DAGSTUHL
REPORTS

Volume 6, Issue 5, May 2016

Dagstuhl Reports, Vol. 6, Issue 5 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/dagpub/2192-5283

Publication date
November, 2016

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

Digital Object Identifier: 10.4230/DagRep.6.5.i

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

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

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

Editorial Office
Marc Herbstritt *(Managing Editor)*
Jutka Gasiorowski *(Editorial Assistance)*
Dagmar Glaser *(Editorial Assistance)*
Thomas Schillo *(Technical Assistance)*

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

Report from Dagstuhl Seminar 16191

Fresh Approaches to Business Process Modeling

Edited by

Richard Hull[1], Agnes Koschmider[2], Hajo A. Reijers[3], and William Wong[4]

1 **IBM TJ Watson Research Center – Yorktown Heights, US,** hull@us.ibm.com
2 **KIT – Karlsruher Institut für Technologie, DE,** agnes.koschmider@kit.edu
3 **VU University Amsterdam, NL,** h.a.reijers@vu.nl
4 **Middlesex University – London, GB,** w.wong@mdx.ac.uk

Abstract

Business Process Management (BPM) has significantly advanced and gained high popularity in industry. However, it remains an open issue why tools frequently are used for business process modeling that are not mainly implemented for this purpose. Often, macros for Microsoft Visio or Microsoft Excel form the first choice to capture the flow of business activities. One reason why these tools might be used is the low training effort and the fast creation of a quick model, which can be generated with these tools. Another reason for the "lower" preference of BPM software tools might be their inability to respond to changes in technology and working styles, e.g. the shift towards "agile" processes and the "flattening" of workforce hierarchies that bring more stakeholders into contact with a much broader array of processing steps than before.

A central question is whether the BPM community should create an entirely new paradigm for process modeling. One can think of more intuitive drawing conventions that laymen would use, and of models of an entirely different kind (i.e. not process-centric and not data- or case-centric) that still bear the possibility to support modern and future business process.

The purpose of this seminar was to bring together a cross-disciplinary group of academic and industrial researchers to foster a better understanding of how to ease the access to, and applicability of, business process modeling. We discussed business process modeling approaches against emerging trends such as Internet of Things, the need for incremental and agile creation of new processes, and the need for workers to understand and participate in multiple contextual levels (e.g. transactional, business goals, strategic directions) while performing processes. The seminar also considered how new technologies, such as modern tools for UI design (e.g. D3, node.js) could be applied to support fundamentally shifts in how processes are modeled and how humans are involved with their execution.

Seminar May 8–13, 2016 – http://www.dagstuhl.de/16191
1998 ACM Subject Classification D.2.10 Design, D.3.2 [Language Classifications] Design languages, H.1 Models and Principles, H.1.1 Systems and Information Theory, H.4.1 [Office Automation] Workflow management
Keywords and phrases business process models, process modeling, visualization
Digital Object Identifier 10.4230/DagRep.6.5.1

Fresh Approaches to Business Process Modeling, *Dagstuhl Reports*, Vol. 6, Issue 5, pp. 1–30
Editors: Richard Hull, Agnes Koschmider, Hajo A. Reijers, and William Wong
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Richard Hull
Agnes Koschmider
Hajo A. Reijers
William Wong

Business Process Management (BPM) has significantly advanced and gained high popularity in industry. However, it remains an open issue why tools frequently are used for business process modeling that are not mainly implemented for this purpose. Often, macros for Microsoft Visio or Microsoft Excel form the first choice to capture the flow of business activities. One reason why these tools might be used is the low training effort and the fast creation of a quick model, which can be generated with these tools. Another reason for the "lower" preference of BPM software tools might be their inability to respond to changes in technology and working styles, e.g. the shift towards "agile" processes and the "flattening" of workforce hierarchies that bring more stakeholders into contact with a much broader array of processing steps than before.

A central question is whether the BPM community should create an entirely new paradigm for process modeling. One can think of more intuitive drawing conventions that laymen would use, and of models of an entirely different kind (i.e. not process-centric and not data- or case-centric) that still bear the possibility to support modern and future business processes.

The purpose of this seminar was to bring together a cross-disciplinary group of academic and industrial researchers to foster a better understanding of how to ease the access to, and applicability of, business process modeling. We discussed business process modeling approaches against emerging trends such as Internet of Things, the need for incremental and agile creation of new processes, and the need for workers to understand and participate in multiple contextual levels (e.g. transactional, business goals, strategic directions) while performing processes. The seminar also considered how new technologies, such as modern tools for UI design could be applied to support fundamental shifts in how processes are modeled and how humans are involved with their execution.

2 Table of Contents

3 Seminar Structure and Schedule

The seminar was mainly structured in a working group mode. The schedule of the seminar was as follows:

- *Monday:* intro by participants, panel discussion and working groups
- *Tuesday:* working groups
- *Wednesday:* new group assignment and working groups
- *Thursday:* working groups
- *Friday:* reflection of the seminar

The panel discussion stimulated the participants on the subject of the seminar. The panelists pointed to different views on the seminar's topic. Based on the discussion several working groups were formed discussing the following questions:

- What are purposes of process modeling?
- What are user perspectives in process modeling?
- How is visualization of process models distinct from visualization in general?
- How to implement a purpose-driven visualization of (business) process models?
- What are characteristics of knowledge-intensive processes?
- Are existing BPM methodologies adequate for Agile BPM?
- What can BPM bring to Internet of Things?
- What are technologies supporting collaborative business processes for mutually untrusting partners?

4 Working groups

4.1 How to implement a purpose-driven visualization of (business) process models?

Banu Aysolmaz (VU University Amsterdam, NL), Fernanda Baião (Federal University – Rio de Janeiro, BR), Artur Caetano (INESC-ID – Lisboa, PT), Kathrin Figl (Wirtschaftsuniversität Wien, AT), Jens Gulden (Universität Duisburg-Essen, DE), Julius Köpke (Alpen-Adria-Universität Klagenfurt, AT), Ralf Laue (Westsächsische Hochschule Zwickau, DE), Chris Snijders (TU Eindhoven, NL), Minseok Song (POSTECH – Pohang, KR), and Michael zur Muehlen (Stevens Institute of Technology, US)

We discussed the existence of a gap between the actual process model, made by some creator, and the users of the model who are supposed to derive some benefits from it in order to achieve goals. The main idea we want to emphasize is that there is a need to consider how to best communicate or connect the underlying model to the user, and that this is not a trivial exercise. We also want to emphasize that the same model may need to be communicated in contrasting but consistent ways to different users according to their specific set of concerns and purposes. Measuring the performance of the communication mechanisms is therefore paramount to assess the effectiveness of a model with regards to the purposes of its users. An example might help clarify the issue: the process of checks before liftoff of an airplane would undoubtedly generate a complex process model if represented as a flow of activities, but pilots are instead given checklists and, in principle, see nothing but a simple transformation (view) of the underlying model. In this case, there is an artefact the checklist in between the actual model (process?) and the user. One could argue about whether the checklist is the best way to get pilots to make sure that an airplane can liftoff safely, but the key issues are the following:

- what is the purpose of the visualization: what kind of message or insight or task support should be given to the user,
- what are the relevant perspectives (data, roles, control-flow, time, ...) to be included in the visualization,
- what are the relevant characteristics of the user: different users may have different prior knowledge, cognitive abilities, preferences and even emotional states that affect their performance when using a particular visualization / representation,
- how to best convey the task support or message or insight that needs to be sent to the user, and
- that the best way to convey this message might not be a standard notated process model (e.g. represented with BPMN, EPC).

Creating a process model with another purpose in mind, but then handing it over to the user with another purpose is certainly not always the best solution, and optimal solutions depend on the model, the goal/purpose, the message, and the user. We argue that more attention should be paid to the model user, and how well a visualization fits the intended use/user. A more systematic approach to creating an appropriate message and to evaluate the performance of different ways of creating this message is needed.

