8].

We are interested in assurances for engineering a particular kind of software system: self-adaptive systems. In such systems, self-adaptation may be used as a mechanism for achieving a particular goal, or may be a mechanism in spite of which a goal must be satisfied. We are particularly interested in the process of building self-adaptive systems out of self-adaptive components (and further, recursively building larger self-adaptive systems from self-adaptive subsystems). Thus, we aim to understand (1) how assurances of self-adaptive components compose to form an assurance of a system composed of those components, and (2) how an assurance of a system can be decomposed into assurances its components must satisfy. While the challenges of composition and decomposition are neither unique to *assurances* within self-adaptive systems engineering [4, 5], nor to *self-adaptive* systems, composing and decomposing assurances within such systems poses special challenges, on which we focus here.

1. **How to accommodate a changing environment?** In traditional software development, assurance cases often make assumptions about fixed goals and environments. By contrast, self-adaptive systems often relax these assumptions. This creates a challenge for composition and decomposition of assurances cases of self-adaptive components and systems. For example, for composition, it is often insufficient to consider the assurance case solely within the scope of the component, and must instead be considered within the larger scope of the system. Unfortunately, this breaks the component abstraction and complicates reasoning about assurances.
2. **What are potential sources of evidence?** As some self-adaptive components can be made up of a separate managed part and managing part [6], understanding how assurances on one part affect or can result in assurances on the whole poses a challenge. At the very least, these pose multiple sources of evidence, the combination of which must be understood.
   Furthermore, what kinds of evidence can be provided by the different activities of self-adaptation. For example, what kinds of evidence can monitoring provide? Is this type

of evidence different to the type that analysis can provide? What kinds of assurances can emerge from more biological forms of self-adaptation? Evidence may also come from other engineering activities, e.g., design decisions, validation and verification, or the engineering process. These all impact how assurances can be composed and decomposed. For example, design decisions such as the choice of architectural style may provide some evidence for assurance cases.

3. **What strategies can be used to combine assurances when self-adaptive systems are composed?** To address this challenge, we must answer further questions such as "How to design assurances so that they can be composed with one another?", "Which properties of assurances lend themselves to composition, and which do not?", "How can assurance composition strategies be reused?", "How can evidence from assurance cases be shared and reused among products or components?", and "How do composition and decomposition of assurance cases affect the amounts of evidence needed to support those cases?"

To begin to address these challenges, we plan to use a well-known method for constructing assurance cases known as the Goal Structuring Notation (GSN) [1, 7]. This notation allows engineers to codify an assurance case as a set of *strategies*, *contexts*, *assumptions*, *justifications*, and *solutions*. This structure serves as the argument for the assurance of a specific goal. It is likely that this notation will need to be adjusted to apply to address the first two challenges.

To address the third challenge, Bate and Kelly [8] outline a strategy for composing assurance arguments. However, this technique imposes constraints on the underlying system architecture. No comprehensive techniques exist today for composing assurances in a more general system-of-systems scenario, but this work may form a basis to build upon.

References

1 Goal Structuring Notation (GSN) community standard version 1, November 2011. Available at http://goalstructingnotation.info.

2 Peter Bishop, Robin Bloomfield, and Sofia Guerra. The future of goal-based assurance cases. In *Proc. Workshop on Assurance Cases*, pages 390–395, 2004.

3 Robin Bloomfield, Bishop Peter, Claire Jones, and P. Froome. *ASCAD — Adelard Safety Case Development Manual*. Adelard, 3 Coborn Road, London E3 2DA, UK, 1998.

4 Yuriy Brun, Giovanna Di Marzo Serugendo, Cristina Gacek, Holger Giese, Holger Kienle, Marin Litoiu, Hausi Müller, Mauro Pezzè, and Mary Shaw. Engineering self-adaptive systems through feedback loops. In Betty H.C. Cheng, Rogério de Lemos, Holger Giese, Paola Inverardi, and Jeff Magee, editors, *Software Engineering for Self-Adaptive Systems*, volume 5525 of *Lecture Notes in Computer Science (LNCS)*, pages 48–70. Springer-Verlag, 2009. `doi:10.1007/978-3-642-02161-9_3`.

5 Danny Weyns, Bradley Schmerl, Vincenzo Grassi, Sam Malek, Raffaela Mirandola, Christian Prehofer, Jochen Wuttke, Jesper Andersson, Holger Giese, and Karl M. Göschka. On patterns for decentralized control in self-adaptive systems. In Rogério de Lemos, Holger Giese, Hausi A. Müller, and Mary Shaw, editors, *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science (LNCS)*, pages 76–107. Springer-Verlag, 2013. `doi:10.1007/978-3-642-35813-5_4`.

6 Danny Weyns, Sam Malek, and Jesper Andersson. FORMS: Unifying reference model for formal specification of distributed self-adaptive systems. *ACM Transactions on Autonomous and Adaptive Systems*, 7(1):8:1–8:61, May 2012. `doi:10.1145/2168260.2168268`.

7 T. P. Kelly and J. A. McDermid. Safety case construction and reuse using patterns. In Peter Daniel, editor, *Safe Comp*, pages 55–69. Springer, London, 1997. URL: http://dx.doi.org/10.1007/978-1-4471-0997-6_5, `doi:10.1007/978-1-4471-0997-6_5`.

8 Iain Bate and Tim Kelly. Architectural considerations in the certification of modular systems. *Reliability Engineering & System Safety*, 81(3):303 – 324, 2003. doi:10.1016/S0951-8320(03)00094-2.

3.2 Feedback Loop and Assurances

Contributors: *H. A. Müller, N. M. Villegas, V. Cortellessa, A. Filieri, H. Giese, G. Karsai, A. Leva, M. Pezzè, R. Rouvoy, E. Rutten, M. Shaw, G. Tamura*

Feedback loops are cornerstones of software-intensive self-adaptive systems (SASs). In this chapter, we study the relationships existing between feedback loops and the types of assurance SASs can provide focusing on the conceptual rather implementation level of the feedback model. The review includes, on the one hand, how feedback loops contribute to providing assurances about the controlled system and, on the other hand, how assurances improve the realisation of feedback loops in SASs. To set the stage for the discussion of concrete examples and their challenges, we first study the parallels between traditional engineering control theory [1, 2] and the more recent research on feedback control of software computing systems (e.g., MAPE-K loops) [3, 4] in the realm of SASs in order to establish a common vocabulary of key concepts, such as the disturbance affecting the system or the control actions used by the controller to adapt the system. This provides a basis to discuss concrete examples that allow us to illustrate open challenges in the engineering of SASs [5]. For each of these examples, we identify the main concepts related to the shared vocabulary and highlight the need and purpose of feedback loops. Furthermore, we outline the related assurances, including concrete assurance goals, techniques employed for assurances, and how goals and techniques relate to the feedback loops present in SASs. By studying concrete examples, we posit key challenges for assurance research related to feedback loops in self-adaptive software.

1. **Identification of the core phenomena to control**: in control theory, system transfer functions based on differential equations are the models on which principles and properties of the phenomena to control are described and analysed [1, 6, 2]. Depending on the behavioural characteristics of the target system, a controller can be defined to make the system behave as desired. The desired behaviour is usually specified by either providing a reference input the system should follow (e.g., PID controllers), or as an optimisation problem (e.g., Model Predictive Control) [1]. The characteristics of a controller are usually defined by adjusting its parameters, which have special significance to generate the signals that will adapt the system depending on how far measured outputs are from the corresponding reference inputs. The controllers are indeed designed as (parametric) transfer functions that generate the control signals that drive the target system towards accomplishing its goals. In software systems the identification of the core phenomena to control is typically a complex task. This is mainly because, in contrast to physical systems, software systems still lack general methods to model the multi-dimensional and non-linear relationships between system goals and adaptation mechanisms [6, 7, 8]. For example, an adaptation mechanism can reconfigure the system by applying an architectural pattern with the goal of improving the system performance. However, it is still an open challenge to model the exact effect of this pattern in the performance of the system. Many research questions remain open in the identification and modelling of the core phenomena to control in software systems. For example, how to model explicitly and accurately the relationship among system goals, adaptation mechanisms, and the effects produced by

actuators? Can we design software systems having an explicit specification of what we want to assure with control-based approaches? Can we do it by focusing only on some aspects for which feedback control is more effective? Can we improve the use of control, or achieve control-based design, by connecting as directly as possible to some "real physics" inside the software system? What techniques can we use for this? How far can we go by modelling SAS systems mathematically? What are the limitations?

2. **Composition and incrementality**: performing validation and verification (V&V) tasks over the entire system—at runtime, to guarantee desired properties and goals, is often infeasible due to prohibitive computational costs. Therefore, the assurance of SAS systems requires composable V&V tasks that can be applied incrementally along the adaptation loop. With respect to composition, relevant research questions include: what are the problems that require the composition of assurance mechanisms? What are suitable techniques to realise the composition of V&V tasks? What approaches can we borrow from testing? How can we reuse or adjust them for the assurance of SAS systems? Regarding incrementality: in which cases is it useful? How can incrementality be realised? How to characterise increments, and their relationship to system changes?
3. **Synergy between control and software engineering**: despite the efforts to apply control theory to the engineering of SASs, evidenced for instance in [5, 9, 10, 6, 3, 11, 12], the assurance of SASs still demands effective ways of integrating control theory and software engineering. Research questions to advance towards this direction include: what are the best practices from both sides that can be exploited in the assurance of SAS systems? How to construct reliable controllers leveraging formal model techniques used in software engineering? Can we define and/or use anti-patterns in control-based assurance of SAS systems? How to apply off-line V&V mechanisms at runtime? How to close the semantic gap between control in physical systems and control in software systems? Can we define a model in the middle of these two worlds? Would these models be domain or problem specific? Can we characterise (at least some of) these problems and corresponding suitable models? How can we educate software engineers in the application of control to the design of software systems?
4. **Management of viability zones**: the viability zone of a SAS system can be defined as the set of possible states in which the system operation is not compromised [13]. Moreover, it can be characterised in terms of relevant context attributes and corresponding desired values [14], and can change with context changes. In effect, the viability zone of a target system under adaptation constantly varies with and along adaptation dimensions. The dynamic nature of viability zones is a relevant research challenge in the assurance of SAS systems because it implies adjusting the domain coverage not only of design and adaptation realisation but also of V&V tasks. Open research questions in this aspect include: what are runtime models that can be used for the incremental and dynamic derivation of software artefacts for implementing V&V tasks? How to maintain the causal connection between viability zones, adapted system, and its corresponding V&V software artefacts? How to adapt these artefacts at runtime?

References

1 Karl Johan Astrom and Bjorn Wittenmark. *Adaptive Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994.

2 Richard M. Murray, editor. *Control in an Information Rich World*. Society for Industrial and Applied Mathematics, 2003. URL: http://epubs.siam.org/doi/abs/10.1137/1.9780898718010, `arXiv:http://epubs.siam.org/doi/pdf/10.1137/1.9780898718010`, `doi:10.1137/1.9780898718010`.

3 Joseph L. Hellerstein, Yixin Diao, Sujay Parekh, and Dawn M. Tilbury. *Feedback Control of Computing Systems*. John Wiley & Sons, 2004.

4 Jeffrey O. Kephart and David M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, January 2003. URL: http://dx.doi.org/10.1109/MC.2003.1160055, doi:10.1109/MC.2003.1160055.

5 Yuriy Brun, Giovanna Di Marzo Serugendo, Cristina Gacek, Holger Giese, Holger Kienle, Marin Litoiu, Hausi Müller, Mauro Pezzè, and Mary Shaw. Engineering self-adaptive systems through feedback loops. In Betty H.C. Cheng, Rogério de Lemos, Holger Giese, Paola Inverardi, and Jeff Magee, editors, *Software Engineering for Self-Adaptive Systems*, volume 5525 of *Lecture Notes in Computer Science (LNCS)*, pages 48–70. Springer-Verlag, 2009. doi:10.1007/978-3-642-02161-9_3.

6 Antonio Filieri, Carlo Ghezzi, Alberto Leva, and Martina Maggio. Automated design of self-adaptive software with control-theoretical formal guarantees. In *Proceedings of the 36th IEEE/ACM International Conference on Software Engineering*, ICSE 2014, page (to appear), New York, NY, USA, 2014. ACM.

7 T. Patikirikorala, A. Colman, J. Han, and Liuping Wang. A systematic survey on the design of self-adaptive software systems using control engineering approaches. In *Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2012 ICSE Workshop on*, pages 33–42, June 2012. doi:10.1109/SEAMS.2012.6224389.

8 Norha M. Villegas, Hausi A. Müller, Gabriel Tamura, Laurence Duchien, and Rubby Casallas. A framework for evaluating quality-driven self-adaptive software systems. In *Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '11, pages 80–89, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/1988008.1988020, doi:10.1145/1988008.1988020.

9 Antonio Filieri, Carlo Ghezzi, Alberto Leva, and Martina Maggio. Self-adaptive software meets control theory: A preliminary approach supporting reliability requirements. In *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, ASE '11, pages 283–292, Washington, DC, USA, 2011. IEEE Computer Society. URL: http://dx.doi.org/10.1109/ASE.2011.6100064, doi:10.1109/ASE.2011.6100064.

10 A. Filieri, C. Ghezzi, A. Leva, and M. Maggio. Reliability-driven dynamic binding via feedback control. In *Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2012 ICSE Symposium on*, pages 43–52, June 2012. doi:10.1109/SEAMS.2012.6224390.

11 Hausi Müller, Mauro Pezzè, and Mary Shaw. Visibility of control in adaptive systems. In *Proceedings of the 2Nd International Workshop on Ultra-large-scale Software-intensive Systems*, ULSSIS '08, pages 23–26, New York, NY, USA, 2008. ACM. URL: http://doi.acm.org/10.1145/1370700.1370707, doi:10.1145/1370700.1370707.

12 Gabriel Tamura, Norha M. Villegas, HausiA. Muller, JoãoPedro Sousa, Basil Becker, Gabor Karsai, Serge Mankovskii, Mauro Pezzè, Wilhelm Schäfer, Ladan Tahvildari, and Kenny Wong. Towards practical runtime verification and validation of self-adaptive software systems. In Rogerio Lemos, Holger Giese, Hausi. Müller, and Mary Shaw, editors, *Software Engineering for Self-Adaptive Systems II*, volume 7475 of *Lecture Notes in Computer Science*, pages 108–132. Springer Berlin Heidelberg, 2013. URL: http://dx.doi.org/10.1007/978-3-642-35813-5_5, doi:10.1007/978-3-642-35813-5_5.

13 Jean-Pierre Aubin, Alexandre Bayen, and Patrick Saint-Pierre. *Viability Theory: New Directions*. Springer, 2011. URL: http://hal.inria.fr/inria-00636570, doi:10.1007/978-3-642-16684-6.

14 Norha M. Villegas. *Context Management and Self-Adaptivity for Situation-Aware Smart Software Systems*. PhD thesis, University of Victoria, Canada, February 2013.

3.3 Perpetual Provisioning of Assurances

Contributors: *G. Tamburrelli, D. Weyns, N. Bencomo, R. Calinescu, J. Camara, C. Ghezzi, V. Grassi, L. Grunske, P. Inverardi, J. M. Jezequiel, S. Malek, R. Mirandola, M. Mori*

The breakout group focused first of finding a clear and unanimous definition of "assurances for self-adaptive systems" and then focused on (1) studying solutions for assurances of self-adaptive systems together with benchmark criteria, and (2) defining a unifying case study to be used by practitioners and academics to assess, challenge and compare their solutions. After a debate we came up with the following definition: "Assurances for self-adaptive systems means providing evidences for requirements compliance (functional and non-functional) on and off-line". With perpetual assurances, the group refers to the perpetual process of evidence provision for assurance that blends system and human activities throughout the live cycle of a self-adaptive system. Given this premise, the group concentrated on two aspects discussed in two subgroups.

One subgroup discussed key challenges of perpetual assurances for self-adaptive systems, requirements for solutions, realisation techniques, mechanism to make solutions suitable, and benchmark criteria to compare solutions. Identified challenges include: uncertainty, incompleteness, heterogeneity of systems, dynamism (changes in requirements, environment . . .), scale of systems, distribution, and user/resource constraints. The following requirements of the solution were discussed: dealing with uncertainty and incompleteness, dealing with variants, timeliness, overhead, and scalability (efficiency), traceability of adaptation decisions (convey the rationale and evidences), user-in-the-loop, and reactive vs. proactive(speculative) vs. predictive provision of assurances. Discussed techniques to provide assurances are model checking, testing, simulation and statistical analysis, proving, runtime verification, and sanity checks. We identified possible mechanisms for making the techniques suitable: incrementally, compositionality, hierarchical reasoning (fallback mechanisms), parallelism/off-loading, parameterisation, abstraction, learning, and caching. Key benchmark criteria identified were: (1) does the approach provides evidence with respect to (can your approach handle); (2) does your approach provide evidence based on the current state, the past, projection in the future (or any combination of these; (3) how efficient (effective) is your approach in providing evidence, e.g., wrt. timeliness given a set of system reconfigurations, and overhead (memory, CPU).

The other subgroup discussed the definition of the case study. We decided do focus on the specific domain of Service Oriented Architectures for several reasons: popularity, flexibility, and incremental complexity. First the subgroup identified three distinct areas of assurances: assurances w.r.t. the services, the users, and the requirements. As a consequence of this initial assessment, the definition of the case study went through several refinement steps that included the definition of a series of challenges, techniques, and mechanisms that the case study implementers may exploit. This process culminated with the definition of an initial list of scenarios and a list of categories that detailed the case study. This refinement process and the scenario characterisation is still an on-going process and is part of the future work of the group. The final goal of the subgroup is to have a repository of one or more concrete case study that may be used as concrete benchmarks for the community.

The joint group discussed the results of subgroup discussions and decided to elaborate further on solutions for perpetual assurances (challenges, requirements, techniques and mechanisms, and benchmark criteria) and case studies that allow benchmarking. To that end, the group plans to write a joint book chapter that integrates both aspects.

4 Outcomes

The two concrete outcomes from this seminar will be a challenges paper and a new book. The challenges paper, which will follow the same format from previous roadmap papers will be structured according to the topics to be identified during the seminar, namely: criteria for assurances, composition and decomposition of assurances, feedback loop and assurances, and perpetual provisioning of assurances. For each topic, the objective is to summarise the current state-of-the-art, discuss its limitations, and identify future challenges for the field.

The book will contain state-of-the-art contributions from participants of the seminar and some invited contributions. In addition to these contributions, the roadmap paper will be the introductory chapter of the book, which should be followed by four chapters containing extended versions of the topics discussed in the roadmap paper. The book will be published by Springer as Lecture Notes in Computer Science volume on their State-of-the-Art series.

5 Overview of Talks

5.1 Self-Adaptive Authorisation Infrastructures: Managing malicious behaviour

Chris Bailey (University of Kent, GB)

Joint work of Bailey, Chris; de Lemos, Rogerio; Chadwick, David

Authorization infrastructures are an integral part of any network where electronic resources require protection. As networks expand and organizations begin to federate access to their resources, authorization infrastructures become increasingly challenging to manage. This talk outlines recent works in regards to self-adaptive security, specifically self-adaptive authorization. This explores the automatic adaptation of authorization assets (such as access control policies and subject access rights) for the handling of malicious user behavior. We identify the core motivation for our work, our current progress, and a short adaptation scenario that demonstrates key aspects and challenges of self-adaptive authorization. Finally, we identify our evaluation approach and discuss how our work applies to self-adaptive assurances, in regards to verification and validation of adaptations.

5.2 A Fine-grained Autonomic Management Solution for Multi-layered Systems

Luciano Baresi (Technical University of Milan, IT)

The service paradigm, together with virtualization technology, is imposing a profound re-thinking of many complex software systems. Providing virtual infrastructures as services is gaining more and more momentum, and is imposing a more cohesive view of the different layers that constitute a software system. Applications, service platforms, and virtualized infrastructures have become tightly integrated. It is now possible to change a software's quality of service (QoS) by changing the configuration of the virtual machines it uses. We can even instantiate new virtual machines to address load problems, or move our software from one infrastructure to another if its quality is not acceptable. Even if one might say that this is nothing new, the key distinctive feature is the ease with which the different parameters can be changed, and the different runtime executors (e.g., virtual machines) added or modified to impact a system's behavior. Originally, monitoring of service-oriented systems was only performed at the application layer, and the lower layers were considered to be a constant. Recently, however, cross-layer monitoring is imposing itself as a promising and challenging research problem. Available technologies provide users with means to tackle the problem of the quality of service of these applications by digging down into the different layers. However, the more complexity we want to tame, the more sophisticated our solutions must become. If we take advantage of the ease and low impact of installing software probes within the different layers, the amount of data we collect can grow very rapidly. This calls for efficient and precise methods for their management. We need a customizable and extensible way to collect, aggregate, and analyze data, in order to identify the causes of anomalous behaviors. This talk introduces a new approach for the cross-layer monitoring of complex service-based systems. The approach proposes two main novel contributions. We present mlCCL, a novel and extensible declarative language that designers can use to define (i) the various data they want to collect from the layers in their system, (ii) how to aggregate them to build higher-level knowledge, and (iii) how to analyze them to discover undesired behaviors. The talk also presents ECoWare (Event Correlation Middleware), an event correlation and aggregation middleware that supports mlCCL specifications. It provides advanced data aggregation and analysis features, and can be used to probe systems. Ecoware also provides a dashboard for reasoning on multiple dimensions of a running system at the same time, and for performing drill-down analyses to discover the causes of a revealed anomaly.

5.3 Artificial software diversity: automatic synthesis of program sosies

Benoit Baudry (INRIA Rennes – Bretagne Atlantique, FR)

Recent work have exploited the plastic properties of software to develop unsound program transformations. These transformations modify the program's behavior while staying in acceptable correctness boundaries with the aim of improving some qualitative attribute (e.g., response-time, fault-tolerance, etc.). This work reports on a novel form of unsound transformation, which consists in generating program 'sosies'. Sosies of a program P are variants that exhibit exactly the same observable behavior, through different execution paths ('sosie' is the French word for look-alike). We use the test suite of the program as the specification of acceptable correctness and we experiment transformations that replace code in the program by other code that comes from the same program. We define transformations at different levels of granularity (expression, statement, block). Here we report on the feasibility of automatic synthesis of sosies, at different granularities, on a set of open source Java programs of very different sizes. We show that transformations, which consider a part of the program's semantics are more effective for sosie synthesis than pure random transformations.

5.4 Bayesian Artificial Intelligence for Tackling Uncertainty in Self-Adaptive Systems: The Case of Dynamic Decision Networks

Nelly Bencomo (Aston University – Birmingham, GB)

In recent years, there has been a growing interest towards the application of artificial intelligence approaches in software engineering (SE) processes. In the specific area of SE for self-adaptive systems (SASs) there is a growing research awareness about the synergy between SE and AI. However, just few significant results have been published. We report and discuss our own experience using Dynamic Decision Networks (DDNs) to model and support decision-making in SASs while explicitly taking into account uncertainty. In this session we talk about the application of our DDN-based approach to the case of an adaptive remote data mirroring system. We discuss results, implications and potential benefits of the DDN to enhance the development and operation of self-adaptive systems, by providing mechanisms to cope with uncertainty and automatically make the best decision. We also discuss the ongoing work on a Bayesian definition of surprise as the basis for quantitative analysis to measure degrees of uncertainty and deviations of self- adaptive systems from normal behavior. A surprise measures how observed data affects the models or assumptions of the world during runtime. The key idea is that a surprising event can be defined as one that causes a large divergence between the belief distributions prior to and posterior to the event occurring. In such a case the system may decide either to adapt accordingly or to flag that an abnormal situation is happening.

5.5 Inferring models for verification

Yuriy Brun (University of Massachusetts – Amherst, US)

Model inference – constructing a model of an implementation based on execution information – can help greatly with ensuring that requirements are satisfied, enable exploration and evaluation of potential adaptations, and predicting the effects of adaptations.

5.6 Self-Adaptive Software Assurance through Continual Verification of Non-Functional Properties

Radu Calinescu (University of York, GB)

Software systems are used in business-critical and safety-critical applications from domains ranging from e-commerce and e-government to finance and healthcare. Many of these systems must comply with strict non-functional requirements while evolving in response to changes in their environment and requirements. My talk will describe how such compliance can be achieved through the run-time use of quantitative verification, and discuss the challenges that must be overcome to make this continual verification feasible for real-world software systems.

5.7 Failure Avoidance using Feature Locality

Myra B. Cohen (University of Nebraska – Lincoln, US)

Joint work of Cohen, Myra B.; Garvin, Brady J.; Dwyer, Matthew B.; Swanson, Jacob

Despite the best efforts of software engineers, faults still escape into deployed systems. Developers need time to prepare and distribute fixes, and in the interim, deployments must either avoid failures or endure their consequences. Configurable software, software in which features can be added or removed at run-time, are known to suffer from failures that appear only under certain feature combinations, and these failures are particularly challenging for testers, who must find suitable configurations as well as inputs to detect them. We believe that these failures are well suited for avoidance by self-adaptation. This talk discusses that possibility by leveraging a phenomenon we call feature-locality that allows us to use historical data to predict failure-prone configurations and hence reconfiguration workarounds. We have evidence from two case studies that feature locality exists, and that our algorithms can improve time to failure avoidance as more and more history is incorporated. We have implemented a version of our avoidance algorithms within the Rainbow framework and show preliminary results of this study.

5.8 Applying Model Differences to Automate Performance-Driven Refactoring of Software Models

Vittorio Cortellessa (University of L'Aquila, IT)

Identifying and removing the causes of poor performance in software systems are complex problems, and these issues are usually tackled after software deployment only with human-based means. Performance anti patterns can be used to harness these problems since they capture design patterns that are known leading to performance problems, and they suggest refactoring actions that can solve the problems. This talk introduces an approach to automate software model refactoring based on performance antipatterns. A Role-Based Modeling Language is used to model antipattern problems as Source Role Models (SRMs), and antipattern solutions as Target Role Models (TRMs). Each (SRM, TRM) pair is represented by a difference model that encodes refactoring actions to be operated on a software model to remove the corresponding antipattern. Differences are applied to software models through a model transformation automatically generated by a higher-order transformation. The approach is shown at work on an example in the e-commerce domain.

5.9 Assurance of Autonomous Adaptive Systems: (Some) Lessons Learned

Bojan Cukic (West Virginia University – Morgantown, US)

Over the past decade, we have been involved in design, verification and validation of self-adaptive systems in two very different domains: avionics and border management. While the first is safety critical and adaptation addresses unanticipated aircraft failures, the second is critical for national security. In border management, adaptation adjusts the accuracy of traveler identification with the throughput of the border crossing under changing arrival patterns and security requirements. Although these domains appear to have little in common, the assurance arguments we made about systems and adaptation appear to point to common underlying principles. The first commonality is the importance of the choice of an appropriate level of detail available in the model of the system that explores adaptation options (the controller). The strength of assurance arguments is inextricable from the granularity the model supports. We also conclude that the overall goal of assurance of self-adaptive systems is maintaining the assurance case made prior to system deployment. In other words, any adaptation should support existing assurance arguments even if system's functionality or non-functional properties evolve. Finally, the selection of assurance techniques used in case studies points to the diversity. As assurance goals are system specific, the techniques to achieve them vary too.

5.10 Software engineering for self-adaptive software: motivational talk

Carlo Ghezzi (Technical University of Milan, IT)

Research on self-adaptive software systems must mature in a way that assurances can be given on dependability of adaptation. Dependability assurance means that one must be able to show satisfaction of the following fundamental argument (FA): S (specification of the software) and E (environment assumptions and properties) entail satisfaction of R (the requirements)

FA was first stated by Jackson and Zave in their foundational work on requirements. Because of changes in D (and R), FA must be shown to hold not only at design time, but it must be continuously checked also at run time. If the run-time check is done by reasoning on models (of S and D), then a violation of FA must lead to a change in S (and hence in the running software). Possibly, the change in S must be accomplished in a self-managed manner.

5.11 Assurance for Self-Adaptive Software and Models

Holger Giese (Hasso-Plattner-Institut – Potsdam, DE)

This presentation discusses, which role development-time models and runtime models can play for the assurance of self-adaptive software. On the one hand the role of models to assure the proper functioning of the adaptation algorithm is addressed. On the other hand also the roles of model for ensuring the correct implementation is covered. Furthermore, concrete research results for the different cases were presented to outline concrete results that have been achieved for these different roles of models for self-adaptive software.

5.12 Fully Decentralized Service Assembly under Non Functional Requirements

Vincenzo Grassi (University of Rome "Tor Vergata", IT)

Joint work of Grassi, Vincenzo; Mirandola, Raffaela; Marzolla, Moreno

We present a fully decentralized solution to the adaptive self-assembly of distributed services. The proposed solution is able to build and maintain an assembly of services, guaranteeing the fulfillment of both functional requirements, and non functional requirements concerning global quality of service (QoS) and structural properties. The key aspect of our solution is the use of a gossip protocol to achieve decentralized information dissemination and decision making. Simulation experiments show the effectiveness of our approach in terms of robustness, and convergence speed.

5.13 Runtime Quality Problem Detection Techniques with Statistical Techniques: Theory and Practical Applications

Lars Grunske (Universität Stuttgart, DE)

Software systems may suffer at runtime from changes in their operational environment or/and requirements specification, so they need to be adapted to satisfy the changed environment or/and specifications [5]. The research community has developed a number of approaches to building adaptive systems that respond to these changes such as Rainbow [6]. Additionally currently, several approaches have been proposed to monitor QoS attributes at runtime with the goal of reactively detecting QoS violations (e.g. [9]).

The presentation will introduce some reactive [1, 3, 8] and proactive [2, 4] detection techniques.