Messages a model may convey to users

Depending on the situation, users might need to receive different messages from a model. Not only the user has a purpose, but most often the "sender" has a purpose in mind the user should use the process model for. Without being complete, we could at least discern that a model can be used as:

1. An imperative: "do this"
2. An advice: "it might be a good idea to do this"
3. A goal: "make sure to accomplish this (in whichever way you see fit)"
4. An insight or explanation: "you now see that in our case A always precedes B"
5. A reason: "(you have to wait a bit,) because x steps have to be carried out by others first"

Obviously, combinations of the above could form new message categories as well. The views on a model can be classified according to multiple dimensions (Steen et al. 2004). We can use, for instance, a dimension to classify a model according to its purpose, e.g. designing, deciding, informing, and another dimension to classify the content of the view, e.g. providing the details of a model, focusing on the coherence between the constructs, or just providing an overview of the model. From (Steen et al. 2004) For the BPM community, our guess is that it is most natural to show (variants of the) actual process models (say, a flowchart of some kind) to users, or perhaps parts or condensed versions of them. However, there are many other ways to get a message across. We can borrow from communication sciences and research in human factors to categorize these: the most natural way for a user to receive messages is visually and sometimes through audio, with touch a more remote possibility and smell as a rarity. We easily came up with a large list of mainly visuals:

- Flat text
- Comics
- Checklists
- Decision trees
- Dashboards
- Calendars
- Gantt chart
- Augmented Reality (e.g. Google Glass)
- 3D virtual world
- Maps (e.g. scatter maps)
- Scientific graphs (e.g. histograms, line charts, bar charts)
- Flowcharts
- Controlflow-based models (e.g. BPMN, EPC)
- Goal models
- Organizational models
- Video and animations (visual, possibly with audio)
- Beeps / ticks (audio)

Given different kinds of messages the next question is how they are transported to the user: on paper, on a phone, on a computer screen, as a sticker on the wall, encapsulated in more general tools that are known to the user (such as Excel or Powerpoint), or in still other ways. Some questions that relate to this issue are:

1. Which modes of transportation are more appropriate for which kinds of messages?
2. Which modes of transportation are more appropriate for which kinds of users?
3. How can we transform the underlying model to an artefact (as a vehicle and as a message), depending on the kind of message we want to get across, and how does this depend on the kind of user, and the kind of model?
4. Which kinds of artefacts do users actually use?
5. Which kinds of artefacts have users created for themselves?

What is important to realize is that these issues, although perhaps not yet very well elaborated in the BPM context, are more or less standard in disciplines such as Human Computer Interaction and Communication Sciences. Applying the tricks of their trade to the BPM context in a systematic way would be a good way to proceed. Maybe it would also be an appropriate approach to clarify which user + tasks a process model is suited for and leave other purposes (which are still related to the "process") to other disciplines, to avoid "inventing the wheel" again. In some sense "everything is a process", but the BPM field cannot answer all research questions related to selecting appropriate visualizations, but should concentrate on its main purposes and users. Indeed, the logical next step would then be to consider whether the appropriate goal or purpose of the user (comprehension, compliance, advice, insight/understanding, etc) has been met given the artefact that has been used, and how to measure or compare the usefulness of different artefacts to get this message across, depending on user and model characteristics, and purposes. We now focus in more detail on visualizations.

4.1.1 Visualization of Process Models

Assumptions:

- Models are conceptual abstractions that are constructed from an actual or hypothetical scenario, either tangible or intangible

- Models are designed with a purpose in mind
- Models are typically represented in visual form
- The intended purpose of the model may or may not match the actual use of the model
- Cognitive fit (Vessey) of task and visualization is an important influence factor whether one visualization will outperform another related to relevant performance indicators
- Representations are "not superior in an absolute sense; rather they are good in relation to specific tasks"
- The performance of the visualization of a model must be measurable
- A central criterion for the efficiency of a model is the way it helps to reduce cognitive load that is required for gaining knowledge from it. This can be achieved by appropriately harvesting the capabilities of the human visual perception apparatus for parallel cognitive processing on multiple levels of granularity. The way to determine this appropriateness is to some extent again relative to the purposes and users of the models.
- Different visual representations may show measurable performance differences when applied to the same purpose and to the domain

4.1.2 Purpose-based Visualization

When discussing the user perspective in process modeling, it is important to first reach consensus on three aspects: (1) who is the user, (2) what is the purpose of a process model and (3) What is the current context of the user and the process (how and in which situation is user using the model, how good is the required data to produce the model). Once these aspects are defined, it is possible to reason on what is the best possible visualization of a process model for a given purpose. What do we mean by process model visualization? Visualization is the act or process of interpreting a model in visual terms or of putting it into visible form. It can be expressed as function that transforms a conceptual model into a Sign [cf. Peirce, Semiotics].

Visualization(Model, Domain) → Transformed Model

The Transformed Model can contain several artefacts such as text or checklists on paper, animated models, video or augmented reality. The Transformed Model is used by a User to achieve a specific Purpose within a given Domain. The fitness of Transformed Model is assessed/measured with regards to its Purpose using a set of Variables that are measured/-operationalized using Measurement functions. A Transformed Model often uses a different visual representation than the source model used to generate it. The Transformed Model may omit or aggregate content of the source model and it may enhance it with additional domain information, e.g. purpose specific, domain specific knowledge, explanations. The additional information can also relate to the current context of the user (augmented reality/virtual world). The visualization can be in the form of direct (1:1) transformations from the source model, but it can also aggregate, exclude or change the way source model constructs are depicted, and add overlays, annotations, dynamic content, etc. The target model may be based on a different modeling paradigm i.e. imperative vs. declarative.

4.1.3 Relationship between purposes and visualizations

The crucial point required to have a thorough conceptual basis for formulating scientifically justified methodical approaches for visualization use, or even automatic mechanisms that are able to suggest visualizations depending on a formalized notion of purposes, is to gain an understanding of the relation between the purposes of a visualization and the characteristics of a particular visualization. This requires to conceptualize

1. A (formal) description mechanism (metamodel) for specifying purposes
2. A (formal) description mechanism (metamodel) for specifying visualization characteristics
3. A mechanism that relates both
4. A mechanism to measure the performance of a visualization with regards to the purposes using a set of variables

Such a conceptual architecture can be created on diverse levels of ambition and complexity. In a simple case, the metamodel of purposes consists of a list of use case types, which are derived from observing typical actions performed with process models. A large potential for researching about how to conceptualize purposes remains on this side, as described in earlier sections.

The metamodel for describing visualization characteristics might in the first place provide a coarse conceptualization of available visualization widget types on a coarse grained level, such as piecharts, barcharts, and other state of the art dashboard components. If visualizations get conceptualized in a more sophisticated way, their metamodel could refer to more abstract visual impressions such as symbols and structures they appear in, or even incorporate phenomena of human visual perception, such as the perception of patterns, perception of spatial distributions, etc.

Finally, a mapping conceptualization could in the first place be conceptualized as a mere relationship between instances of elements from the purpose metamodel, and elements from the visualization metamodel. A deeper going approach would be to find scientific judgements about the plausibility of possible mappings, e.g., in terms of the expected cognitive load of the use of a visualization type for a given purpose. A mapping conceptualization which takes such criteria into account would provide an actual advance in the way how visualizations are conceptualized today, and how the efficiency and effectiveness of their use can be assessed.

References

1 Maarten W. A. Steen, David H. Akehurst, Hugo W. L. ter Doest, Marc M. Lankhorst. *Supporting Viewpoint-Oriented Enterprise Architecture*. EDOC 2004: 201–211

2 Sander G. van de Wouw, Hajo A. Reijers. *An Integrative Framework of the Factors Affecting Process Model Understanding: A Learning Perspective*. AMCIS 2010, paper 184

4.2 What are characteristics of knowledge-intensive processes?

Achim D. Brucker (University of Sheffield, GB), Fernanda Baião (Federal University – Rio de Janeiro, BR), Avigdor Gal (Technion – Haifa, IL), Alexander Herwix (Universität Köln, DE), Richard Hull (IBM TJ Watson Research Center – Yorktown Heights, US), Massimo Mecella (Sapienza University of Rome, IT), Hamid Reza Motahari Nezhad (IBM Almaden Center – San Jose, US), Flávia Maria Santoro (Federal University – Rio de Janeiro, BR), Tijs Slaats (IT University of Copenhagen, DK), and William Wong (Middlesex University, GB)

Main reference C. Di Ciccio, A. Marrella, A. Russo, "Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches," J. Data Semantics, 4(1): 29–57, 2015.
URL http://dx.doi.org/10.1007/s13740-014-0038-4

Knowledge-intensive processes (KiP's for short) represent a new trend in the world of business process management, which was enabled by the availability of big amount of data and tools to to transform such data into knowledge. KiP's also represent a possible solution to a world where processes are no longer structured, long standing, and carefully designed. To accommodate agile processes, which continuously evolve and have to deal with missing, uncertain and erroneous data, we go back in this work to the basic elements that would allow us to better understand what KiP's are, what can they do for us, what type of technology will be at their basis, and what may a plausible research agenda for KiPs.