References

1 Ayman Amin, Alan Colman, and Lars Grunske. Using Automated Control Charts for the Runtime Evaluation of QoS Attributes. In Proc. of the 13ht IEEE Int. High Assurance Systems Engineering Symposium, pages 299–306. IEEE Computer Society, 2011.
2 Ayman Amin, Alan Colman, and Lars Grunske. An Approach to Forecasting QoS Attributes of Web Services Based on ARIMA and GARCH Models. In Proc. of the 19th Int. Conf. on Web Services, pages 74–81. IEEE, 2012.
3 Ayman Amin, Alan Colman, and Lars Grunske. Statistical Detection of QoS Violations Based on CUSUM Control Charts. In Proc. of the 3rd ACM/SPEC Int. Conf. on Performance Engineering, pages 97–108. ACM, 2012.
4 Ayman Amin, Lars Grunske, and Alan Colman. An automated approach to forecasting QoS attributes based on linear and non-linear time series modeling. In Proc. of the 27th IEEE/ACM Int. Conf. on Automated Software Engineering, pages 130–139. IEEE, 2012.
5 Radu Calinescu, Lars Grunske, Marta Z. Kwiatkowska, Raffaela Mirandola, and Giordano Tamburrelli. Dynamic QoS Management and Optimization in Service-Based Systems. IEEE Trans. Software Eng., 37(3):387–409, 2011.
6 David Garlan, Shang-Wen Cheng, An-Cheng Huang, Bradley Schmerl, and Peter Steenkiste. Rainbow: Architecture-based self-adaptation with reusable infrastructure. Computer, 37(10):45–54, 2004.
7 Lars Grunske: An effective sequential statistical test for probabilistic monitoring. Information & Software Technology 53(3): 190–199 (2011)
8 Lars Grunske, Pengcheng Zhang: Monitoring probabilistic properties. ESEC/SIGSOFT FSE 2009: 183–192
9 Raffaela Mirandola and Pasqualina Potena. A QoS-based framework for the adaptation of service-based systems. Scalable Computing: Practice and Experience, 12(1), 2011.

5.14 Pluggable Verification for Models at Runtime

Jean-Marc Jezequel (University of Rennes, FR)

We propose an approach to integrate the use of ad hoc verifiers (e.g. time-related stochastic properties) in a continuous design process based on models at runtime. Assurance guarantees are an important aspect of component-based architectures, for instance in distributed, volatile networks of computation nodes. The models at runtime approach eases the management of such architectures by maintaining abstract models of architectures synchronized with the physical, distributed execution platform. For self-adapting systems, prediction of quality attributes such as delays and throughput of a component assembly is of utmost importance to take adaptation decision and accept evolutions that conform to non functional specifications. To this aim we propose a modular way of defining metamodel extensions to capture quality attributes. Model transformations are then executed at runtime to process these extensions and connect to off-the-shelf verification engines. The result of the verification is then fed back into the decision engine to help it choose the right reconfiguration model, before it is enacted on the component platform level. Of course one challenge is that both the transformation machinery and the verification engine are fast enough to be used at runtime.

5.15 Can RE Contribute to SAS Assurance?

Zhi Jin (Peking University, CN)

When systems will run in open, changing, uncertain execution and/or interaction environments, self-adaptation becomes a must-be requirement. That is the expectation to the systems to cope with variable resources and variable interactors that may cause system errors, while maintaining the business goals envisioned by the engineers and expected from the users. The task of the requirement phase is to identify such a requirement and explore the requirement into sufficient detailed specification that is ready for system design. This talk proposes an environment-based methodologies for engineering the self- adaptation requirements, that is the first step assurance. The following are the three features: (1) Identifying the requirement should start from the environment conceptualisation that allowing to identifying the uncertainty and changing patterns of the environment; (2) Modelling the system-to-be as discrete control system which captures the capabilities of monitoring and detecting the potential changes in both environment and system itself; (3) Defining precisely the specification of the run-time mechanism for allowing the system to be self- adaptive.

5.16 Self-adaptivity vs. Latent Software Defects: Software Health Management

Gabor Karsai (Vanderbilt University, US)

The complexity of software systems is reaching the point where latent defects remain in deployed systems. To detect, isolate, and mitigate the effects of such defects new paradigms are needed. One such paradigm is software health management that is interesting application area of self- adaptive software techniques. Software health management borrows the language and techniques from system health management (frequently used in aerospace vehicles), where anomaly detection, fault diagnostics, and reconfiguration are used to remove faults from systems while in operation. The talk describes a software framework that was created to enable developers to design and implement software health management functions in their systems.The framework is based on a software component model, local and global health managers. The latter incorporates a system-wide fault diagnostics component as well as a reasoner engine that computes architectural adaptations of the component architecture to mitigate the effect of component faults. The framework is supported by a model-driven development toolchain.

5.17 Challenges in Autonomous Vehicle Validation

Philip Koopman (Carnegie Mellon University, US)

Creating safe Transportation Cyber-Physical Systems (CPSs) presents new challenges as autonomous operation is attempted in unconstrained operational environments. The extremely high safety level required of such systems (perhaps one critical failure per billion operating hours) means that validation approaches will need to consider not only normal operation, but also operation with system faults and in exceptional environments. Additional challenges will need to be overcome in the areas of rigorously defining safety requirements, trusting the safety of multi-vendor distributed system components, tolerating environmental uncertainty, providing a realistic role for human oversight, and ensuring sufficiently rigorous validation of autonomy technology.

5.18 Control-theoretical computing system design

Alberto Leva (Technical University of Milan, IT)

In the last years, we have been addressing several problems related to computing systems by adopting a control-theoretical approach. With respect to the mainstream research on the matter, our work has a relevant peculiarity: instead of taking the system "as is" and just adding a control layer on top of it, we try to isolate the parts of the overall problem that can be formally stated as control ones, and design parts of the considered system consequently. To explain with an example relative to task scheduling, we do not take an existing and fully functional scheduler and use a feedback controller to adapt its tuning parameters (e.g., priorities). On the contrary, we build the entire scheduler as a controller, that directly decides the CPU times to be allotted to the tasks. Applying this idea to different contexts, ranging from time synchronisation to service composition and dynamic binding, we found that isolating the "core" problems – in the sense above – tends to allow for the application of simple modelling and control methods. We also noticed that once formalised this way, many heterogeneous problems come to assume a significantly uniform mathematical structure. We finally observed that some properties of interest for control systems are potentially very keen to be re-formulated as "assurances" in the computing systems sense. As a result, besides favouring efficiency, the proposed approach also eases the formal assessment of some properties of interest. We therefore conjecture that by setting up (wherever possible and convenient) a sound correspondence between said properties – that are natively expressed in control-theoretical terms – and assurances in the computing system sense, significant benefits can be achieved.

5.19 Adaptation in Software Defined Infrastructures

Marin Litoiu (York University – Toronto, CA)

To be efficient and robust, an adaptive feedback loop has to aggregate the contributions of different software layers. For example, the performance of an application can be controlled not only by tuning the parameters of the application but also by adjusting the underlying computing, storage, network or hardware infrastructures. With the advent of Software Defined Infrastructure, those adjustment can be effected at runtime. In this presentation, we introduce the Smart Applications on Virtual Infrastructure (SAVI), a Canadian project that builds an experimental Software Defined Infrastructure testbed. As an example of problems that we can solve more efficiently with the cross layer approach are the Low and Slow Distributed Denial of Service attacks. Those attacks are becoming a serious issue because, due to low resource consumption and slow ramping, they are hard to detect. A possible solution is to identify the attack at the application layer and then to use the underlying layers to mitigate the attack.

5.20 Toward the Making of Software that Learns to Manage Itself

Sam Malek (George Mason University – Fairfax, US)

A self-managing software system is capable of adjusting its behavior at runtime in response to changes in the system, its requirements, or the environment in which it executes. Self-management capabilities are sought-after to automate the management of complex software in many computing domains, including service-oriented, mobile, cyber-physical and ubiquitous settings. While the benefits of such software are plenty, its development has shown to be much more challenging than the conventional software.

At the state of the art, it is not an impervious engineering problem in principle to develop a self-adaptation solution tailored to a given system, which can respond to a bounded set of conditions that are expected to require automated adaptation. However, any sufficiently complex software system once deployed in the field is subject to a broad range of conditions and many diverse stimuli. That may lead to the occurrence of behavioral patterns that have not been foreseen previously: in fact, those may be the ones that cause the most critical problems, since, by definition, they have not manifested themselves, and have not been accounted for during the previous phases of the engineering process. A truly self-managing system should be able to cope with such unexpected behaviors, by modifying or enriching its adaptation logic and provisions accordingly.

In this talk, I will first provide an introduction to some of the challenges of making software systems self-managing. Afterwards, I will provide an overview of two research projects in my group that have tackled these challenges through the applications of automated inference techniques (e.g., machine learning, data mining). The results have been promising, allowing the software engineers to empower a software system with advanced self-management capabilities with minimal effort. I will conclude the talk with an outline of future research agenda for the community.

5.21 On the uncertainties in the modeling of self-adaptive systems

Raffaela Mirandola (Technical University of Milan, IT)

The complexity of modern software systems has grown enormously in the past years with users always demanding for new features and better quality of service. The satisfaction of non-functional requirements like performance and availability is of paramount importance if a product hopes to be considered in the marketplace. Model-based evaluation techniques at software design-time have been proposed to ensure the delivering of software that meets its non-functional requirements. However, since a large part of modern software is embedded in dynamic execution contexts where requirements, environment assumptions, and usage profiles continuously change, this quality assessment at design time becomes more difficult. As an answer to dynamic execution context, self-adaptive systems have been adopted. Self-adaptation endows a system with the capability to accommodate its execution to different contexts in order to achieve continuous satisfaction of requirements. Often, self- adaptation process also makes use of runtime model evaluations to decide the changes in the system. However, even at runtime, context information that can be managed by the system is not complete or accurate; i.e, it is still subject to some uncertainties. This work motivates the need for the consideration of the concept of uncertainty in the model-based evaluation as a primary actor, classifies the avowed uncertainties of self-adaptive systems, and illustrates examples of how different types of uncertainties are present in the modeling of system characteristics for availability requirement satisfaction.

5.22 A Software Lifecycle Process For Data-intensive Self-adaptive Systems

Marco Mori (University of Namur, BE)

Joint work of Cleve, Anthoy; Inverardi, Paola; Mori, Marco

Nowadays ubiquitous software systems have to meet user expectations while considering an ever-changing environment. The increasing space of possible contexts and the limited capacity of mobile devices make no longer possible to incorporate all necessary software alternatives and the required data for all possible contexts. Thus, upon variations to user task, user role, user preferences or physical environment, the current software alternative and its required data have to be reconfigured. In order to prevent incorrect system behaviours, reconfigurations should avoid inconsistencies of both data and software by providing assurance to the uncertainty of changes affecting the system.

In this talk we present a generic lifecycle process for self-adaptive systems which supports predictable and unpredictable system evolutions and different notions of inconsistencies which apply to code artifacts and system requirements. Consequently we introduce the problem of supporting adaptivity of data belonging to a global database and we improve the former definition of assurance by considering consistency of database schema and instances. We claim the need for a new lifecycle process which has to consider together adaptivity of software with adaptivity of data occurring in a consistent predictable and unpredictable manner.

5.23 Managing Viability Zone Dynamics for the Assurance of Self-Adaptive Systems

Hausi A. Mueller (University of Victoria, CA)

We define the viability zone of a self-adaptive software (SAS) system as the set of possible states in which the system operation is not compromised, that is, the set of states where the system's requirements and desired properties (i.e., adaptation goals) are satisfied. Viability zones can be characterized in terms of relevant context attributes and corresponding desired values. These context attributes correspond to either measurements of internal variables of the target system or the adaptation mechanism, or environmental variables whose variations can take the system outside its viability zone. Viability zones are N-dimensional. Therefore, a particular SAS system may have more than one associated viability zone (e.g., one for each adaptation goal). The global viability zone of a system thus results from the composition of these partial viability zones. Moreover, existing viability zones can be added, replaced or adjusted by adding or removing variables of interest at runtime. Viability zones can change with context changes, as opposed to the solution space concept, which is assumed to be fixed. In effect, the viability zone of a target system under adaptation constantly varies along adaptation dimensions. These variations take place every time the adaptation operation modifies either the target system architecture (e.g., adding or removing components and connectors) or the controller itself (e.g., modifying its parameters or replacing the control algorithm), thus introducing new, or removing existing variables and associated domain types. To extend the V&V coverage of the expanded viability zone, runtime models are required for the incremental derivation of software artifacts for V&V monitoring and checking. Therefore, not only are runtime V&V methods required to cope with the viability zone dynamics problem, but these V&V methods also need to be automatically generated according to the modifications that result from dynamic adaptation to keep the adaptive system inside its viability zone. We believe that managing viability zones at runtime is crucial for the assurance of self-adaptive systems.

5.24 Self-Adaptive Cloud Controllers

Mauro Pezze (University of Lugano, CH)

Cloud technologies are rapidly substituting classic computing solutions and challenge the community with new problems. In our research we focus on cloud controllers and we work on novel solutions for self-adaptive cloud controllers based on Kriging models. While in classic software engineering solutions, scheduling problems are mostly hidden from the application developers, in Cloud based applications the responsibility of allocating the required resources is assigned to the developers and depends on the application requirements and the nature of the cloud. General-purpose Cloud schedulers provide sub-optimal solutions to the problem with respect to application-specific solutions that we call cloud controllers. We are investigating the use of surrogate models, and in particular Kriging models, that present interesting properties to support adaptive control.

5.25 Feedbacks Control Loops as 1st Class Entities – The SALTY Experiment

Romain Rouvoy (Université de Lille I, FR)

Joint work of Collet, Philippe; Krikava, Filip; Rouvoy, Romain; Seinturier, Lionel
Main reference F. Krikava, P. Collet, R. France, "ACTRESS: Domain-Specific Modeling of Self-Adaptive Software Architectures," in Proc. of the 9th DADS Track of the 29th ACM Symp. on Applied Computing, ACM, 2014.
URL http://salty.unice.fr/

This talk shortly reports on the results of the SALTY R&D project (https://salty.unice.fr) funded by the French funding agency (ANR). The key outcome of this project is a software framework that covers both design-time and runtime support for integrating self-adaptive behaviours into potentially complex legacy systems. SALTY therefore provides a reflective domain-specific model to externalise and make explicit the control layer of legacy systems. While the adoption of a domain-specific model leverages the mapping on different middleware stacks (FraSCAti or Akka in our case studies), it also acts as a pivot, within a modular toolchain, to implement design-time verifications and to inject runtime guards. Future case studies of this approach will cover green computing, crowd- sensing, and big data systems.

5.26 Model-driven infrastructure for reliable service compositions using dynamic software product lines

Cecilia Mary Fischer Rubira (UNICAMP, BR)

Joint work of Rubira, Cecilia Mary Fischer; Nascimento; Amanda S. Castor; Fernando
Main reference A. Nascimento, C. Rubira, F. Castor, "ArCMAPE: A Software Product Line Infrastructure to Support Fault-Tolerant Composite Services," in Proc. of the 15th IEEE Int'l Symposium on High Assurance Systems Engineering (HASE'14), pp. 41–48, IEEE CS, 2014.
URL http://doi.ieeecomputersociety.org/10.1109/HASE.2014.15

A number of solutions use software fault tolerance techniques based on design diversity to create fault-tolerant composite services that leverage functionally equivalent services. Nevertheless, these solutions are not able to adapt themselves at runtime to cope with dynamic changes of user requirements and fluctuations in the quality of services (QoS). We propose a self-adaptive solution, called ArCMAPE, that leverages ideas from Software Product Line Engineering to support fault-tolerant composite services. In particular, we specify a feature model and product line architecture to capture the common and variable features among a number of software fault tolerance techniques based on design diversity. ArCMAPE provides software com- ponents implementing the common features; and a foundation on which plug-in components, or variable components, can be easily added to realise the target variable features. At runtime, ArCMAPE dynamically instantiates software fault tolerance techniques tailored to the specific needs of different clients and contexts by employing feature-based runtime adaptations. Outcomes obtained from an empirical study suggest the feasibility and efficiency of our solution to support self-adaptive, fault- tolerant composite services. We discuss the obtained outcomes and present directions for future work.