4.2.1 Use cases

There are many processes which are not well addressed by state-of-the-art process-aware technologies and support systems. Essentially, they share (to a certain extent) the set of requirements (further detailed in this document), which characterize them as KiP's, as it follows:

- Perform a scientific experiment. Scientific experiments are typical scenarios that explore a hypothesis space, frequently embedding what-if analysis and analytical operations. Typically, the scientist defines a sequence of (automated) steps to validate his/her hypothesis and specifies a (often huge) set of scenarios, varying the input dataset to be considered, the parameters of each step execution, or different resources. There is a need to keep track of all scenario variations, decide on the next new scenario to be experimented – and explain why, collaborate with other scientists from the same research team, or from other teams. Provenance data plays a major role, and may serve as input data from which knowledge may be derived.
- Design a marketing campaign. Marketing is a typical creativity-driven process, meaning that there might be interesting scenarios in which the executor come up with new activities that were never executed before. This surely points to the flexibility requirement, but goes beyond it by ideally deciding on innovations of the process through new activities conducted by the agent intentionally to reach a goal.
- Perform medical diagnosis. Medical diagnosis is a typical goal-driven, decision-making scenario. In several cultures, procedures and protocols are very well-regulated and, to a certain level of granularity, well-defined. However, there are relevant scenarios (such as "Doctors without borders" or emergency treatments) in which those pre-defined routines are not effective or the required resources are not available during execution. In these situations, collaborating with different partners so as to decide upon the treatment or searching for new external data/knowledge are frequent actions doctors intentionally do to reach their goal. Moreover, they would be happy to share their new experiences and explain their decision-making process on both successful and unsuccessful cases.
- Air traffic control. Air traffic control scenarios are typically procedure-based, but frequently require context-based adaptations, thus requiring decision-making and a flexible execution environment, so as to cope with real-time adaptations.

4.2.2 Requirements

A KiP is a type of process that comprises sequences of activities based on intensive acquisition, sharing, storage, and (re)use of knowledge, so that the value added to the organization depends on knowledge. One of the main characteristic or requirement of those processes is having an unpredictable flow of activities, i.e., changes might occur from one instance to another. They should be flexible enough so that the next step is typically defined by decisions made after the completion of the previous activity. We consider the following requirements essential to knowledge intensive processes:

- goal oriented, knowledge about the domain available or learned along a specific instance;
- flexibility; adaptiveness;
- possibility to make the rationale for decisions made explicit;
- and collaboration among actors (human or machines).
- No apriori-defined process: The activities to be performed might not be known in advance as well as the flow to be followed. Activities could be defined in real time as a result of a decision. In the case of flows (the set of interrelated activities), the definition could be supported by templates already known or existing best practices. That is why adaptations should be possible and this also requires that the process should be continuously monitored.

Given the goal-oriented nature of KiPs, and decisions are made about actions in the course of a process, it is fundamental that decisions made could be captured and represented, so that, as a result they could also be converted in new knowledge to be consumed by the process. Finally, although not a strict requirement, collaboration is expected to take place in order to foster the exchange or even the creation of new knowledge within the process.

4.2.3 Main Elements

While the notion of KiP is broad, there is set of building blocks that are common to most of them. This section introduces these core building blocks and describes each of them briefly. The next section places these elements into a conceptual model, with an emphasis on how they relate to each other.

- Goal/Subgoal: A central component of KIP's is the set of goals and subgoals that are to be achieved. We use the term 'goal' to refer both to the overarching goals of a process and to subgoals used along the way. These goals might include concrete, world-affecting goals such as determining how much money should be transferred and to whom, allocating appropriate resources to an activity, or the creation of a document for distribution. They can also include more "intermediate" goals focused on acquiring data or knowledge relevant to the overall process, or the identification of agents (human or machine) that can contribute towards the process. The family of goals typically evolves over time, with goals being achieved, goals being created, and goals being deleted (because they are deemed irrelevant based on other accomplishments.
- Data & Knowledge: A central aspect of KIP's is the accessing of relevant data and knowledge "in the world", and also the creation of new data and knowledge relevant to the goals at hand. Data and Knowledge lie along various dimensions. One dimension ranges from structured data to unstructured content (including text, image, video). Another dimension ranges from factual data to structured knowledge representations (e.g., OWL) to modeling formalisms (e.g., UML) to knowledge captured in documents to tacit knowledge (i.e., in people's minds). The knowledge may be produced through various mechanisms, e.g., asking an expert, ingesting documents manually or by machine, machine learning, etc., and may have varying levels of confidence, precision, etc. Yet another aspect is that some knowledge relates to the actions that might be taken (e.g., pre- and post-conditions), "best practices" that have evolved from previous process executions, and patterns or templates involving combinations of actions. During a KIP some knowledge may be captured into a machine readable form that permits automated reasoning, while other knowledge may be provided for human consumption, with humans making further inferences and conclusions – which may or may not be recorded explicitly.

- Decisions: Because of the prominence of knowledge in KIP's there is an associated prominence attributed to decisions. These decisions might be in the form of choosing to invoke a world-affecting action, such as paying money, allocating resources, or requesting assistance from some agent. The decisions might also be in the form of choosing a single action to be taken as part of the process, or choosing a "pattern" or "template" of actions to be taken, possibly with some parameters assigned. Decisions might also create new goals or determine that some existing goals are no longer relevant.
- Action: In the context of KIP's, actions will include both the types of actions typical of standard business processes, and also actions relating to the acquisition and management of knowledge. The latter category includes actions such as making decisions based on the available knowledge, acquiring additional knowledge, and reasoning about knowledge to create additional knowledge. This knowledge creation might be achieved through a variety of mechanisms, including human thought, automatic reasoning, machine learning, hybrids of the above, etc. Importantly, the family of possible actions may be known to the actors in the system, but it is also quite possible that the full family of actions is not known a priori, and potential actions are discovered along the way.
- (Sub)process Pattern/(Sub)process Template: In many cases it is convenient to classify groups of actions into re-usable, parametrized patterns. This permits a modularity in the process specification, which is useful for both manual and automated reasoning and selection. These patterns might be organized based on existing process management standards or paradigms (e.g., BPMN, CMMN, or declarative ones such as DECLARE, or more informal ones such as Task Lists), or might be more ad hoc in nature.
- Agent/Actor: It is typical of KIP's that multiple agents are involved; these may be human or automated. The agents may have specialized roles, which may be interchangeable. Often the agents may collaborate on achieving goals, or work towards a consensual understanding and agreement about certain information or decisions. In many cases agents will need to communicate with each other in rich ways, including intricate requests (e.g., using "conversational APIs") and transferring rich knowledge. Reasoning about what agents know may be relevant, e.g., if expecting an agent to make a fully informed decision. This is an area where explanation of the activities performed, goals attempted, and decisions made will be very important.
- State/Context/Environment: KIP's typically progress in terms of their internal state and body of goals, plans, and knowledge. KIP's are also operating in an external context of environment. We use the term 'context' to refer to those aspects of the external environment which are relevant or potentially relevant to the process. The context will be evolving, both as a result of KIP actions and due to independent forces. The KIP may pull information from the environment from time to time. In some cases the KIP will include continuous monitoring of the context, with the possibility of actions (including decisions) being triggered by phenomena
- Event: As with most kinds of process, events are a central to KIP's. The events might occur from the environment, including both discrete events and also defined thresholds being exceeded. The events might be internal to the KIP, e.g., when an automated decision process completes its activity and yields a decision value. Events may trigger immediate responses in the KIP, or might lead to the accumulation of knowledge which impacts the KIP later on.
- Exploration/What-if: A particular kind of decision making that arises in KIP's involves the exploration of various possible scenarios, as a component of comparing different alternatives. These comparisons might be short-lived (e.g., minutes) or longer-lived

(e.g., days or months). The information/knowledge obtained about different alternatives contribute to the overall knowledge base of the KIP, and may be used to inform subsequent decisions

4.2.4 Research questions and fresh directions

To support knowledge workers of the future, the state of the art in modeling and executing knowledge-intensive business needs to be advanced in several areas. For example:

- Hybrid manual/automated process models: Knowledge-intensive processes often consist out of small routine tasks or sub-processes that can and should be automated to enable users to concentrate on the parts that need flexibility, creativity as well as "on-the-fly" gathering of information. New approaches are needed for modeling such hybrid models in a uniform way that still allows to specify the overall goals and requirements of a knowledge-intense business process.
- Dynamically evolvable process enactment models. There is a need for defining a rich data model, considering the main elements outlined above, for knowledge-intensive processes to enable both modeling of such processes, as well as enacting them, at run time. Such a modeling approach should allow adding and updating the set of available actions, operations on the model, process flow and fragments, and instantiation from best practices and more abstract processes, in a dynamic manner. In addition, the underlying goals and constraints of the process should be easily adaptable to changes in the context (e.g., changes to laws and business rules) and knowledge (e.g. when new medical treatments are discovered).
- Integration of knowledge information systems: Knowledge-intensive business processes depend, by definition, on a large set of information. The exact amount and type of information might depend on the actual execution of a business process as well as the situational context (e.g., the environment). Thus, new approaches for presenting information, in the context of the business process execution are needed that allow the user to access the situational relevant (i.e., all information that is required to take an informed decision or execute a next step/tasks in the process and, in parallel, avoiding an information overload).
- Compliance analysis of "incomplete" models: Knowledge-intensive business processes need to allow for certain flexibility in their execution to allow users to react on unforeseen events. Thus, they usually cannot be represented by a complete and detailed model that contains all tasks/subtasks. At the same time, knowledge-intensive processes are often needed in domains that are need to fulfill strict audit on compliance regulations or are safety or security critical (e.g., air traffic management, merger and acquisition processes, disaster management processes). How to statically analyze (or reason about) such incomplete models to ensure that they are actually executable and ensure compliance during their execution (respectively, fulfill the basic safety or security requirements) is an open problem. Alternatively, research in enriching models with aspects of run time monitoring (logging) as well as enforcement seem promising to enable a "post-hoc compliance" approach.

4.3 What are user perspectives in process modeling?

Artur Caetano (INESC-ID – Lisboa, PT), Christian Janiesch (Universität Würzburg, DE), Jan Recker (Queensland University of Technology – Brisbane, AU), Chris Snijders (TU Eindhoven, NL), Victoria Torres (Technical University of Valencia, ES), Barbara Weber (Universität Innsbruck, AT), William Wong (Middlesex University, GB), and Michael zur Muehlen (Stevens Institute of Technology, US)

Models are (shared) representations of some actual or desired system created by an individual or a group. Process models capture the behavioral aspects of a system, and are used to explain, predict, design, enact or constrain past, current and/or future situations. It is important to realize that models are constructions by an individual or a group and intended for use by a (different) individual or a (different) group. Sometimes, these individuals might be computerized agents but more often than not they are human agents or a combination thereof. The creation and use of process models are sometimes distinct activities, sometimes not. The goal of some models is to be consumed as they are created (e.g. to create a shared understanding among a group of participants), other models are invariant upon creation (e.g. standards, reference processes or procedure manuals), others are transformed after an initial stable state has been reached (e.g. a model that is re-designed for optimization purposes, a conceptual process model that is subsequently used for the configuration of a workflow system), and yet others are created but are intended to be used only rarely, such as evacuation procedures or the preparation for emergency cases. In each of these scenarios, different people are involved in the creation, validation, and use of the models. Sometimes, not the individual user but a group of users, e.g. a team or a department, is formative to the process. Therefore, better understanding of the different types of users, their perspectives, and how they influence the creation and use of process models may result in better approaches to process modeling and to models that fit better with the users' needs.

4.3.1 Clarifications Required to Understand the User Perspective

When discussing the user perspective in process modeling, it is important to first reach consensus on two aspects: (1) who is the user and (2) what is the purpose of a process model, as described next.

4.3.2 Who is the user?

1. Which roles does the user play in relation to the model?
   Creator (e.g. information provider, modeler), reader (e.g. interpreter, consumer), transformer (e.g. implementer, enactor, redesigner), and so forth.
2. How can the model user be distinguished?
   - What cognitive characteristics does the user have? (e.g., working memory capacity, executive functions, ability to learn, etc.)
   - How experienced is the user? (in understanding the domain and the system of interest being modeled, in understanding the modeling method and supporting modeling tools, in performing the actual modeling process, in understanding the model constructs and semantics).

   - Which other static (e.g., personality) or dynamic traits (e.g., knowledge, attitude, user context) are relevant to consider?
   - Does the user correspond to a specific archetype (i.e., a persona)?
3. How does a user create, interpret, use, and adapt the model? Possible approaches to answer this question will likely be exploratory in nature, with the objective to discover usage strategies, difficulties, value and importance, as well as concepts and constructs that can inform subsequent measurement, and include:
   - Cognitive Task Analysis (CTA) of users can be conducted while they create the models, use them (concurrent CTA), and also after (retrospective CTA).
   - Observational Studies of users can be carried out by combining various techniques such as Think Aloud methods (Concurrent Protocol Analysis) to determine the issues and difficulties, considerations and strategies they encounter during the creation and use of the models.
   - Neurophysiological methods such as eye tracking, EEG, heart rate monitoring and galvanic skin response measurements, for assessing or identifying and correlating stressors in the model comprehension and creation. This can include identifying difficult or challenging situations, or frustrations with the model understanding and creation, and how they correlate with the performance of the task. Correlates can help to better understand antecedents of (IT) behavior (e.g., stress as an antecedent of an error).
   - Observation Studies and Tracking of user interactions while using or creating the system (e.g., model interactions, mouse events, scrolling, click streams). A longer-term goal of tracking user interactions is to be able to directly map the user behavior to measured variables of interest such as acceptance, userfriendliness, etc.
   - Ethnographic Studies of users interacting with artifacts resulting from the model (systems, policies, procedures, instructions) (e.g., creation of workarounds as opposed to following model(er) intent, time to change existing behavior, contemporaneous commentary on perceptions regarding the new situation).
4. The difference in the fundamental nature of work that information users do will influence the nature of the models that need to be produced to support that work:
   - As clerical workers, they are not informed by the information and their work is dictated by the procedures and policies of the organization, e.g. the completion of forms, the input of data to complete a task;
   - As knowledge workers, their roles focus on the creation and construction of new understandings as a result of being informed by the new information. This new understanding enables the formulation of solutions and problems.
   - This difference suggests that models required to support clerical type of work will necessarily be more transaction-oriented, while models needed to support knowledge workers will need user interfaces in the resulting systems that support the creation of understanding, methods for sense-making, and ways for defining new strategies and tasks.

4.3.3 What is the purpose of a process model?

Can the purpose of modeling be defined concludingly before the start of a business process management project or any other initiative that involves process modeling? If so, in which situations should the purpose be defined a priori or be emergent? Should or can the purpose be derived from the needs/concerns of the users as concerns may be contradictory? The purposes of a process model may include the following:

- Informal purposes, e.g., understanding, communication, brainstorming.
- Semi-formal purposes, e.g., documentation/knowledge management, training.
- Formal purposes, e.g., software development, workflow execution, artifact evolution (transformations).

It is, however, important to consider that there are previously undefined, unanticipated and/or exploratory purposes, e.g., re-purposing, which should not be excluded prematurely in order not to restrict the user's ingenuity with regard to process modeling.

The question of what the modeling purpose is and who the model users are, is closely linked to the question of what defines the utility of a model. Possible constructs to operationalize and measure model utility include understandability, shared understanding, consensus, executability, effectiveness, acceptance, cost/benefit-ratio and task performance.

4.3.4 Future Research on the User Perspective in Process Modeling

Based on our interpretation of the literature on business process management and in particular on process modeling to date, we believe that several questions remain unanswered, incomplete or simply unbeknownst until now. Partly, this may or may not be related to the fact that there are not enough established (and thus universally accepted) research methods in Information Systems to deal with these questions. In the following, we present several of these questions:

1. Are there differences between models that are used or created for groups of users and those that are created or used for and by individuals?
2. How do we better understand the mental model of the users? Which factors influence and/or support model creation and understanding? Can available methods be applied to a single user or also to groups of users? How can we capture and translate user expertise into the models?
3. How, when, and why is a user affected directly or indirectly by a process model? For example, does a user interact with the process model, or does the user interact with a process that is defined by the model, or with a system that is being controlled through the process model? Are there users that have an interest in the process model but neither interact with the model nor the resulting process?
4. What roles does a user play in different process modeling phases, such as information gathering, (re-)designing, using, changing, optimizing, etc.?
5. What are appropriate measures to assess the alignment of model creation and use with the user perspective?
6. How do we observe the impact of using these measures? Which of these have we not considered at all so far? Are the current measures adequate?
7. How can we support/advise organizations to consider and implement a user perspective in their business process management efforts? Can we generalize universal guidelines? Would a "user-perspective process maturity model" be relevant? Do current modeling standards satisfy the needs of users and the needs of organizations? Do they need to be adapted before use? If so, how can this process be guided.
8. What are the differences between process design and process modeling in terms of the activities performed and the user perspectives on the processes and outcomes? Is the user perspective on process modeling different from the user perspective on process design? Could we support both with similar methods, tools and systems? If not, what makes process design different from process modeling?
9. What are the differences between user-centric and use-centric modeling? Often, one fundamental question in design is what we design for? Designs based on user actions and user's mental models can be overly restrictive for a number of reasons, such as users having

a flawed mental model. As such approaches to design that incorporate use as a focus of the design and modeling process have a greater opportunity to capture the affordances, expectations, and factors that constraints the nature of the work and their relationship to higher order goals of the organization. How can business process management incorporate and test the effectiveness of such an approach? How can the outcomes of procedures and processes be understood from the perspective of an organization's higher order goals, such as profitability, efficiency or market competitiveness?