5.27 Modular Discrete Control for Adaptive Software Systems

Eric Rutten (INRIA Grenoble – Rhône-Alpes, FR)

This talk will present a synthesis on our work on safe design of controllers for autonomic computing systems, using techniques originally conceived in Control thoery, more specifically Discrete event Systems. Our approach is supported by a programming language from the family of reactive languages, and based on a formalism of Labelled Transition Systems. Applications concern control of logical and synchronization aspects in reconfigurable FPGA-based architectures, coordination of multiple loops in Data-center management, and smart-environments in the Internet of Things.

5.28 Managing Non-Functional Uncertainty via Model-Driven Adaptivity

Giordano Tamburrelli (University of Lugano, CH)

Modern software systems are often characterized by uncertainty and changes in the environment in which they are embedded. Hence, they must be designed as adaptive systems. We propose a framework that supports adaptation to non-functional manifestations of uncertainty. Our framework allows engineers to derive, from an initial model of the system, a finite state automaton augmented with probabilities. The system is then executed by an interpreter that navigates the automaton and invokes the component implementations associated to the states it traverses. The interpreter adapts the execution by choosing among alternative possible paths of the automaton in order to maximize the system's ability to meet its non-functional requirements. To demonstrate the adaptation capabilities of the proposed approach we implemented an adaptive application inspired by an existing worldwide distributed mobile application and we discussed several adaptation scenarios.

5.29 Managing Viability Zone Dynamics for the Assurance of Self-Adaptive Systems

Gabriel Tamura (Universidad Icesi, CO)

We define the viability zone of a self-adaptive software (SAS) system as the set of possible states in which the system operation is not compromised, that is, the set of states where the system's requirements and desired properties (i.e., adaptation goals) are satisfied. Viability zones can be characterized in terms of relevant context attributes and corresponding desired values. These context attributes correspond to either measurements of internal variables of the target system or the adaptation mechanism, or environmental variables whose variations can take the system outside its viability zone. Viability zones are N-dimensional. Therefore, a particular SAS system may have more than one associated viability zone (e.g., one for each adaptation goal). The global viability zone of a system thus results from the composition of these partial viability zones. Moreover, existing viability zones can be added, replaced or adjusted by adding or removing variables of interest at runtime.

Viability zones can change with context changes, as opposed to the solution space concept, which is assumed to be fixed. In effect, the viability zone of a target system under adaptation constantly varies along adaptation dimensions. These variations take place every time the adaptation operation modifies either the target system architecture (e.g., adding or removing components and connectors) or the controller itself (e.g., modifying its parameters or replacing the control algorithm), thus introducing new, or removing existing variables and associated domain types. To extend the V&V coverage of the expanded viability zone, runtime models are required for the incremental derivation of software artifacts for V&V monitoring and checking. Therefore, not only are runtime V&V methods required to cope with the viability zone dynamics problem, but these V&V methods also need to be automatically generated according to the modifications that result from dynamic adaptation to keep the adaptive system inside its viability zone. We believe that managing viability zones at runtime is crucial for the assurance of self-adaptive systems.

5.30 Managing Viability Zone Dynamics for the Assurance of Self-Adaptive Systems

Norha Milena Villegas Machado (Universidad Icesi, CO)

We define the viability zone of a self-adaptive software (SAS) system as the set of possible states in which the system operation is not compromised, that is, the set of states where the system's requirements and desired properties (i.e., adaptation goals) are satisfied. Viability zones can be characterized in terms of relevant context attributes and corresponding desired values. These context attributes correspond to either measurements of internal variables of the target system or the adaptation mechanism, or environmental variables whose variations can take the system outside its viability zone. Viability zones are N-dimensional. Therefore, a particular SAS system may have more than one associated viability zone (e.g., one for each adaptation goal). The global viability zone of a system thus results from the composition of these partial viability zones. Moreover, existing viability zones can be added, replaced or adjusted by adding or removing variables of interest at runtime.

Viability zones can change with context changes, as opposed to the solution space concept, which is assumed to be fixed. In effect, the viability zone of a target system under adaptation constantly varies along adaptation dimensions. These variations take place every time the adaptation operation modifies either the target system architecture (e.g., adding or removing components and connectors) or the controller itself (e.g., modifying its parameters or replacing the control algorithm), thus introducing new, or removing existing variables and associated domain types. To extend the V&V coverage of the expanded viability zone, runtime models are required for the incremental derivation of software artifacts for V&V monitoring and checking. Therefore, not only are runtime V&V methods required to cope with the viability zone dynamics problem, but these V&V methods also need to be automatically generated according to the modifications that result from dynamic adaptation to keep the adaptive system inside its viability zone. We believe that managing viability zones at runtime is crucial for the assurance of self-adaptive systems.

5.31 Modeling Self-Adaptive Software

Thomas Vogel (Hasso-Plattner-Institut – Potsdam, DE)

Self-adaptive software typically uses a self-representation to reflect on its adaptable parts. While runtime models are employed for such a self-representation, runtime models and model-driven engineering methods may additionally be employed to specify, execute, and adjust (on-line and off-line) the individual adaptation activities of a feedback loop as well as the whole feedback loop. Besides leveraging flexibility, keeping models alive at runtime and considering them as first class entities support the provisioning of assurances for on-line and off-line adaptation based on such models. Therefore, a formal underpinning of modeling languages and models, the composition and decomposition of models and assurances, and assurances for the models themselves are critical aspects that, among others, have to be addressed.

5.32 ActivFORMS: Active FORmal Model for Self-adaptation

Danny Weyns (Linnaeus University – Växjö, SE)

Self-adaptation enables a software system to adapt itself at runtime to deal with uncertainties, such as dynamic operating conditions that were difficult to predict or unanticipated changes of goals. Self-adaptation is realized with a feedback loop, which typically consists of monitor, analysis, plan, and execution functions. To provide guarantees of the adaptations, state of the art approaches propose to equip the feedback loop with formal models of the managed system, its environment and goals. However, existing approaches do not systematically formalize and verify the behavior of the adaptation functions themselves. Furthermore, there is limited attention for adaptation of unanticipated changes. We propose ActivFORMS (Active FORmal Model for Self-adaptation) that uses a formal model of the complete feedback loop. This model is directly executed at runtime by a virtual machine realizing adaptation. ActivFORMS assures that the verified system goals are met at runtime, and the approach enables dynamic updates of self-adaption behaviors to support unanticipated changes.

5.33 Reconciling self-adaptation and self-organization towards effective assurances

Franco Zambonelli (University of Modena, IT)

Two complementary software engineering approaches currently exist to make complex software systems adaptive. Self-adaptation approaches attack the problem by engineering proper feedback loops around components and systems, so as to make them adaptive by explicit design. Self-organization approaches, on the other hand, attack the problems by trying to mimic in software the capabilities of collective adaptation of natural systems, so as to make systems adaptive by emergence. A key challenge is that, while self-organization can be much more effective, assurances can be better achieved via self-adaptation. Accordingly, I analyze the issue of reconciling the two approaches towards a novel, innovative, approach that to synthesize the benefits of both approaches, efficiency and assurance.

5.34 Runtime Testing of Self-Adaptive Systems

Carlos Eduardo da Silva (Federal University of Rio Grande do Norte, BR)

In our previous work we have tackled the problem of dynamically generating adaptation plans in order to deal with situation not foreseen during development. We have developed a framework for the dynamic generation of processes that factors out common process generation mechanisms and provides explicit customisation points to tailor process generation capabilities to different application domains. Such framework has been focused on the planning aspects of the MAPE-K loop, and has benn applied in two different domains, namely, architectural reconfiguration and integration testing. We have also considered the problem of dynamic modifying the mechanisms responsible for generating processes.

Currently, we have been looking further on the aspects related to testing at run-time as the means of providing assurances about self-adaptive systems. We have also been looking on "feature phases", that is, MAPE-K loops that deals with different features (concerns) of each of the phases of the main MAPE-K loops, investigating patterns of interaction between MAPE-K loops and their phases as the means to position run-time testing in the MAPE-K loop.

5.35 Architecting Resilience: Handling Malicious and Accidental Threats

Rogerio de Lemos (University of Kent, GB)

Resilience is the persistence of service delivery that can justifiably be trusted, when facing changes. While architecting is the art and science of creating and building complex systems, and which covers the following basic activities: scope, structure and certification. One important aspect of resilience is the provision of assurances, and these are obtained by building arguments about system resilience. However in order to build arguments, one needs to collect, structure and analyse evidence. In self-adaptive systems, evidence can be obtained either at development-time or run-time.

This talk has covered three contributions. In the first contribution, we describe how for self-adaptive software systems integration testing can be performed at run-time. This activity should be implemented as a feedback control loop, which should be associated with the analysis phase of the autonomic MAPE-K loop.

The second contribution is related to a stepwise progress for the provision of assurances about the resilience of self-adaptive software systems, and it covers the following topics: (i) resilience evaluation based on environmental stimuli in which probabilistic model-checking is used for obtaining levels of confidence, (ii) resilience evaluation by comparing adaptation mechanisms of self-adaptive software systems, (iii) robustness evaluation of controllers by injecting faults into the probes of Rainbow, (iv) effectiveness of architecture-based self-adaptation by evaluating the effort of evolving industrial middleware into a architectural-based self-adaptive software system, finally (v) robustness-driven resilience evaluation of self-adaptive software systems in which system properties are evaluated by injecting faults.

The third contribution concerns an approach based on self-adaptation as a means to improve the management of malicious behaviour, by adapting authoris ation policies and access rights. The goal is to adapt to mitigate malicious behaviour, and prevent future attacks.

References

1 J. Camara, R. de Lemos, M. Vieira, R. Almeida, R. Ventura. Architecture-Based Resilience Evaluation for Self-Adaptive Systems. Computing Journal (Special "Software Architecture for Code Testing and Analysis") 95(8). 2013. pp. 689–722.

2 J. Camara, R. de Lemos. Evaluation of resilience in self-adaptive systems using probabilistic model-checking. Proceedings of the International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2012). Zurich, Switzerland. June 2012. pp. 53–62.

3 J. Camara, P. Correia, R. de Lemos, D. Garlan, P. Gomes, B. Schmerl, R. Ventura. Evolving an Adaptive Industrial Software System to Use Architecture-Based Self-Adaptation. Proceedings of the International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2013). San Francisco, CA, USA. May 2013. pp. 13–22.

4 J. Camara, R. de Lemos, N. Laranjeiro, R. Ventura, M. Vieira. Robustness Evaluation in Self-Adaptive Software Systems. Latin American Symposium on Dependable Computing (LADC 2013). Rio de Janeiro, RJ, Brazil. April 2013. pp. 1–10.

5 C. Bailey, D. W. Chadwick, R. de Lemos, K. W. S. Sui. Enabling the Autonomic Management of Federated Identity Providers. 7th International Conference on. Autonomous Infrastructure, Management and Security (AIMS 2013). June 2013, UPC Barcelona, Spain. 2013. pp. 100–111.

6 C. E. da Silva and R. de Lemos. Dynamic plans for integration testing of self-adaptive software systems. Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2011). Honolulu, HI, USA. May 2011. pp. 148–157.

Participants

- Jesper Andersson
Linnaeus University – Växjö, SE
- Chris Bailey
University of Kent, GB
- Luciano Baresi
Technical University of Milan, IT
- Benoit Baudry
INRIA Rennes – Bretagne Atlantique, FR
- Nelly Bencomo
Aston Univ. – Birmingham, GB
- Yuriy Brun
University of Massachusetts – Amherst, US
- Radu Calinescu
University of York, GB
- Javier Cámara
Carnegie Mellon University, US
- Myra B. Cohen
Univ. of Nebraska – Lincoln, US
- Vittorio Cortellessa
University of L'Aquila, IT
- Bojan Cukic
West Virginia University – Morgantown, US
- Carlos Eduardo da Silva
Federal University of Rio Grande do Norte, BR
- Rogerio de Lemos
University of Kent, GB
- Antonio Filieri
Universität Stuttgart, DE
- Carlo Ghezzi
Technical University of Milan, IT
- Holger Giese
Hasso-Plattner-Institut – Potsdam, DE
- Alessandra Gorla
Universität des Saarlandes – Saarbrücken, DE
- Vincenzo Grassi
University of Rome “Tor Vergata”, IT
- Lars Grunske
Universität Stuttgart, DE
- Paola Inverardi
Univ. degli Studi di L'Aquila, IT
- Jean-Marc Jezequel
University of Rennes, FR
- Zhi Jin
Peking University, CN
- Gabor Karsai
Vanderbilt University, US
- Philip Koopman
Carnegie Mellon University, US
- Seok-Won Lee
Ajou University, KR
- Alberto Leva
Technical University of Milan, IT
- Marin Litoiu
York University – Toronto, CA
- Sam Malek
George Mason University – Fairfax, US
- Raffaela Mirandola
Technical University of Milan, IT
- Marco Mori
University of Namur, BE
- Hausi A. Müller
University of Victoria, CA
- Mauro Pezzè
University of Lugano, CH
- Romain Rouvoy
Université de Lille I, FR
- Cecilia Mary Fischer Rubira
UNICAMP, BR
- Eric Rutten
INRIA Grenoble – Rhône-Alpes, FR
- Bradley Schmerl
Carnegie Mellon University, US
- Mary Shaw
Carnegie Mellon University, US
- Giordano Tamburrelli
University of Lugano, CH
- Gabriel Tamura
Universidad Icesi, CO
- Norha Milena Villegas Machado
Universidad Icesi, CO
- Thomas Vogel
Hasso-Plattner-Institut – Potsdam, DE
- Danny Weyns
Linnaeus University – Växjö, SE
- Franco Zambonelli
University of Modena, IT

Report from Dagstuhl Seminar 13512

Social Issues in Computational Transportation Science

Edited by

Glenn Geers[1], Monika Sester[2], Stephan Winter[3], and Ouri E. Wolfson[4]

1 **NICTA – Kensington, AU,** Glenn.Geers@nicta.com.au
2 **Leibniz Universität Hannover, DE,** monika.sester@ikg.uni-hannover.de
3 **The University of Melbourne, AU,** winter@unimelb.edu.au
4 **University of Chicago, US,** wolfson@cs.uic.edu

Abstract

The Dagstuhl Seminar, "Social Issues in Computational Transportation Science" (13512) took place from 15 to 19 December 2103, attracting 27 participants active in a wide range of academic, commercial, and public sector areas. CTS is an emerging discipline that combines advances in computer science and engineering with the modeling, planning, social, and economic aspects of transportation in order to improve the safety, mobility, and sustainability of transportation systems. The aim of this seminar was to focus on the social computing aspects of CTS, including such areas as social networks and crowd-sourcing for transportation, as well as the integration of persuasive technologies and behavioral economics in social computing. In their time at the workshop, participants discussed and debated these and other topics, as shown in the workshop's summary report.