10. What can we learn from other fields that examine human perspectives, such as Human-computer Interaction, Cognitive Systems Engineering, User-centric Design, Technology Appropriation, Human-in-the-loop, Human-Performance Management, Cognitive Psychology, Neuroscience

4.3.5 Emerging principles

In answering these and related questions, we also find that several principles emerge that can govern user perspectives in process modeling. Importantly, depending on the purpose of the model, the type of user and the domain of the system, we may find.

A need to adequately interface the model to the user, i.e. techniques that "translate" the information from the underlying model for a user in a way that is more appropriate for them (e.g. checklists are used in aviation as an interface to a far more complex underlying process model). This implies that

- the representation techniques should be aligned with the concerns of the users of the model
- the same process model can be viewed from multiple perspectives according to the concerns of its users, and
- we would expect different representation techniques to show measurable differences (e.g. different levels of comprehension, different levels of reusability) in constructs that are important to the purpose of the respective user group.

Different types of model usage, such as models defined for executing just once or never (as is the case with evacuation processes which not meant to be used often), models meant for training people, models for ensuring that quality standard levels are reached, or models for building consensus between stakeholders. We would expect that the effort of model creation and consumption can be viewed in relation to the utility derived from the creation and consumption of the model. This may require to define a utility function over time for models with a long "shelf life".

Differences in assisted/guided versus unassisted processes for process modeling. Developing an assisted process for model creation may lead not only to syntactically correct but also to better models, i.e. models that fulfill the purposes of its users:

- Enabling the user to create models in a guided but generally unrestricted fashion and facilitating the purpose these models may lead to an increased uptake and training of process modeling and improve model use, user satisfaction as well as model quality.
- We would expect that articulating the "meta-process" of framing the modeling purpose prior to modeling leads to a measurable increase in model utility.

Different ways to operationalize and to measure the quality of a process model (e.g. shared understanding, comprehension, etc.). This may lead to the definition of indicators, measures, maturity levels, etc. A framework that may be useful for the user perspective in process modeling may be the emergence of work routines and processes over time. This is because we expect that not only processes and their models evolve but also the types and communities of users that engage with these models. The framework differentiates three distinct stages:

1. Magic: Someone figures out how to do something. The details of the work practice or process are non-transparent to outsiders. So, the model of the work is either unknown, or the result of the work violates the preconceived notions that were built on existing models.
2. Heuristic: Over time, rules/constraints/sequences emerge that achieve the outcome in a more consistent/predictable fashion. The process becomes discernible and repeatable. A model at this stage represents a rough approximation of the work being conducted, but certain aspects as to "why" things work remain hidden. Also, models at this stage may not be generalizable but only cover specific scenarios such as the normal routine that works in most cases.
3. Algorithmic: The process can be described in a formal and systematic fashion, and repeated even by those unfamiliar with the domain or without any human involvement at all. A model developed at this stage is generally detailed and precise, and often fit for machine consumption.

4.3.6 Implications

Two sets of implications flow from our discussion. The first set relates to the way the user perspective is incorporated into research conduct and reporting about process modeling. We suggest a set of recommendations for business process management research conduct and reporting:

- Identify and name the roles of (intended) users or user groups in focus.
- Delineate the specific purpose of the model in focus.
- Identify the relevant study variables of interest that relate to a user or user group, such as purpose, goal, ambition, value, utility, experience and so forth.
- Clarify whether a study is about the model, the process being modeled or the process of modeling itself.
- Separate process design from process modeling. That is, decouple the act of creating a design for a process (what should it do, how should it perform, what options should it encompass) from the act of creating a model that captures this design.

The second set of implications relates to the way research studies are designed and executed to examine user perspectives in process modeling. Specifically, we suggest to encourage presently under-utilized methods of research that are particularly well-suited to developing an understanding of the user (as well as groups thereof). Examples of research methods that suit this purpose, in our view, include

- Ethnography / descriptive field studies
- Inductive field studies
- Qualitative empirical studies
- Development of measurement scales for use in a business process management context to facilitate quantitative analysis
- Usage of multimodal data collection to enable triangulation It is important to state that we do not argue that no such research exists. Our point is merely that we believe more such research should be conducted, and our academic community should dedicate more efforts to encourage, incentivize and appreciate such modes of research.

4.3.7 Conclusion

We found that our engagement in the topic raised several important questions about definitional, procedural, methodological, and theoretical matters. Designing novel, fresh approaches to process modeling will be a consequential action enabled by seeking answers to some of the questions and matters we have raised. Hints at these answers can (perhaps should) also be sought in neighboring disciplines. We believe that drawing attention to the user perspective will (re-)fresh how we approach process modeling in future research.

4.4 What are technologies supporting collaborative business processes for mutually untrusting partners?

Soren Debois (IT University of Copenhagen, DK), Marlon Dumas (University of Tartu, EE), Tijs Slaats (IT University of Copenhagen, DK), Christian Stahl (TWT GmbH – Stuttgart, DE), and Ingo Weber (Data61 / NICTA – Sydney, AU)

Main reference I. Weber, S. Xu, R. Riveret, G. Governatori, A. Ponomarev, J. Mendling, "Untrusted business process monitoring and execution using blockchain," in Proc. of the 14th Int'l Conf. on Business Process Management, LNCS, Vol. 9850, pp. 329–347, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-319-45348-4_19

This working group has considered applications of distributed ledger technology (e.g., blockchain) in the context of the execution of distributed, collaborative business processes involving mutually untrusting partners. Distributed ledger technologies solve a variant of the distributed consensus problem in which processes have to achieve consensus on a history of events: "What happened?". The canonical application and example is Bitcoin, where the question of "what happened" amounts to "how was the money spent". Historically, Bitcoin exploited the blockchain distributed ledger to solve the problem of double spending in virtual currencies without a central trusted authority. When every participant is aware of the ledger, no participant can spend digital currency twice, without the other participants discovering this double spending. Contrast this to the functioning of, e.g., credit cards, where credit companies serve as a single trusted party ensuring absence of double spending. Bitcoin, in contrast, relies on no single trusted party. The problem of determining canonically "what happened" is acutely relevant for distributed execution of business processes in a setting of mutually untrusting partners. This problem extends naturally to the one of ensuring that "what happens next" is in accordance with the underlying business process.

4.4.1 Potential Applications

The Working Group envisions three applications of distributed ledgers (DLs) in the context of collaborative business process execution:

1. DL as an audit trail. Since no participant can realistically change the transactions in the DL after it has been accepted by the network, the DL can serve as a trusted audit trail. This audit trail can be consulted in case of disputes, for example if the customer states that they have not accepted a given purchase order for which an invoice has been issued to them, or if they contest the contents of that purchase order. Note that in this case, it is not necessary for the blockchain to store the entire contents of the data objects manipulated in the process hash codes suffice if the parties are able to produce the original objects in case of an audit.

2. DL as a monitoring mechanism : A “smart contract” (i.e. a piece of executable code executed by the DL network) can verify and/or enforce the constraints of the process model. E.g., the smart contract can reject an invoice submitted by a vendor to the DL if no corresponding purchase order exists in the DL.
3. DL as an active coordination mechanism. While “smart contracts” are by definition restricted to acting on data in the DL, they can be connected to outside systems by running “trigger” nodes acting (outside the DL) upon transactions recorded in the DL via a smart contract. E.g., the smart contract asserts a purchase order in the DL, a trigger a the vendor picks this up, and submits a “ship” event back to the smart contract.

4.4.2 Trust and Privacy

DL technologies entail questions linked questions of trust and privacy. There is a spectrum of trust. If the participants are fully mutually untrusting, a public DL should be used. However, a public DL, especially the current Ethereum Smart Contracts implementation, leaks information: The process being executed is itself publicly available in the form of Etherium bytecode. The details of execution of a particular instance are similarly publically available (with the caveat that information that does not form the basis of decisions in the process can of course be encrypted). It the participants would prefer to keep the process execution itself private, they could run a private DL. This approach implies a relatively high degree of trust among participants: all are allowed to see all collaborative process variables, know the account holders, etc. In between, there may be cases where permissioned blockchains could be most appropriate.

4.4.3 Conclusion

The working group contends that the application of distributed blockchain technologies is a fresh avenue of research for BPM research, with the potential to provide practical solutions for achieving trustworthy process executions in collaborative execution of processes between mutually untrusting process participants.

4.5 Are existing BPM methodologies adequate for Agile BPM?

Marlon Dumas (University of Tartu, EE), Udo Kannengiesser (Metasonic GmbH – Pfaffenhofen, DE), Agnes Koschmider (KIT – Karlsruher Institut für Technologie, DE), Ingo Weber (Data61 / NICTA – Sydney, AU), and Liang Zhang (Fudan University – Shanghai, CN)

Business processes, especially knowledge intensive ones, are subject to frequent changes in their environment (e.g. changes in demand or in customer expectations) as well as unforeseen circumstances that may require special treatment. Effectively responding to such changes and special circumstances has become an imperative in modern organizations, calling for agile approaches to Business Process Management (BPM). These approaches depart from traditional “rigid” process management approaches that start from the premise that the process is planned in advance, that the plan is documented in detail, and that the plan is enforced by setting up normative policies and procedures and/or by instrumenting information systems in order to enforce “the plan”. We define Agile BPM as the practice of managing business processes in a way that fosters adaptability. In this setting , process adaptability is the ability for a process to adapt to changes in its environment fast and effectively. Adaptations of a process can occur at different levels of granularity, ranging from adaptations of an entire process including all its related (sub) processes and all instances thereof, to adapting parts of a process (e.g. specific subprocesses or activities), down to adapting one instance of the subprocess to requirements arising from a given situation. Arguably, the most adaptable process is the one that can be freely executed in any way by the process performers, such that process workers are allowed (and encouraged) to perform every process instance and every activity in any way they see fit taking into account the circumstances surrounding the case (i.e. in a "casebycase" manner). The least adaptable process on the other hand is the one that is performed always in the way it has been planned and where process workers are encouraged or forced to follow the plan to the letter. A key element to achieve process adaptability is the type of schematization of the process meaning the level to which process workers implicitly or explicitly follow a plan. In this respect, we observe different types of schematization, ranging from the most liberal to the strictest one:

- Entry schematization: The creation of an instance of the process follows a (data) schema – e.g. a form. This is generally the case in an application to approval processes, where the creation of a new instance of a process requires the customer to fill in a specific form, or in an issue to resolution process where the creation of a new instance starts with the creation of a ticket in an issue tracking system.
- Milestone schematization: At this level, the process is divided into phases delimited by milestones. For example, an application may reach a "presubmitted" or "validated" phase before being assessed. An issue on the other hand may reach an "accepted" milestone and later on reach a "closed" milestone.
- Activity schematization: At this level, the set of possible activities of the process are enumerated, scoped, defined, assigned to roles, and possibly grouped according to phases. For example, in an issue to resolution process, one will find activities such as "Investigate issue", "Isolate issue", "Propose resolution", "Discuss resolution with customer", and "Test resolution". Note that in addition to activities also events could be defined e.g. "Issue escalated", "Issue closed", etc.
- Workflow schematization: At this level, dependencies between activities and/or events are visible – e.g. one activity is triggered when another activity is completed or an activity might be triggered following a certain event and if certain conditions are fulfilled. Importantly, schematization may occur even in the absence of an explicit representation of the schema. It may be that in some cases that the schema of a process (e.g. the workflow) is not explicitly captured but simply exists by convention. It is also important to note that there is a tension between agility and other dimensions of business process performance management. Attempts to increase agility may come for example at the expense of quality. For example, in a complaint to resolution, if two complaints from related customers are handled differently despite being essentially identical, this may give rise to customers perceiving a lack of equal treatment. Similarly, agility can come at the expense of transparency if every case is allowed to deviate in an ad hoc manner, we lose the global picture of the entire process.

A central tradeoff of agile BPM is how to determine the right type and degree of schematization for every process, which maximizes adaptability while achieving performance targets with respect to other performance dimensions. In other to strike the above tradeoffs, organizations need to rethink the way they approach business process management. Below we discuss a set of "values" that can be used to guide a transformation from a rigid BPM culture to an agile one.

4.5.1 Values of Agile BPM

An existing framework can be used to define basic values of agile BPM approaches. This framework is called the “manifesto for agile software development” (2001). Its applicability to the development of business processes (or “the process of process modeling”) was discussed in a panel session at BPM 2015. The values included in the agile manifesto (adapted to BPM) can be stated as follows:

1. Individuals and interactions are valued more than rigid procedures and tools: This value statement emphasizes the importance of selforganising teams, individual creativity and adhoc communication, in the way carried out in many design thinking approaches. Heavyweight procedures and complex tools and notations are seen as obstacles for creative process development.
2. Executable processes are valued more than comprehensive process diagrams: This value statement stresses the importance of experiencing the results of process modeling: the actual processes as they are executed. The executed process (even if it is only “simulated” through physical or software supported role play) is seen as more effective than analyzing a comprehensive process diagram.
3. Stakeholder collaboration is valued more than upfront process specification: This value statement highlights that all stakeholders including the process participants need to be included in the modeling process. This should ensure that what is modeled correctly represents the stakeholders’ needs and perceptions of the process. Upfront specification, in contrast, limits stakeholder participation to a separate, upstream phase of elicitation. What happens in the subsequent phases is then no longer influenced by them and therefore risks losing alignment with their needs.
4. Responding to change is valued more than following a plan: This value statement is about accepting that process changes will occur rather than assuming that everything that was initially modeled remains the same.

4.5.2 Research Questions

In our discussions, we collected a set of research questions through brainstorming and subsequent refinement. The result is the following list:

- Are existing BPM methodologies adequate for Agile BPM? For instance, do existing lifecycle models still hold? If not, what is the best methodology for Agile BPM?
- What are dimensions and important characteristics of agility in BPM? What are the variables that drive the answers to the questions below?
- To what extent should we schematize, when do we have “sufficient” schematization? How do we determine the minimal requirements stemming from, e.g., needs in terms of efficiency and regulatory compliance? How does the purpose influence the appropriate level of schematization?
- How is the schema created and updated? When should we model, and when should we derive schemas from data and observations?
- How to respond to changes or the need for changes?
- How do you learn about the need for change and understand what needs to change?

4.6 How is visualization of process models distinct from visualization in general?

Jens Gulden (Universität Duisburg-Essen, DE), Banu Aysolmaz (VU University Amsterdam, NL), Soren Debois (IT University of Copenhagen, DK), Ralf Laue (Westsächsische Hochschule Zwickau, DE), Tamara Mchedlidze (KIT – Karlsruher Institut für Technologie, DE), and Minseok Song (POSTECH – Pohang, KR)

The working group observes that many contemporary visual process notations only tenuously connects spatial and semantic relations. Many such notations are graphs (sets of nodes and edges), and the visual representation of these graphs exploit spatial relations only in so far that an edge is represented as an arrow from a box representing the source node to a node representing the target node. The relative positioning of, e.g., the nodes themselves often has little or no semantic significance. Practitioners will often compensate by choosing layouts where one axis corresponds, somehow, to causality or the passage of time. This is the situation for, e.g., BPMN, DCR, and DECLARE, The Working Group contends that BPM notations has room for improvement in comprehensibility by a further strengthening of the connection between spatial and semantic relationships.

4.6.1 Visualisation, comprehension, and abstraction

A common complaint of BPM diagrams (in any notation) is the "But this is too complex!" outcry. The Working Group maintains that this complexity is not a property of visualization, but rather a property of the processes we model: Were they not complex, we would not bother to model them. Process models typically serve as an integrating perspective on knowledge that has references to other perspectives. E.g., the specification of a task may reference to accompanying actors and/or resources. As a central integrating perspective, the process model thus inherently is embedded in a multidimensional structure of relationships, for which to provide cognitive access is the main purpose of a process visualization. This complexity manifests itself, e.g., in declarative modeling notations primarily as an implicit need that the visualization consumer having to reason about the ramifications of the rules represented visually. This is unacceptable; computation is for computers. The complexity manifests itself in "imperative" modeling notations primarily as unacceptable growth in (visualized) model size. If this complexity is inherent to the process, then any visualization of that process must contain that complexity. The only way to remove complexity from the visualization is to remove it from the process. Whether or not removing complexity is feasible and reasonable depends on the exact reason the process is being visualized in the first place. The consumer of the visualization will have a goal in looking at the visualization; usually, that goal will not require every last detail of the process model, and so some removal of detail is possible. Being able to provide this ability is a requirement specific toward process visualizations, which is not demanded from visualizations in general. We emphasize that the exact means of removing complexity is immaterial: It may be as simple as filtering certain elements from the visualization, or as complex as performing model transformations before visualizations. Thus, the Working Group contends that the useful visualization of process models is inextricably linked to on the one hand to the questions the visualization is supposed to help answer, on the other our ability to suitably simplify the model before visualization. There can be no useful visualization without abstraction.