Seminar 15.–19. December, 2013 – www.dagstuhl.de/13512
1998 ACM Subject Classification K.4 Computers and Society
Keywords and phrases Computational Transportation Science
Digital Object Identifier 10.4230/DagRep.3.12.97
Edited in cooperation with Caitlin Cottrill

1 Executive Summary

Glenn Geers
Monika Sester
Stephan Winter
Ouri E. Wolfson

The Dagstuhl Seminar "Social Issues in Computational Transportation Science" (13512) brought together researchers working in various areas contributing to Computational Transportation Science (CTS). CTS is an emerging discipline that combines computer science and engineering with the modeling, planning, social, and economic aspects of transportation. It is the discipline behind intelligent transportation systems (ITS), i.e., emerging from the convergence of ICT and transportation. The discipline studies how to improve the safety, mobility, and sustainability of transportation systems by taking advantage of information technologies and ubiquitous computing.

After a first Dagstuhl Seminar on CTS in 2010 (10121), in this seminar we focused on the social computing aspect of CTS, reflecting on the potential of many recent developments in transportation, such as social networks, crowdsourcing of spatial data, persuasive technologies, and behavioural economics in social computing.

In fact, the seminar (which was a day shorter because of Christmas) consisted of three parts: a number of tutorials and short talks, a competition for the best application challenge in CTS, and a joint sketch of an introductory course on CTS. An excursion to the Christmas Market in Trier rounded up the week.

The CTS application challenge was inspired by above mentioned social aspects, such as incentives to change travel behaviour, data integration / analytics required to feed these incentives, multimodal integrated door-to-door travel, autonomous vehicles, automated crowdsourcing for travel statistics, or smart solutions for the parking problem. In case you are curious which team won the best proposal award, their proposal is online[1].

The sketch of an introductory course on CTS clearly profited from the broad variety of expertise at the seminar. Everybody was learning from the sketches of modules contributed by others, to a degree that we all wished we could take this course in full length.

Overall, this report collects material that wants to be taken into practice. We hope that we inspire teams all over the world to contribute ICT expertise for more sustainable mobility choices, and perhaps add to the development of curricula in this area.

[1] https://sites.google.com/site/karmobility/home

2 Table of Contents

3 Overview of Talks

3.1 Informed Rural Passenger

Caitlin Cottrill (University of Aberdeen, GB)

Joint work of Edwards, Pete; Nelson, John; Sripada, Yaji; Pan, Jeff; Cottrill, Caitlin; Beecroft, Mark; Corsar, David; Baillie, Christopher; Markovic, Milan; Papangelis, Konstantinos
Main reference N. R. Velaga, M. Beecroft, J. D. Nelson, D. Corsar, P. Edwards, "Transport poverty meets the digital divide: accessibility and connectivity in rural communities," Journal of Transport Geography, Vol. 21, March 2012, pp. 102–112, 2012.
URL http://dx.doi.org/10.1016/j.jtrangeo.2011.12.005

The Informed Rural Passenger project aims to address problems of inadequate information sources for public transport in rural areas via the use of a smartphone app ('GetThere Bus') and crowdsourced information. Our research indicates that passengers desire real-time information that is accurate, timely, and personalised, and informs travellers about potential disruptions and transport alternatives. To address these needs via the app, we have worked to develop an ecosystem whereby travellers may both send and receive information on bus schedules and delays, with visual representations of information quality and provenance. Our initial research has resulted in a demonstration app and pilot test, which indicated that more work was needed to address alternate methods of information dissemination in rural areas. This finding has informed our current and planned research on the use of text messages for information provision, as well as more systematic use of social media.

3.2 Urban Informatics at NICTA

Glenn Geers (NICTA – Kensington, AU)

Joint work of Geers, Glenn; Cai, Chen; Economou. Dean; Liu, Wei; Tyler, Paul

In the talk I will give a brief overview of some work that is currently being carried out at NICTA. It asks more questions than it answers.

1. Co-operative Intelligent Transport Initiative This project aims to build the world's first heavy goods vehicle specific V2X test-bed on a 42 km route located south of the City of Sydney, Australia. Phase 1 is due for completion mid-2014 and will comprise thirty instrumented trucks, three connected signalised junctions and several portable roadside DSRC systems. An additional aim is to make as much data available to the research community as possible. The project is an initiative of the Centre for Road Safety which is part of Transport for New South Wales.
   Phase 2 of the CITI Project (subject to funding) will add various road-side sensors and integrate light vehicles and pedestrians into the connected system. It is hoped to run the site as a managed open research facility.
2. Data Driven Traffic Modelling This project is developing methods of applying machine learning to the road network. Is it possible to predict the location of potential black spots from traffic data? Can incident duration be predicted and reduced? How can data driven models be linked to more traditional micro- or meso-scale simulations?

There are challenging issues for machine learning: how can new data types be added without having to completely re-learn the model (transfer learning). How can the large spatial and temporal scale issues (intersection to city-wide; seconds to years) be handled?

3. Computable Liveability Every year many organisations produce and publish 'Liveability' Indices for cities around the world. However, there is no agreed definition of liveability and therefore no way to compare indices produced by different organisations. Most measures rate Western cities highly. Why?
   Is it possible to go from survey data to physical data? Does building a new hospital compensate for increasing traffic on the roads? Can we derive a personal liveability index based on measured data?

3.3 Towards Intelligent and Generic LBS for Drivers and Mobile Users

Sergio Ilarri (University of Zaragoza, ES)

In this talk I will focus on Location-Based Services (LBS) for hybrid networks composed of both vehicles and mobile users. The motivation is the interest of studying data management solutions that take into account a generic environment where different types of moving objects share different types of data and possibly using different communication technologies (ad hoc wireless communications forming a pure mobile P2P network, hybrid mobile P2P network with support infrastructure nodes, wide-area communications like 3G, etc.).

I will start by summarizing some data management challenges for vehicular networks, related to the exchange of events (efficient and effective content-based data dissemination for push-based data access), query processing (pull-based data access by using query dissemination or mobile agent technology), data item relevance evaluation, management of information about scarce resources (like available parking spaces or charge stations for electric vehicles), semantic data management, automatic knowledge extraction from the data items, multimedia data management, incentives, and trust. Then, I will show some use cases that exploit sensors embedded in moving objects to obtain interesting information (environment monitoring and multimedia data). Afterwards, I will emphasize the role that semantic technologies can play in this context and the benefits that they can provide as facilitators for the development of intelligent and generic LBS. Finally, I will present as an example the basics of our current prototype SHERLOCK (System for Heterogeneous mobilE Requests by Leveraging Ontological and Contextual Knowledge), which exploits shared knowledge about different types of objects and services (encoded in ontologies) to offer interesting services and information to mobile users.

References

1 Thierry Delot and Sergio Ilarri, "Data Gathering in Vehicular Networks: The VESPA Experience (Invited Paper)", Fifth IEEE Workshop On User MObility and VEhicular Networks (LCN ON-MOVE 2011), in conjunction with the 36th IEEE Conference on Local Computer Networks (LCN 2011), Bonn (Germany), IEEE Computer Society, ISBN 978-1-61284-927-0, pp. 801–808, October 2011.

2 Sergio Ilarri, Eduardo Mena and Arantza Illarramendi, "Location-Dependent Queries in Mobile Contexts: Distributed Processing Using Mobile Agents", IEEE Transactions on Mobile Computing, ISSN 1536-1233, 5(8): 1029–1043, IEEE Computer Society, August 2006.

3 Sergio Ilarri, Eduardo Mena and Arantza Illarramendi, "Location-Dependent Query Processing: Where We Are and Where We Are Heading", ACM Computing Surveys, ISSN 0360-0300, 42(3): 12:1–12:73, ACM Press, March 2010.
4 Sergio Ilarri, Arantza Illarramendi, Eduardo Mena, Amit Sheth, "Semantics in Location-Based Services – Guest Editors' Introduction for Special Issue", IEEE Internet Computing, ISSN 1089-7801, 15(6):10–14, IEEE Computer Society, November/December, 2011.
5 Sergio Ilarri, Dragan Stojanovic and Cyril Ray, "Semantic Management of Moving Objects: A MOVEment Towards Better Semantics", COST IC0903 MOVE Final Conference, Vienna (Austria), 2 pp., September/October 2013.
6 Raquel Trillo, Sergio Ilarri and Eduardo Mena, "Comparison and Performance Evaluation of Mobile Agent Platforms", Third International Conference on Autonomic and Autonomous Systems (ICAS'07), Athens (Greece), IEEE Computer Society, ISBN 978-0-7695-2859-5, pp. 41–46, June 2007.
7 Óscar Urra, Sergio Ilarri, Raquel Trillo and Eduardo Mena, "Mobile Software Agents for Mobile Applications", in Handbook of Research on Mobile Software Engineering: Design, Implementation and Emergent Applications, Paulo Alencar and Donald Cowan (ed.), IGI Global, ISBN 978-1615206551, ISBN13: 9781615206551, ISBN10: 1615206558, EISBN13: 9781615206568, pp. 725–740, May 2012.
8 Óscar Urra and Sergio Ilarri, "Using Mobile Agents in Vehicular Networks for Data Processing", 14th International Conference on Mobile Data Management (MDM 2013), Ph.D. Forum, Milan (Italy), IEEE Computer Society, ISBN 978-0-7695-4973-6, volume 2, pp. 11–14, June 2013.
9 Roberto Yus, Eduardo Mena, Sergio Ilarri and Arantza Illarramendi, "SHERLOCK: Semantic Management of Location-Based Services in Wireless Environments", Pervasive and Mobile Computing journal, Special Issue on Information Management in Mobile Applications, ISSN 1574-1192, to appear, DOI: 10.1016/j.pmcj.2013.07.018.

3.4 Issues in Agent-based Route Choice Models

Franziska Klügl (Örebro University, SE)

Agent-based simulation forms one of the most prominent microscopic simulation paradigms. It is applied basically in all areas in which actors situated and interacting in an environment are to be modeled and analyzed. It is best characterized as "generative" simulation as the overall system properties and behavior are not merely described, but generated from lower level agent behavior and interaction. Agent-based simulation promises to solve many problematic issues of modeling in general, ranging from the possibility to formulate heterogeneity on various levels to integration of individual-level adaptation and population-level evolution that allows the simulation of self-organization and generative analysis of emergent phenomena.

Meanwhile, agent-based simulation plays an important role in traffic simulation: activity-based approaches for travel demand modeling as well as advance routing and mobility simulations (for a review see [1]). Modeling and simulation of route choice plays hereby a prominent role, as the problem can be mapped to gametheoretic scenarios such as the El-Farol Bar Problem or Minority Games. These scenarios can even be approached by experimentation for analyzing human decision making [2]: Classically, there are two options (to go to the bar or to stay at home, go on highway or country road, stay on route or change to alternative) between which each agent has to choose. The agents choosing the less crowded alternative receive the higher reward and based on that adapt their decision making. Over the years, many studies also with related problems (e.g. the Braess Paradox) have been conducted, mainly focusing on the question whether and how information on previous choices or predictions of load influence the overall outcome of equilibriums (for example [4]). These scenarios were intended to analyze human decision making in traffic situation, so an attempt has been made to scale them to real-world networks [3]. The main results of these experiments also involving information given at different locations in the network were that the scaling up to more realistic problems is not trivial with respect to:

- Generation of options to choose: It is hard to generate a set of independent routes. This is a well-known problem in discrete choice modeling.
- Integration of feedback. If there are no full routes as options, the agents cannot assign feedback in form of overall travel time to routes, rather than to links. This feedback can be used to build a belief set of traveltimes on links that can be used in shortest path algorithm. So, the general decision making needs to be adapted to more complex decision making instead of just selecting between options.

Nevertheless, these experiments show that one can analyze the spontaneous redistribution of traffic load based on individual decision making available while moving through the network. Yet, for what form of simulation objectives such intermediate models between the highly abstract game-theoretic approaches and full detail simulations containing daily plans with mode- and route choice informing detailed mobility simulations, is not yet clear. If it is so difficult to map the gametheoretic scenarios to real world situation, their explanatory value needs to be discussed.

References

1 A. L. Bazzan and F. Klügl. A Review on Agent-Based Technology for Traffic and Transportation, In. The Knowledge Engineering Review, published online May 2013.
2 T. Chmura, T. Pitz and M. Schreckenberg. 2003. Minority Game – Experiments and Simulations. In. S. Hoogendoorn, et al. Traffic and Granular Flow, 2003, pp. 305-316
3 F. Klügl and G. Rindsfüser. 2011. Agent-based Route (and Mode) Choice Simulation in Real-World Networks, Proceedings of the WI-IAT 2011, Lyon, August 2011.
4 F. Klügl and A.L.C Bazzan, 2004. Route Decision Behaviour in a Commuting Scenario: Simple Heuristics Adaptation and Effect of Traffic Forecast in: JASSS Vol. 7(1), 2004.

3.5 Stream Processing and Crowdsourcing for Urban Traffic Management

Thomas Liebig (TU Dortmund, DE)

Joint work of Artikis, Alexander; Bockermann, Christian; Boutsis, Ioannis; Gal, Avigdor; Gunopulos, Dimitrios; Kalogeraki, Vana; Kinane, Dermot; Liebig, Thomas; Mannor, Shie; Marecek, Jakub; Morik, Katharina; Piatkowski, Nico; Schnitzler, Francois; Weidlich, Matthias
URL http://www.insight-ict.eu/

The recent development of innovative technologies related to mobile computing combined with smart city infrastructures is generating massive, heterogeneous data and creating the opportunities for novel applications. In traffic monitoring, the data sources include traditional ones (sensors) as well as novel ones such as micro- blogging applications like Twitter; these provide a new stream of textual information that can be utilized to capture events, or allow citizens to constantly interact using mobile sensors.

The long term goal of the related INSIGHT project is to enable traffic managers to detect with a high degree of certainty unusual events throughout the network. We report on the design of a monitoring system that takes input from a set of traffic sensors, both static (intersection located, traffic flow and density monitoring sensors) and mobile (GPS equipped public transportation buses). We explore the advantages of having such an infrastructure available and address its limitations.

We give an overview of the system developed to address the veracity, velocity and sparsity problems of urban traffic management. The system has been developed as part of the European FP7 project INSIGHT under grant 318225. The general architecture is given in [1]. We describe the input and output of the system, the individual components that perform the data analysis, and the stream processing connecting middleware.

We base the stream processing based on the Streams framework [2]. Streams provides a XML-based language for the description of data flow graphs that work on sequences of data items which are represented by sets of key-value pairs, i.e. event attributes and their values. The actual processing logic, i.e. the nodes of the data flow graph, is realised by processes that comprise a sequence of processors. Processes take a stream or a queue as input and processors, in turn, apply a function to the data items in a stream. All these concepts are implemented in Java, so that adding customized processors is realised by implementing the respective interfaces of the Streams API. In addition, Streams allows for the specification of services, i.e. sets of functions that are accessible throughout the stream processing application.

We extend and apply the system for individual trip planning that incorporates future traffic hazards in routing [3]. Future traffic conditions are computed by a Spatio-Temporal Random Field [6] based on a stream of sensor readings. In addition, our approach estimates traffic flow in areas with low sensor coverage using a Gaussian Process Regression [4,5]. The conditioning of spatial regression on intermediate predictions of a discrete probabilistic graphical model allows to incorporate historical data, streamed online data and a rich dependency structure at the same time. We demonstrate the system and test model assumptions with a real-world use-case from Dublin city, Ireland.

References

1. A. Artikis, M. Weidlich, F. Schnitzler, I. Boutsis, T. Liebig, N. Piatkowski, C. Bockermann, K. Morik, V. Kalogeraki, J. Marecek, A. Gal, S. Mannor, D. Gunopulos, and D. Kinane. Heterogeneous stream processing and crowdsourcing for urban traffic management. In *Proc. of the 17th International Conference on Extending Database Technology*, *to appear*, 2014.
2. C. Bockermann and H. Blom. The streams framework. Technical Report 5, TU Dortmund University, 12 2012.
3. T. Liebig, N. Piatkowski, C. Bockermann, and K. Morik. Predictive trip planning – smart routing in smart cities. In *Proceedings of the Workshop on Mining Urban Data at the International Conference on Extending Database Technology*, page (to appear), 2014.
4. T. Liebig, Z. Xu, and M. May. Incorporating mobility patterns in pedestrian quantity estimation and sensor placement. In J. Nin and D. Villatoro, editors, *Citizen in Sensor Networks*, volume 7685 of *Lecture Notes in Computer Science*, pages 67–80. Springer Berlin Heidelberg, 2013.
5. T. Liebig, Z. Xu, M. May, and S. Wrobel. Pedestrian quantity estimation with trajectory patterns. In P. A. Flach, T. Bie, and N. Cristianini, (eds.), *Machine Learning and Knowledge Discovery in Databases*, vol. 7524 of *Lecture Notes in Computer Science*, pp. 629–643. Springer, 2012.
6. N. Piatkowski, S. Lee, and K. Morik. Spatio-temporal random fields: compressible representation and distributed estimation. *Machine Learning*, 93(1):115–139, 2013.

3.6 Social networks and activity/travel behaviour

Nicole Ronald (The University of Melbourne, AU)

An overview of research into social networks and activity/travel behaviour was presented. The main areas discussed were the influence of social networks on travel behaviour, how social network data has been collected [4], how data can be used to synthesise relationships in a population [1], and how models of social activities and influence can be developed [2,3,5]. Parallels to the field of computational transportation science were also discussed.

References

1. T. Arentze, P. van den Berg, and H. Timmermans. Modeling social networks in geographic space: approach and empirical application. Environment and Planning A, 44:1101-1120, 2012.
2. J. Hackney. Integration of social networks in a large- scale travel behaviour microsimulation. PhD thesis, ETH Zurich, 2009.
3. J. Illenberger. Social Networks and Cooperative Travel Behaviour. PhD thesis, Technische University at Berlin, 2012.
4. M. Kowald, P. van den Berg, A. Frei, J. Carrasco, T. Arentze, K. Axhausen, D. Mok, H. Timmermans, and B. Wellman. The spatiality of personal networks in four countries: A comparative study. In Proceedings of the 13th International Conference on Travel Behaviour Research, 2012.
5. N. Ronald. Modelling the effects of social networks on activity and travel behaviour. PhD thesis, Eindhoven University of Technology, 2012.

3.7 Intelligent Parking Assistant

Bo Xu (Argonne National Laboratory, US)

Joint work of Xu, Bo; Wolfson, Ouri

Finding parking is a major hassle for drivers in urban areas. Studies conducted in 11 major cities reveal that the average time searching for curbside parking was 8.1 minutes and cruising for these parking slots accounted for 30% of the traffic congestion in those cities. Each parking slot generates at least 1,825 vehicle miles traveled (VMT) per year. Thus, in a city like Chicago with over 35,000 curbside slots, cruising for parking generates 63 million VMT. This wastes over 3.1 million gallons of gasoline and produces over 48,000 tons of CO2 emissions.

Intelligent Parking Assistant (IPA), currently being developed in UIC, is a software application that runs on smartphones and car navigation systems, and guides a driver to a parking slot similarly to a Car Navigation System that guides her to the destination.

IPA consists of two subsystems, a Parking Detector (PD) and a Parking Navigator (PN). PD automatically estimates the average number of parking slots on city blocks. It uses a novel method that builds a historical profile of parking availability on each city block, and combines it with real-time information from smartphones. In contrast, existing solutions that detect parking availability either use specialized expensive sensors, or require manual input. PN guides the driver through a path where she is most likely to compete effectively for parking. PN uses a novel Gravity-based Parking Algorithm, developed using game-theory to address the competitive aspect

4 Introductory CTS Course

Seminar participants, in groups, developed modules for an introductory level course in CTS as it could be offered in the third or fourth year of a computer science or engineering program. Each module was to contain material of core knowledge for one to two class lectures, with reading lists added as appropriate. The following nine themes were identified and are further described below:

1. Knowledge discovery
2. Route planning
3. Optimization
4. Visualization and visual analytics
5. Simulation and prediction analysis
6. Stream processing
7. Data modelling
8. Incentive design
9. Human factors

4.1 Knowledge Discovery

Bo Xu, Sergio Di Martino, and Franziska Klügl

The Knowledge Discovery topic would be designed as follows:

Subsections

- Introduction: Available Data Sources, Typical Questions, General Knowledge Discovery Process
- Methods: Problem classes; Method categories and Clustering in more detail
- Tools: WEKA

Learning Objectives

- Overview of knowledge discovery process with special emphasis on spatial data
- Students can evaluate which method category is suitable for which problem
- Students are enabled to perform simple knowledge discovery tasks

A potential homework assignment would be to use WEKA with self-programmed similarity metrics (requires JAVA) for k-means clustering on data such as that from SFpark[2].

[2] http://sfpark.org/

4.2 Route Planning

Harvey Miller, Stefan Funke, and Peter Sanders

The recommended Route Planning course would be structured as follows:

Recommended Prerequisites
- graphs, graph representations
- Dijkstra's algorithm

Class subsections
- Modeling: route planning on road networks
- Basic algorithms
- Speed-Up techniques
- Modeling: route planning in PT networks
- NP-hard route planning variants (this will link to optimization topics)

Learning objectives
- ability to model basic route planning tasks
- being able to solve them using basic algorithms
- awareness of existing speedup-techniques
- intuition for complexity of different route planning tasks

4.3 Optimization in the Context of CTS

Jan Fabian Ehmke

Joint work of Ehmke, Jan Fabian; Fitzpatrick, Rob; Storandt, Sabine

Prerequisites
Ideally, students should be familiar with complexities and (linear) optimization, as of course those topics can not be covered in a single course. Nevertheless, we try to motivate and explain optimization in the CTS context relying on prior knowledge as little as possible.

Learning Objectives
The overall goal is to show how optimization can be useful in the context of CTS. For that purpose, a basic understanding of mathematical modelling and problem solving should be conveyed. Moreover, the difference between exact optimization and the usage of heuristics should become clear. To motivate these approaches, different levels of complexity need to be explained, most convenient based on some well-known optimization problem (e.g. Travelling Salesman). Finally, the incorporation of real-world features and constraints should be sketched.

Structure of the Lecture
We present a template structure with the steps we consider important to introduce students to CTS related optimization. We accompany each step with a concrete suggestion, but we want to emphasize that depending on the expertise and interest of the lecturer the discussed problems and examples can easily be exchanged.

1. **Motivating example problem.** This problem should be transportation related but simple and easy to explain at the same time, like e.g. vehicle scheduling, facility location or the vehicle routing problem (VRP). We focus on the latter in the following. Based on the motivating example it should be explained what an objective is in general, what the concrete objective for the example is and what other common objectives are relevant in CTS.
2. **Methods for Solving Optimization Problems.** Now we want to give some insights in solution methods. We concentrate on the motivation example for this purpose. We even suggest to use the most simplified version of this example for consideration, for VRP we suggest to use the Travelling Salesman Problem (TSP). We concentrate on three ways to solve optimization problems:
   - Naive, i.e., for TSP compute all possible permutations. We would illustrate the impracticability of this approach with a runtime estimation for a reasonably sized TSP instance on some machine. Also the theoretical runtime formula explaining this combinatorial explosion should be given along. This should arise the question how to handle such computational complex problems in a better way.
   - Mathematical Modelling is a logical next step. Here the objective should be formulated as a function and students should get a glimpse on how constraints can be realized. To show the usefulness of mathematical models, the existence of solvers should be mentioned, maybe showing a small example of the TSP formulation for a specific solver and the received result (along with the runtime).
   - Heuristics should then be introduced as a remedy for retrieving solutions also for larger instances which cannot be solved exactly in a reasonable time span. Some simple heuristics for the chosen example should be explained, e.g. for TSP the nearest neighbor greedy algorithm, the insertion approach and the savings technique are recommendable. It should be emphasized that we compromise solution quality for the sake of run time with these approaches.
3. **Modelling Problems More Realistically.** Now the way from the abstract optimization problem back to real-world problem should be found. This can be done by enumerating constraints one has to deal with in practice. For our VRP example, this could include time windows, time-dependent travel times along with traffic congestions, varying capacities of the vehicles and possible drivers' or driving restrictions. Normally, there are ways to extend the mathematical model in order to incorporate such constraints, but, of course, it gets more complex in that way. Also a remark on the integrability of real-world data should be made here (reference to the lecture on 'Data Modelling'). Then the extension of a heuristic for a real-world constraint can show the power of such techniques, e.g. the extension of the insertion heuristic for TSP taking time windows into account. Again, a look on the complexity (here factorial versus quadratical) and solution quality makes sense.
4. **Outlook.** As obviously in a single course the huge world of optimization can not be covered, we finally give an overview of exemplary extensions that consider time-space dimensions important in the CTS context, like
   - static versus dynamic problems
   - several types of heuristics (approximation algorithms, metaheuristics, ...)
   - other solution techniques (dynamic programming, local search, ...)
   - linear versus non-linear optimization

Useful Resources
- Online OpenStreetMap TSP[3]
- Online GoogleMaps TSP[4]
- Excel VRP Solver (Open Source)[5]
- Vehicle Routing Game[6]
- Online course on Discrete Optimization (van Hentenryck)[7]

References

1 George L. Nemhauser (1988). *Integer and combinatorial optimization*, Wiley-Interscience series in discrete mathematics and optimization. Wiley, New York.
2 Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J., 2011. *The Traveling Salesman Problem: a Computational Study.* Princeton University Press, Princeton.
3 Jan Fabian Ehmke 2012. *Integration of information and optimization models for routing in city logistics*, International series in operations research & management science. Springer, New York.

4.4 Visualisation and Visual Analytics

Walied Othman

In this course we will convince the students of the need for visualization and visual analytics. There are multiple examples, even dating back to two centuries ago, that the eye can absorb a lot of information when presented visually, i.e., a train schedule from 1885 between Paris and Lyon. Not only is visualization important in conveying information, it is a key player in knowledge discovery via visual analytics. In this course we will show that visual analytics is a back-and-forth process between visually inspecting data, visually selecting an area of interest and applying a transformation if necessary, from which knowledge can be generated and visualized. This is a creative process, and the best way to approach this is to provide the students with a plethora of examples from which they can draw inspiration from. The class will go through a hands-on process of visual analytics on an example data set and receive a similar take-home assignment. In the following class, students get a chance to present their work to each other and validate their visualizations with, for example, a survey.

---

[3] http://algohol01.informatik.uni-stuttgart.de/
[4] http://www.gebweb.net/optimap/
[5] http://verolog.deis.unibo.it/news-events/general-news/vrp-spreadsheet-solver
[6] http://game.costtoserve.net/
[7] https://www.coursera.org/course/optimization

4.5 Simulations in transportation science

Francesco Ciari, Nicole Ronald, and Seng Wai Loke

Computer power and availability of data have grown dramatically in recent years making even data and computational intensive simulations a viable planning option. This lecture offers students an overview of simulation techniques as applied to transportation science. The main goal of this lecture is to provide students with some background on the topic and an understanding of the situations in which simulations can be appropriate transport planning tools. The students will be given the opportunity to "play around" with some simulation tools in order to get a direct experience of pros and cons of this type of technique. Below, a structure for a 90-minute lecture is provided.

Motivation
- The application of computational tools to assist in making decisions about planning design and operations of transportation systems
- Many problems cannot be explored directly but need to be simulated

Learning Objectives
- Be able to select the appropriate simulation for a given problem
- Understand and apply simulation tools to solve transportation problems
- Being able to interpret and analyze the outputs

Cellula Automata (CA) + Traffic Microsimulation
- CA Model
    - Discrete time and space (grid-based)
- Traffic Microsimulation
    - Discrete time, continuous space
    - Examples (Videos)
    - Topics include car following algorithms and applications

Agent-based simulations
- Based on MATSim
- Discrete time, queue-based
- Topics include overall process, utility functions, and applications
- Example (Videos)
- Can also make use of existing teaching materials (see Readings)

Scope and Limitations
- For each approach:
    - What is the scope
    - What are their inherent limitations
    - Which kind of output
    - How they do compare

Assignment
- Students get a small scenario, some options to investigate policies and are assigned the appropriate model (i.e. MATSim or Sumo) to investigate it
- Students should implement the given policy scenario, run the simulation and draw conclusions based on the analysis of the results

References

1 Balmer, M. (2007) Travel Demand Modeling for Multi-Agent Transport Simulations: Algorithms and Systems, Dissertation, ETH Zürich, Zürich.
2 Barceló, J. (2010) Models, Traffic Models, Simulation, and Traffic Simulation, Springer, New York.
3 Helbing, D. (2012) Social Self-Organization, Understanding Complex Systems, Springer, Berlin.
4 Heppenstall, A., A. T. Crooks, L. M. See and M. Batty (2012) Agent-Based Models of Geographical Systems, Springer, Dordrecht.
5 Matsim (2012) Matsim Agent-based Transport simulations – tutorials, webpage, March 2014, http://www.matsim.org/docs/tutorials
6 Sumo (2014) Simulation of Urban Mobility, webpage, March 2014, http://sumo-sim.org/
7 Horni, A. (2013) Destination choice modeling of discretionary activities in transport microsimulation, Dissertation, IVT, ETH Zurich, Zurich.
8 Meister, K. (2011) Contribution to Agent-Based Demand Optimization in a Multi-Agent Transport Simulation, Dissertation, Department Bau, Umwelt und Geomatik, ETH, Zürich.
9 Rieser, M. (2010) Adding Transit to an Agent-Based Transportation Simulation: Concepts and Implementation, Dissertation, TU Berlin, Berlin.

4.6 Stream Processing in the Context of CTS

Sergio Ilarri, Thomas Liebig, Steve Liang, and Glenn Geers

The recent development of innovative technologies related to mobile computing combined with smart city infrastructures is generating massive, heterogeneous data and creating opportunities for novel applications in transportational computation science. The heterogeneous data sources provide streams of information that can be used to create smart cities. The knowledge on stream analysis is thus crucial and requires collaboration of people working in logistics, city planning, transportation engineering and data science.

We provide a list of materials for a course on stream processing for computational transportation science. The objectives of the course are:

- Motivate data stream and event processing, its model and challenges.
- Acquire basic knowledge about data stream processing systems.
- Understand and analyze their application in the transportation domain.

Since the subject is large and comprises many aspects, we propose that the course should start with an examplary application which is familiar to the audience. The chosen example expands through the whole course and illustrates a particular aspect in each section.