4.6.2 Visualization of Process Models

To examine the particulars of process visualization, the Working Group proposes considering the problem from three angles:

1. Use cases.
2. Requirements.
3. Examples.

4.6.3 1. Use Cases

Below is a nonexhaustive list of potential use cases for visualizations. Note the overlap with the purposes of having a process model in the first place. Each visualization is categorized as either static, animation, or interactive:

- A static visualization is a set of fixed diagrams.
- An animation is, well, an animation.
- An interactive visualization can be either of the former types, but additionally allows the user to dynamically alter the parameters or form of the visualization.

  Nonexhaustive list of use cases:
- Understand what the activities of a process are ("the domain"). [static]
- Understand flow of activities. [static]
- Understand causal relationships among activities (perhaps from different processes). [static interactive]
- Analyze related perspectives (process models inherently multi-perspective: actors, resources, cost, . . .). [static]
- Focus into details (zoom, detailing). [interactive]
- Finding similarities and differences among processes. [static]
- Find out bottlenecks. [static]
- See progress of execution. [animation]
- Find violations of rules.

Typically, different stakeholders with different interests are involved.

4.6.4 2. Requirements

In this Section, we list the attributes of visualizations that appears to the Working Group as particular to process model visualizations.

- Visualize fragments/perspectives only (requires conceptualization of the workflow)
- Collapse / filter: abstraction, navigation.
- Uncollapse / enrich with information
- Select which part is shown (e.g. by zooming, or other means [most simple: enter start/end textually])
- Create different views
- Decrease computation needs based on user type
- Guide user in where to start reading, where to read next (e.g. by interactivity)
- Give control over impression of time distances and lengths
- Give control over how to perceive causal relationships
- Exploit spatial relationship to represent semantic relationships (especially because there are so many semantic relationship)
- Enable user controls

4.6.5 3. Examples

As a starting point for investigating the state of the art, this working group proposes collecting example visualizations from Seminar participant's recent research papers. Example visualizations should transcend basic BPMN diagrams. As part of the collection of examples, the Working Group proposes also forming a catalog of non examples: A list of important properties of process models that currently have no visual representation. Examples of such non-examples could be:

- Security properties
- Resource consumption

The working group considers this collection of non-examples the most promising avenue for finding fresh approaches to BPM visualizations.

4.6.6 Conclusions

The working group contends that visualization of process models, while not necessarily distinct from visualization in general, has the property that because of the richness of the underlying process models, no single visualization will fit all purposes. The object is therefore not to find the visualization of a process, but to to find a visualization suitable for the question at hand. The working group proposes:

1. The collection of a library of published example visualizations.
2. The construction of a catalog of non-examples in the above sense: Attributes of process models currently not being visualized.
3. The evaluation of comprehensibility of different visualization approaches (taking into account the background, needs and goals of different visualization consumers).

4.7 What can BPM bring to Internet of Things?

Agnes Koschmider (KIT – Karlsruher Institut für Technologie, DE), Avigdor Gal (Technion – Haifa, IL), Christian Janiesch (Universität Würzburg, DE), Udo Kannengiesser (Metasonic GmbH – Pfaffenhofen, DE), Massimo Mecella (Sapienza University of Rome, IT), Andreas Oberweis (KIT – Karlsruher Institut für Technologie, DE), Jianwen Su (University of California – Santa Barbara, US), Victoria Torres (Technical University of Valencia, ES), Barbara Weber (Universität Innsbruck, AT), and Liang Zhang (Fudan University – Shanghai, CN)

The Internet of Things (IoT) postulates not only the interaction of humans and software systems but also the interaction with "things", objects in the real world that can act as an autonomous agent. These agents execute small business processes that are triggered either through command for authority dependent things or events that signify a particular (complex) situation that the agent recognizes deems important enough to (re) act. The processes may be full-fledged business processes to deliver a good or simple notifications similar to a personal assistant reminding you of your own tasks which are due or close to failing. This is possible because sensor data is available in the large from stationary sensors in smart spaces as well as automated and manually transmitted data human and artificial agents which can be used in an automated fashion. Due to the increase in computing capacity and miniaturization, decisions cannot only be made from a central instance but also locally by the agent in a timely and useful fashion. In the future, even humans may be wired into this event cloud of information.

4.7.1 The promise BPM brings to IoT

BPM brings a process perspective to the way IoT is being used. Just like in traditional process management, the fundamental understanding of process data involves the notion of a process. IoT data may be created in a way that is completely independent of any notion of a process (e.g., by emitting GPS location every 20 seconds). However, once put in the right context (e.g., a process of a bus following a route in the city) such data all of a sudden become correlated (e.g., a bus cannot fly from one place to another). Therefore, BPM can provide an infrastructure for improving the use of IoT, control its quality, and improve its prediction ability.

4.7.2 The promise IoT brings to BPM

IoT is part of a recent trade knows as big data. As such, it provides continuous monitoring by using sensing devices, being wearable devices, positioning beacons, mobile phone sensing, etc. Such data is valuable for process management and mining is multiple ways. IoT data can assist in discovering situational processes or unknown processes. It can also serve in online process compliance. It assists in predicting operational measures such as length of stay, accumulations of queues, etc. Finally, it provides a way for AB testing of newly designed processes.

4.7.3 Process-related Internet of Things scenarios

Ideas about Internet of Things currently have many forms and range from intelligent coffee mugs with a craving for more black gold to sophisticated transport and logistics solutions linking suppliers and customers based on sensor data and their processes. In the following, we detail some examples which require business process modeling in the Internet of Things and present some common characteristics. One scenario would be personal assistant. The emergence of wearables and IoT devices allows to employ raw data coming from sensors to make decisions and then to apply these decisions through actuators, either virtual or physical, or proposed to the final user in the form of suggestions or alerts. Specific tasks can have many different goals, but they are mainly performed according to a set of models representing environmental dynamics and, noteworthy, user habits and desires. These models can be either handmade by experts or automatically extracted (through learning and mining techniques) from previously acquired sensor logs.

4.7.4 Open questions for research

On a more general level there are several broader questions that can be extracted from the scenarios introduced above. We have extracted those and present them in the following in no particular order.

- How to best organize events stemming from different event sources into an event log?
- How to connect low-level events with high-level activities?
- How much process structure is necessary when designing a flexible and mostly autonomous system?

- What part of the system has to be pre-modeled and what is rather discovered? How to combine model execute with discover predict?
- How to link a large amount of autonomous and asynchronous micro processes?
- How does the central role of communication in IoT fit with the control-flow centric view of most BPM approaches?
- How do we model the interaction of reasoning (analytics) and action (process) of things?
- How do we specify and, thus, model the autonomy of agents?
- How to link organizational (business) processes to personal/individual processes?
- How to support online conformance checking?

4.7.5 Conclusion

The Internet of Things provides a massive playground for industry as well as for personal use through the meaningful yet dynamic interaction of man, software, and machine. Due to the inherent size, complexity, and volatility, it may not be possible to model comprehensive business processes but only parts thereof. However, the coordinated effort of planning, executing and measuring systems based on a large amount of smaller and autonomous business processes is still necessary and new technical and technological solutions need to emerge. We have formulated a number of research questions on process modeling that we deem important to be tackled in the near future for an Internet of Things to make benefit of business processes.

4.8 What are purposes of process modeling?

Hajo A. Reijers (VU University Amsterdam, NL), Alexander Herwix (Universität Köln, DE), Thomas Hildebrandt (IT University of Copenhagen, DK), Julius Köpke (Alpen-Adria-Universität Klagenfurt, AT), Henrik Leopold (VU University Amsterdam, NL), and Han van der Aa (VU University of Amsterdam, NL)

The process model is the starting point and central artifact of many scientific contributions in the area of Business Process Management. This includes both empirical as well as technical contributions. Interestingly, most of these works abstract from the actual purpose of the process model. That is, they do not consider why and for whom the process models were created. In this working group, we thus started out with analyzing the range of purposes of process models. Based on this analysis, we identified areas where process models do not live up to the expectations we have about them. Our contribution is threefold. First, we listed the purposes of process models and the main organizational roles that are associated with these purposes. Second, we elaborated on three key areas in more detail and identified potential for new research directions. Third, we elaborated on the problem of integrating the different representations resulting from our proposals.