Topics to be covered

1. Introduction
   - Literature
     - Models and Issues in Data Stream Systems [4, 13]
     - Data Stream Management: [21]
     - Transportation and Data Streams: [40]
     - Smart Cities and Heterogeneous Data Streams: [36]
   - Event stream examples in transportation

  - linear ordered sequence events, e.g., bus arrival times
  - an event cloud consists of many event streams, e.g., traveler arrival time and bus arrival time at interchange
  - moving car trajectories
- Challenges in stream processing
- OGC standards and interfaces [15]

2. Data Stream Management Systems (DSMSs)
   - Lambda Architecture for Stream Processing [31]
   - Speed Layer: STREAM [1], Aurora [6], Borealis [39], Storm [38], streams [9], S4 [32], Kafka [19]
   - Batch Layer: MapReduce [14], Hadoop [41], Spark [42], Disco [16]
   - Distributed NoSQL Databases: Cassandra [24], MongoDB [35]
3. Data Analysis
   - Query Languages: Esper [30], NiagaraCQ [10], and others [5, 22, 25]
   - Complex Event Processing (CEP): [7, 12, 27]
   - Learning: streams [9], Mahout [33], MOA [8]
   - Distributed streams: [37]
   - Sketches: [11, 18] privacy with sketches [23]
4. Example applications in the transportation domain: [2, 3, 17, 20, 26, 28, 29, 34, 43]

Possible home assignment
- Study a certain DSMS and summarize its features in a report.

References

1 Arvind Arasu, Brian Babcock, Shivnath Babu, Mayur Datar, Keith Ito, Rajeev Motwani, Itaru Nishizawa, Utkarsh Srivastava, Dilys Thomas, Rohit Varma, and Jennifer Widom. STREAM: The Stanford stream data manager. *IEEE Data Eng. Bull.*, 26(1):19–26, 2003.

2 Arvind Arasu, Mitch Cherniack, Eduardo Galvez, David Maier, Anurag S. Maskey, Esther Ryvkina, Michael Stonebraker, and Richard Tibbetts. Linear Road: A stream data management benchmark. In *Proceedings of the 30th International Conference on Very Large Data Bases – Volume 30*, VLDB '04, pages 480–491. VLDB Endowment, 2004.

3 Alexander Artikis, Matthias Weidlich, Francois Schnitzler, Ioannis Boutsis, Thomas Liebig, Nico Piatkowski, Christian Bockermann, Katharina Morik, Vana Kalogeraki, Jakub Marecek, Avigdor Gal, Shie Mannor, Dimitrios Gunopulos, and Dermot Kinane. Heterogeneous stream processing and crowdsourcing for urban traffic management. In *Proceedings of the 17th International Conference on Extending Database Technology*, page (to appear), 2014.

4 Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, pages 1–16, New York, NY, USA, 2002. ACM.

5 Shivnath Babu and Jennifer Widom. Continuous queries over data streams. *SIGMOD Rec.*, 30(3):109–120, September 2001.

6 Hari Balakrishnan, Magdalena Balazinska, Don Carney, Uğur Çetintemel, Mitch Cherniack, Christian Convey, Eddie Galvez, Jon Salz, Michael Stonebraker, Nesime Tatbul, Richard Tibbetts, and Stan Zdonik. Retrospective on Aurora. *The VLDB Journal*, 13(4):370–383, December 2004.

7 Tim Bass. Mythbusters: Event stream processing versus complex event processing. In *Proceedings of the 2007 Inaugural International Conference on Distributed Event-based Systems*, DEBS '07, page 1, New York, NY, USA, 2007. ACM.

8 Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: Massive online analysis. *J. Mach. Learn. Res.*, 11:1601–1604, August 2010.

9 Christian Bockermann and Hendrik Blom. The streams framework. Technical report, TU Dortmund University, 2012.

10 Jianjun Chen, David J. DeWitt, Feng Tian, and Yuan Wang. NiagaraCQ: A scalable continuous query system for internet databases. *SIGMOD Rec.*, 29(2):379–390, May 2000.

11 Graham Cormode, Minos Garofalakis, Peter J. Haas, and Chris Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Found. Trends databases*, 4:1–294, January 2012.

12 Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.*, 44(3):15:1–15:62, June 2012.

13 Lukasz Golab and M. Tamer Özsu. Issues in data stream management. *SIGMOD Rec.*, 32, 2:5–14, June 2003.

14 Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

15 Johannes Echterhoff and Thomas Everding. OpenGIS Sensor Event Service Interface Specification. Technical Report OGC 08-133, http://portal.opengeospatial.org/files/?artifact_id=29576, October 2008.

16 Jared Flatow, Prashanth Mundkur, and Ville Tuulos. Disco: A computing platform for large-scale data analytics. In *Proceedings of the 10th ACM SIGPLAN workshop on Erlang – Erlang '11*, Tokyo, Japan, 2011. ACM Press, ACM Press.

17 Simona Florescu, Christine Körner, Michael Mock, and Michael May. Efficient mobility pattern stream matching on mobile devices. In *Proc. of the Ubiquitous Data Mining Workshop (UDM 2012)*, pages 23–27, 2012.

18 Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. Mining data streams: A review. *SIGMOD Rec.*, 34(2):18–26, June 2005.

19 N. Garg. *Apache Kafka*. Packt Publishing, 2013.

20 Sandra Geisler, Christoph Quix, Stefan Schiffer, and Matthias Jarke. An evaluation framework for traffic information systems based on data streams. *Transportation Research Part C: Emerging Technologies*, 23(0):29–55, 2012.

21 L. Golab and M. Tamer Özsu. *Data Stream Management*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.

22 Namit Jain, Shailendra Mishra, Anand Srinivasan, Johannes Gehrke, Jennifer Widom, Hari Balakrishnan, Uğur Çetintemel, Mitch Cherniack, Richard Tibbetts, and Stan Zdonik. Towards a streaming SQL standard. *Proc. VLDB Endow.*, 1(2):1379–1390, August 2008.

23 Michael Kamp, Christine Kopp, Michael Mock, Mario Boley, and Michael May. Privacy-preserving mobility monitoring using sketches of stationary sensor readings. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 8190 of *Lecture Notes in Computer Science*, pages 370–386. Springer Berlin Heidelberg, 2013.

24 Avinash Lakshman and Prashant Malik. Cassandra: Structured storage system on a P2P network. In *Proceedings of the 28th ACM Symposium on Principles of Distributed Computing*, PODC '09, pages 5–5, New York, NY, USA, 2009. ACM.

25 Yan-Nei Law, Haixun Wang, and Carlo Zaniolo. Query languages and data models for database sequences and data streams. In *Proc. of the 30th International Conference on Very Large Data Bases – Volume 30*, VLDB '04, pages 492–503. VLDB Endowment, 2004.

26 Thomas Liebig, Nico Piatkowski, Christian Bockermann, and Katharina Morik. Predictive trip planning – smart routing in smart cities. In *Proceedings of the Workshop on Mining Urban Data at the International Conference on Extending Database Technology*, (to appear), 2014.

27 Gérard Ligozat, Zygmunt Vetulani, and Jedrzej Osinski. Spatiotemporal aspects of the monitoring of complex events for public security purposes. *Spatial Cognition & Computation*, 11(1):103–128, 2011.

28 Wei Liu, Yu Zheng, Sanjay Chawla, Jing Yuan, and Xie Xing. Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1010–1018, New York, NY, USA, 2011. ACM.

29 Ying Liu, Alok N. Choudhary, Jianhong Zhou, and Ashfaq A. Khokhar. A scalable distributed stream mining system for highway traffic data. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *PKDD*, volume 4213 of *Lecture Notes in Computer Science*, pages 309–321. Springer, 2006.

30 Floyd Marinescu. Esper: High volume event stream processing and correlation in Java. Online article, July 2006.

31 Nathan Marz. *Big Data: Principles and best practices of scalable realtime data systems.* O'Reilly Media, 2013.

32 Leonardo Neumeyer, Bruce Robbins, Anish Nair, and Anand Kesari. S4: Distributed stream computing platform. In *Proc. of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 170–177, Washington, DC, USA, 2010. IEEE CS.

33 Sean Owen, Robin Anil, Ted Dunning, and Ellen Friedman. *Mahout in Action.* Manning Publications Co., Manning Publications Co. 20 Baldwin Road PO Box 261 Shelter Island, NY 11964, first edition, 2011.

34 Oliver Pawlowski, JÌrgen Dunkel, Ralf Bruns, and Sascha Ossowski. Applying event stream processing on traffic problem detection. In Luis Seabra Lopes, Nuno Lau, Pedro Mariano, and Luis M. Rocha, editors, *Progress in Artificial Intelligence*, volume 5816 of *Lecture Notes in Computer Science*, pages 27–38. Springer Berlin Heidelberg, 2009.

35 Eelco Plugge, Tim Hawkins, and Peter Membrey. *The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing.* Apress, Berkely, CA, USA, 1st edition, 2010.

36 IBM Research. System requirements spec, standards and guidelines for development and architecture. Technical Report FP7-318225 D2.1, INSIGHT Consortium, August 2013.

37 Izchak Sharfman, Assaf Schuster, and Daniel Keren. A geometric approach to monitoring threshold functions over distributed data streams. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 301–312, New York, NY, USA, 2006. ACM.

38 *Storm – Distributed and fault-tolerant realtime computation*, Available: http://storm-project.net/ [Last accessed: 27 June 2013], 2013.

39 Nesime Tatbul, Yanif Ahmad, Ugur Cetintemel, Jeong-Hyon Hwang, Ying Xing, and Stan Zdonik. Load management and high availability in the Borealis distributed stream processing engine. In Silvia Nittel, Alexandros Labrinidis, and Anthony Stefanidis, editors, *GeoSensor Networks*, volume 4540 of *Lecture Notes in Computer Science*, pages 66–85. Springer Berlin Heidelberg, 2008.

40 Piyushimita Vonu Thakuriah and D. Glenn Geers. *Transportation and Information: Trends in Technology and Policy.* Springer Publishing Company, Incorporated, 2013.

41 Tom White. *Hadoop: The Definitive Guide.* O'Reilly Media, Inc., 1st edition, 2009.

42 Matei Zaharia, N. M. Mosharaf Chowdhury, Michael Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. Technical Report UCB/EECS-2010-53, EECS Department, University of California, Berkeley, May 2010.

43 Jiadong Zhang, Jin Xu, and Stephen Shaoyi Liao. Aggregating and sampling methods for processing GPS data streams for traffic state estimation. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1629–1641, 2013.

4.7 Data Modeling

Dirk Christian Mattfeld, Patrick Laube, and Monika Sester

The recommended Data Modeling course would be structured as follows:

Class subsections
- The term ‘model’, definition and properties (mapping, abstraction and reduction, goal orientation)
- Three levels for CTS tasks: sensing, control, and planning
- Conceptual data models for movement traces and movement spaces
  - conceptual data models for spatial information: entities and fields
  - graphs, networks
  - trajectories

Learning objectives
- The students can give a concise definition of the concept ‘model’, and can list its key properties.
- The students can assign given CTS tasks to the three task levels (1) sensing, (2) control, and (3) planning, and can name methods suitable for the given tasks.
- The students acquire a workbench of basic conceptual data models required for CTS (including conceptual data models for spatial information, ...).
- The students understand that depending on the task at hand, different data models are required.

4.8 Incentive Design

Alexandros Labrinidis and Ouri Wolfson

The recommended Incentive Design course would be structured as follows:

Class subsections
- introductory microeconomics
- game theory/mechanism design
- examples of incentive schemes in transportation

Learning objectives
- Understand Nash equilibrium/Pareto efficiency – personal versus global optima
- Understand law of supply and demand / differential pricing
- Understand mechanism design, taking into account “attacks”
- Review uses of incentives, with regards to transportation/mobility
- Understand types of incentives

Sample assignments
- Simulate introduction of HOV lane
- Walk-or-wait at stalled elevator

References

1 Grady Klein, Yoram Bauman *The Cartoon Introduction to Economics, Volume 1: Microeconomics*, Hill and Wang, New York, NY (2010).
2 Harvey J. Miller. Beyond sharing: cultivating cooperative transportation systems through geographic information science, *Journal of Transport Geography*, 31 (2013).

4.9 Human Factors

Stephan Winter, Sabine Timpf, and Caitlin Cottrill

The recommended Human Factors course would be structured as follows:

Class subsections
- information availability and usability
- relevance theory
- activity theory
- spatial cognition
- socio-technical systems theory
- user satusfaction/usability testing
- incentives (see above)

Learning objectives
- Students will understand that people are at the centre of transport. People's transport choices determine traffic, and information *about* their choices influences their choices, along with the trip purpose and existing constraints (context).
- Students will experience and evaluate the direct and indirect costs caused by the particular design of transport information (systems). The direct costs are those of finding information, and the indirect costs are those that follow from not using systems that are inconvenient to use and using the private car instead.

Potential assignments
- Urban trail: The student will design a route description for a route passing five landmark features in the city, for various types of people. Compare your descriptions by chosen references and the spatial granularity of each of them.
- Unfamiliar mode: The student should come to the university with a mode different from that usually taken, and report about any usability issues (such as consistency of information, availability of information needed, etc.)
- Usability testing: The student will test two common tasks for mobile public transport planners (1. When does my next tram or bus depart? 2. I need to be at uni at 10am, so at what time do I need to be at my local tram/bus stop?)
- Can the student take the car navigation out of the car and use it for walking? Report issues.
- The student should plan his or her next holiday trip online and (a) observe the time needed, (b) report usability issues
- The student will use competing platforms for trip planning and compare them from a usability perspective.

Core concepts and theoretical bases

- Relevance theory [14]
  - Discuss along the patterns of a natural language dialog about start and destination point.
- Activity theory [5]
  - Discuss purpose of trip and implications on what information is relevant
  - Decision-support with mobile devices [13]
- Spatial Cognition
  - Mental spatial representations [1], schematic geometry [10]
  - Place, route and survey knowledge [11]
  - Structure of place and route descriptions [12]; [15] – include hierarchies of spatial (and temporal) granularities. Related: the challenge of transfers – not a node, but most complex part of a journey [4]
- Socio-technical systems theory [3]
  - Reliance on technology (e.g., navigation systems) relieves from forming mental spatial representations
  - Impacts of communication technology & associated temporal shifts on expectations of travel and travel planning [8]
  - Technological artifacts and their behaviour should conform to mental representations and models [9]

References

1 Kevin Lynch *The Image of the City.* The MIT Press, Cambridge, MA, 194 pp.
2 Wilson, D.; Sperber, D. (2004): Relevance Theory. In: Horn, L. R.; Ward, G. (Eds.), *Handbook of Pragmatics*, Blackwell, Oxford, pp. 607–632.
3 Geels, F. W. (2004): From sectoral systems of innovation to socio-technical systems: Insights about dynamics and change from sociology and institutional theory. *Research Policy*, 33 (897–920).
4 Heye, C., and Timpf, S. “Factors influencing the physical complexity of routes in public transportation systems.” *International Conference on Travel Behavior Research*, Luzern, Switzerland, 2003.
5 Kaptelinin, V.; Nardi, B.; Macaulay, C. (1999): Methods & tools: The activity checklist: a tool for representing the “space” of context. *interactions*, 6 (4): 27–39.
6 Kjeldskov, J.; Graham, C. (2003): A Review of Mobile HCI Research Methods. In: Chittaro, L. (Ed.), Mobile HCI 2003. *Lecture Notes in Computer Science*, 2795. Springer, Berlin, pp. 317–335.
7 Kjeldskov, J.; Graham, C.; Pedell, S.; Vetere, F.; Howard, S.; Balbo, S.; Davies, J. (2005): Evaluating the usability of a mobile guide: The influence of location, participants and resources. *Behaviour & Information Technology*, 24 (1): 51–65.
8 Kwan, M. P. (2007): Mobile Communications, Social Networks, and Urban Travel: Hypertext as a New Metaphor for Conceptualizing Spatial Interaction. *The Professional Geographer*. Volume 59, Issue 4, pp. 434–446.
9 Norman, Donald A. (1991): Cognitive artifacts. In: Carroll, John M. (ed.). *Designing Interaction: Psychology at the Human-Computer Interface*. Cambridge, UK: Cambridge University Press, pp. 17–38
10 Rüetschi, U. J. and Timpf, S. “Modelling Wayfinding in Public Transport: Network Space and Scene Space”. In: C. Freksa and et.al. (Editors), *Spatial Cognition 2004*. Springer, Lake Chiemsee, Bavaria, Germany, pp. 24–41, 2004.

11 Siegel, A. W.; White, S. H. (1975): The Development of Spatial Representations of Large-Scale Environments. In: Reese, H. W. (Ed.), *Advances in Child Development and Behavior*, 10. Academic Press, New York, pp. 9–55.

12 Tenbrink, T.; Winter, S. (2010): Presenting Spatial Information: Granularity, Relevance, and Integration (Theme Section Editorial). *Journal of Spatial Information Science*, 1 (1): 49–51.

13 Timpf, S. "Wayfinding with mobile devices – decision support for the mobile citizen". In: S. Rana and J. Sharma (Eds.), *Frontiers in GI-Technology*. Springer, 2006, pp. 209–228.

14 Wilson, D.; Sperber, D. (2004): Relevance Theory. In: Horn, L. R.; Ward, G. (Eds.), *Handbook of Pragmatics*, Blackwell, Oxford, pp. 607–632.

15 Wu, Y.; Winter, S. (2010): Interpreting Place Descriptions for Navigation Services. In: Bateman, J.; Cohn, A. G.; Pustejovsky, J. (Eds.), *Spatial Representation and Reasoning in Language: Ontologies and Logics of Space*. Dagstuhl Seminar Proceedings. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany, Schloss Dagstuhl, Germany.

5 Application Challenge

For the application challenge, participants competed in five groups to develop an original CTS application and a feasibility assessment. Presentations took place on the final afternoon of the seminar. The following abstracts were received.

5.1 karmobility

Alexandros Labrinidis, Patrick Laube, Steve Liang, Harvey J. Miller, Nicole Ronald, Stephan Winter, and Yang (Tara) Yue

karmobility[8] is an app designed to "increase your mobility karma" in the following ways:

Integration
- in the physical world, allow for the integration of multi-modal transportation options
- in the digital world, enable the integration of information from different sources and with other apps

Incentives
- in the physical world, encourage/reward behavior change
- in the digital world, encourage/reward participatory sensing

Karma
- in the physical world, people's good mobility choices have a positive impact on the environment and on public health
- in the digital world, people's good mobility choices are shared through social networks

[8] For more information, see https://sites.google.com/site/karmobility/home

5.2 HappyParking

Sergio di Martino, Rob Fitzpatrick, Sergio Ilarri, and Ouri Wolfson

There are estimations that indicate that about half of the vehicles on the move are searching for parking and that more than 40% of the total fuel consumption is spent while looking for an available parking space. This also contributes to significant urban traffic congestion. So, it would be interesting to have software tools that can help drivers to park easily.

For the application challenge, our group proposed a *HappyParking* application, which would offer some interesting benefits:

- It ackowledges the importance of considering parking in the context of a displacement between a source location and a target location. This implies that the final target location has to be considered when deciding an appropriate parking space. Moreover, the application can be integrated into existing GPS-based navigation applications.
- It considers multimodality, that is, that parking a car could be just a leg within a longer trip using different modes of transportation.
- It exploits real-time constraints (e.g., time-based parking restrictions).
- It can accommodate a variety of methods to capture information about available parking spaces (e.g., magnetic sensors on the parkings, crowdsourcing information provided by drivers releasing a spot, cars with different types of sensors able to detect free places, etc.).
- It supports different types of parking spaces: on-street parking, private parkings and garages, home parking available for rental during specific time periods, etc.

At a very high-level, the architecture supporting HappyParking is composed by four modules: the *Data Capture* module, the *Knowledge Discovery* module, the *Payment* module, and the *Incentives* module. The system is able to offer real-time parking recommendations. It computes the likelihood of parking at the estimated time of arrival: in a certain area, at certain times, and taking into account especial external real-time events (e.g., a big concert or music festival in the city). On the one hand, it offers recommendations that maximize the success probability, the user's satisfaction, and the global benefit for the community of drivers. On the other hand, those recommendations can minimize the distance to the final destination, the time to park, and the economic cost (fuel consumption). For this purpose, it learns the user preferences over time. Finally, for some types of parking spaces it is possible to book and pay for a guaranteed parking spot (e.g., based on a dynamic pricing schema).

In accordance to the above description, this kind of application would be useful for a variety of parties, such as drivers, garages, owners of private parking spaces, municipalities, car manufacturers, OEMs, ecologists and health systems. Nevertheless, the unique combination of features indicated requires additional research work to make such an application a reality.

References

1 F. Abad, R. Bendahan, S. Wybo, S. Bougnoux, C. Vestri and T. Kakinami, "Parking Space Detection", 14th World Congress on Intelligent Transport Systems (ITS), 2007.
2 Daniel Ayala, Ouri Wolfson, Bo Xu, Bhaskar DasGupta and Jie Lin, "Parking in Competitive Settings: A Gravitational Approach", 13th International Conference on Mobile Data Management (MDM 2012), IEEE Computer Society, pp. 27-32, 2012.

3 Murat Caliskan, Daniel Graupner and Martin Mauve, "Decentralized Discovery of Free Parking Places", Third ACM International Workshop on Vehicular Ad Hoc Networks (VANET'06), ACM Press, pp. 30-39, 2006.

4 Thierry Delot, Sergio Ilarri, Sylvain Lecomte and Nicolas Cenerario, "Sharing with Caution: Managing Parking Spaces in Vehicular Networks", Mobile Information Systems, volume 9, number 1, pp. 69-98, IOS Press, 2013.

5 E. Gantelet and A. Lefauconnier, "The Time Looking for a Parking Space: Strategies, Associated Nuisances and Stakes of Parking Management in France", European Transport Conference, IOS press, 2006.

6 Imperial College Urban Energy Systems Project, http://www3.imperial.ac.uk/urbanenergysystems.

7 Leon Stenneth, Ouri Wolfson, Bo Xu and Philip S. Yu, "PhonePark: Street Parking Using Mobile Phones", 13th International Conference on Mobile Data Management (MDM 2012), IEEE Computer Society, pp. 278-279, 2012.

8 sfpark SFpark, http://sfpark.org/.

9 The Telegraph, Motorists spend 106 days looking for parking spots, http://www.telegraph.co.uk/motoring/news/10082461/Motorists-spend-106-days-looking-for-parking-spots.html.

10 Bo Xu, Ouri Wolfson, Jie Yang, Leon Stenneth, Philip S. Yu and Peter C. Nelson, "Real-Time Street Parking Availability Estimation", 14th International Conference on Mobile Data Management (MDM 2013), IEEE Computer Society, pp. 16-25, 2013.

5.3 Ichibi – Everything you ever wanted in a multi-modal travel app

Caitlin Cottrill, Jan Fabian Ehmke, Glenn Geers, Franziska Klügl, and Sabine Timpf

Due to the multitude of travel options and prices, travel planning has become challenging nowadays. For the individual traveller, it is not easy to determine her best travel option. Existing travel apps offer only limited functionality for the consideration of complex traveller's preferences. Furthermore, there is hardly any en-route assistance, especially when it comes to multi-mode travel alternatives. Multi-mode travel options tend to be much more complex, though. En-route assistence would allow for real-time adapation and reaction to travel disruptions.

Ichibi provides pre-trip and en-route functionality and thus comprises everything you ever wanted in a multi-modal travel app. *Ichibi* is

1. Smart – learns the travelers' preferences and constraints,
2. Efficient – sets the travelers' preferences, selects the traveler's preferred alternative, books tickets, and it is
3. Timely – it reacts to travel disruptions and provides alternative travel options if needed.

Ichibi uses a middleware based approach that leverages available transport services and relevant data for travel planning by canonicalizing travel information into a consistent internal format that is able to be processed efficiently. It is able to intelligently manage journey risk and reliability of travel options by pre-computing alternative routes-to-destination from each interchange point. Location awareness enables Ichibi to provide correct and up-to-date travel planning wherever you are on the planet. Available in both free and paid versions, Ichibi is the perfect companion for any traveller.

The *free* version offers no booking options and supports individual travel options only. The *paid* version adds multi-modal booking options, information on user reputation, and the ability to coordinate with other travellers. Both versions support data privacy and location cloaking by default and multi-device synchronisation.

In the following, we discuss pre-trip and en-route support prototypically. We also propose an overview of the corresponding architecture. In this context, we assume that services such as route planning, transport booking, etc. are available; that we have perfect positioning functionality and internet connectivity; that adequate transport network information (timetables, delays, arrival time, etc.) and real-time data (either from transport operators or crowd-sourced from the app) is available; that there is adequate cloud storage on a central server; that the app is designed following the privacy-by-design by design principle.

Pre-trip functionality. Planning of multi-modal travel options is supported as follows. Before Ichibi can be used for travel planning, the traveler has to provide a minimum amount of personal data as well as her travel preferences such as prefered modes, personal risk aversity and the importance of typical travel objectives such as overall travel time, maximum number of transfers, importance of travel costs etc. For a particular trip, information on origin and destination, date, whether it is a return trip or not as well as well as the underlying trip profile (business, leisure, etc.) have to be specified. According to the given information, Ichibi combines information on available services and their expected service quality. As a result, the traveler can choose from a set of different options and book her tickets subsequently. The app informs the traveler if any disruptions occur at any point.

En-Route functionality. Assistance while being en-route is provided as follows. The traveler logs in for a particular trip. If any disruptions become known, the traveler is alerted, especially when delays or disruptions might impact the rest of the journey. In case of severe disruption, Ichibi would provide alternative travel options that the traveler can select. If the traveler switches to an alternative option, detailed information and assistance for the revised trip are provided, and tickets are booked if necessary. After the trip, the user is asked to evaluate the assistance functionality and her individual travel experience. This information can be used to automatically refine the traveler's preferences.

Architecture. Ichibi is built on a cloud solution that integrates the multitude of available travel options and services. The central server provides middleware functionality integrating information on flights, the road network, train services, availability of car sharing services, and related services such as weather forecasts. There are powerful planning, querying and filtering components that are able to determine the best travel options from the multitude of above services. A particular focus of Ichibi is on the assessment of a travel option's reliability and the adherence to buffers that correspond to the risk aversity of the particular traveler and the expected reliability of the (combined) travel options. User preferences and user travel experiences are stored in a secured travel database and revealed to the Ichibi server if approved by the particular traveler.

Added value. The Ichibi travel app is a combination of pre-trip travel planning and on-trip travel assistance. It offers complete booking of multi-mode travel options, information on risks for specific connections, trip coordination with others and timely detection of disruptions.

5.4 Multimodal Pre-trip Planner

Bo Xu, Dirk Mattfeld, and Benjamin Kickhöfer

Motivation. Pre-trip planners are information systems that plan transportation modes, routes, inter-modal transitions for user specified source and destination(s). Typical pre-trip planners for the train/bus mode are based on schedules; typical pre-trip planners for the car mode are based on average travel times and current traffic conditions. These planners do not take into considerations the reliability of travel time of a route. However, there are situations in which the reliability of travel time is important for trip planning. For example, when a traveler needs to catch a flight, the traveler may sacrifice the length of travel time for the predictability of travel time. We propose an application called "Multimodal Pre-trip Planner" to enable pre-trip planning with travel time reliability taken into account.

The Application. The proposed application combines historical data from various sources (web, mobile phones, sensors, ...) in order to provide pre-trip information on an O-D level. The pre-trip information includes the average travel time and reliability (e.g., standard deviation of travel time) for an arbitrary O-D pair. For the train/bus mode it also includes the price for arbitrary time bins of departure. For example, for a trip from Dagstuhl to Frankfurt Airport, the application may provide information such as the following. If departing at 7am, for the car mode, the average travel time is 2 hours, the standard deviation is 20 minutes, and with 85% probability the delay is less than 10 minutes; for the train mode, the average travel time is 3 hours, the standard deviation is 1 minute, and with 98% probability the delay is less than 10 minutes; and so on. Similarly for the departure time of 8am, 9am, etc. In this way the application creates a data cube that can be viewed from different perspectives in terms of departing time, transportation mode, etc.

Architecture and Challenges. The input of the Multimodal Pre-trip Planner is recorded historical travel-time data and real-time travel-time data. For the train mode the input data is on O-D basis. This is because the travel-time collected for the train mode is usually per O-D pair. For the car mode the input data is on link basis. This is because existing traffic monitoring approaches, such as loop sensors or floating car data, usually generate travel time per road-segment. By combining the historical travel-time data and real-time travel-time data, the application computes travel time reliability on O-D basis for the train/bus mode and on link basis for the car mode. The application then aggregates link reliability to get O-D level reliability for the car mode. The data cube is updated accordingly. The data cube can be updated on a daily basis.

One challenge is the integration of data from different sources. Another challenge is aggregation from link-based reliability to O-D based reliability.

Participants

- Francesco Ciari
ETH Zürich, CH
- Caitlin Cottrill
University of Aberdeen, GB
- Sergio Di Martino
Volkswagen AG – Wolfsburg, DE
- Jan Fabian Ehmke
FU Berlin, DE
- Rob Fitzpatrick
NICTA – Eveleigh, AU
- Stefan Funke
Universität Stuttgart, DE
- Glenn Geers
NICTA – Kensington, AU
- Sergio Ilarri
University of Zaragoza, ES
- Benjamin Kickhoefer
TU Berlin, DE
- Franziska Klügl
University of örebro, SE
- Alexandros Labrinidis
University of Pittsburgh, US
- Patrick Laube
ZHAW – Zürich, CH
- Steve Liang
University of Calgary, CA
- Thomas Liebig
TU Dortmund, DE
- Seng Wai Loke
Latrobe Univ. – Bundoora, AU
- Dirk Christian Mattfeld
TU Braunschweig, DE
- Harvey J. Miller
University of Utah, US
- Walied Othman
Universität Zürich, CH
- Nicole Ronald
The University of Melbourne, AU
- Peter Sanders
KIT – Karlsruhe Institute of Technology, DE
- Monika Sester
Leibniz Univ. Hannover, DE
- Sabine Storandt
Universität Freiburg, DE
- Sabine Timpf
Universität Augsburg, DE
- Stephan Winter
The University of Melbourne, AU
- Ouri E. Wolfson
University of Chicago, US
- Bo Xu
Argonne National Laboratory, US
- Yang Yue
Shenzhen University – Guangdong, CN