4.8.1 Purposes of Process Modeling

To determine directions for future improvements of business process modeling, we first identified a number of different purposes for which process models are used in organizations.

As a starting point for a discussion, we used the purposes identified by Reijers (2003). Without claiming completeness, this resulted in the following 10 purposes:

1. Training: Models used to introduce new employees to the business processes they will take part in.
2. Simulation & Analysis: (Executable) specifications of processes used to simulate and analyze the behavior business processes.
3. Costing & Budgeting: The usage of process models to measure and then price out the resources used for activities to generate goods and services for clients.
4. Enterprise Management: The usage of models to control and manage the enterprise or parts of it (spanning multiple business processes)
5. Documentation & Knowledge Management: Models used to capture and preserve tacit knowledge on an organization's business processes
6. Work Guidance: Models or artifacts used to support users (e.g. as a reference) while they are executing a process.
7. Enactment: Models that are used to automatically control & manage the execution of processes, e.g. as implemented in a workflow management system.
8. System Development: Models used to specify functional requirements to be used as input for systems development.
9. Organization Design: Models used as basis for business process improvement activities.
10. Compliance & Security: Models used to ensure compliance of business process-es to regulations or (ISO) standards.
11. Furthermore, we identified the organizational roles that make use these models for the aforementioned purposes in order to identify the stakeholders to which business process models are relevant.

4.8.2 Topic I: Work Guidance – Process Participant

Process models are being used by process participants to obtain information on how to carry out their work. For example, a process participant may understand a particular step in a process in more depth or needs to understand which steps are to be taken. The current nature of process models is such that they are not optimal to provide this in-sight.

4.8.3 Solution

There are different directions in which process models can be enhanced such that they can better guide workers. The following additional information can be added:

1. Status: the particular status of the work item that a process participant is interested in,
2. Explanation: the reasons why the work item that a process participant is interested in has reached the particular status it is in,
3. Goal: the particular goal for the work item or the overall process,
4. Route: the suggested route to further deal with the work item in question. These different aspects are partially dependent on each other, e.g. there is no route that can be recommended without knowledge about the particular status. Also, the incremental addition of these aspects lead to an increasingly sophisticated artifact. In the end, the most sophisticated manifestation of the artifact may be characterized as a context-sensitive recommendation system for process workers.

4.8.4 Topic II: Training – Process Participant

Problem: Consider the rehearsal of the emergency procedure for an accident at a train bridge (Debois et al., 2016). Today it is carried out using a storyboard for an emergency scenario, describing the anticipated events. Some of the events are initiated by the trainee (e.g., the accident, oil leaking from train, train bursting in fire, passengers falling in the water) other events / activities are omitted and replaced by questions to the participants (What do you do now?). The storyboard reflects the ”correct” procedures – and the trainee can help participants doing the right thing, but the rehearsal should also be able to continue if some users do not follow the right procedure. The problem is that current process models typically do not allow for describing re-hearsal storyboards / scenarios nor do tools support simulation of such scenarios as ”games”. This means that the rehearsal storyboards are created separately from the models of the emergency procedure (which is required to be described by law) and thus they need to be manually aligned. Also, there is very limited support for feedback during simulation.

4.8.5 Solution

A fresh approach to modeling for the purpose of process training would be to integrate the process model required for documenting the procedure with modeling of (good and bad) scenarios/story-boards for training games, i.e. to model the process needed to add gamification elements to the simulation. This could perhaps be done declaratively (Debois et al., 2016), e.g. by modeling the actions and goal of each participant in the game, actions that are preconditions, and responses (likely with deadlines) to actions / events happening during the training, penalties and rewards. The model should also allow for recording feedback to each action / event during the training. Another fresh approach could be to use process mining of both past rehearsals, real process executions and physical rehearsals to suggest scenarios relevant for training, and also to provide feedback on a finished training instance (e.g. how did this training in-stance related to previous ones?). It could even be possible that one could derive the game-model for training automatically based on historical data analyzed using e.g. process mining.

4.8.6 Topic III: Enterprise Management – Top Manager

Problem: Enterprise management is a challenging and knowledge-intensive activity that requires the synthesis of a variety of data sources (e.g., data warehouses or unstructured data about customer needs). Currently, process models used to facilitate this activity include, for example, process architectures and value chains. While these process models already provide a general overview of the enterprise and its processes, they do not facilitate the act of integrating this knowledge in an effective way. For example, current process architectures provide an overview of the existing processes and their relationships within a company. However, the process models themselves are modeled at a level of detail, which makes them difficult to use and interpret for top managers.

4.8.7 Solution

A fresh approach to modeling for the purpose of enterprise management could involve the development of new notations that facilitate the integration of process executions and management relevant objectives. This would allow for the presentation of management relevant perspectives on processes, for example, in the form of smart dashboards that provide managers with the possibility to drill-down into the most appropriate level of abstraction. Moreover, it would be interesting to pursue the development of advanced planning and simulation features that allow top managers to explore the consequences of adjusting management objectives and/or process models based on the historic data of process executions.

4.8.8 Alignment

Process models are used for different purposes and by user groups with different requirements. For example, the IT department needs a very detailed, enactable process model, while the management is typically much less interested in the sequencing of activities, but in the achievement of goals on various levels. The process models applied for training purposes typically only emphasize the typical cases ignoring much of the behavior modeled in the executable models. The compliance department is typically using rule-based modeling. These different requirements on models lead to the use of completely different tools such as Powerpoint for training, Excel for managers, rules for people being in charge of ensuring compliance, and executable BPM models for the IT department. Even on the modeling level, different modeling paradigms are being used by different user groups. For example, declarative models are often the first choice for modeling compliance whereas imperative models are the prominent candidate for modeling a business process. However, these different models actually describe the same real world process. This leads to alignment issues between the different artifacts. For example, a change in the management model or in the compliance rules needs to be reflected in the implementation and also in the training material. Process views have been proposed to support different user-groups with information on the right level of abstraction (Reichert et al., 2012). However, such views only allow abstracting or omitting elements of the process model and do not allow to add missing knowledge about the process (e.g., business goals or more details) and to translate a model to a different representation or modeling paradigm. Consequently, new methods are required:

- Semi-automatic creation of alignments between process models of different user groups and potentially different modeling paradigms by including domain knowledge. For example, mapping an executable process to the management's goal model.
- Deriving one model from another one including abstractions and refinement with the inclusion of domain knowledge. For example, generation of training material from an executable model.
- Automatic identification of parts of a group specific process model that are being affected by another group specific model, for example visualize on the level of a BPMN model for documentation purposes those parts being affected by a compliance rule.

References

1 Debois S., Hildebrandt, T., Sandberg, L. (2016): Experience Report: Constraint-based Modelling and Simulation of Railway Emergency Response Plans. Procedia CS, Elsevier.

2 Reichert M., Kolb, J., Bobrik, R., Bauer, T.(2012). Enabling personalized visualization of large business processes through parameterizable views.

3 Reijers H. A. (2003): Design and control of workflow processes: business process management for the service industry. Springer-Verlag.

Participants

- Banu Aysolmaz
VU University Amsterdam, NL
- Fernanda Baião
Federal University – Rio de Janeiro, BR
- Achim D. Brucker
University of Sheffield, GB
- Artur Caetano
INESC-ID – Lisboa, PT
- Soren Debois
IT Univ. of Copenhagen, DK
- Marlon Dumas
University of Tartu, EE
- Kathrin Figl
Wirtschaftsuniversität Wien, AT
- Avigdor Gal
Technion – Haifa, IL
- Jens Gulden
Universität Duisburg-Essen, DE
- Alexander Herwix
Universität Köln, DE
- Thomas Hildebrandt
IT Univ. of Copenhagen, DK
- Richard Hull
IBM TJ Watson Research Center – Yorktown Heights, US
- Christian Janiesch
Universität Würzburg, DE
- Udo Kannengiesser
Metasonic GmbH – Pfaffenhofen, DE
- Julius Köpke
Alpen-Adria-Universität Klagenfurt, AT
- Agnes Koschmider
KIT – Karlsruher Institut für Technologie, DE
- Ralf Laue
Westsächsische Hochschule Zwickau, DE
- Henrik Leopold
VU University Amsterdam, NL
- Tamara Mchedlidze
KIT – Karlsruher Institut für Technologie, DE
- Massimo Mecella
Sapienza University of Rome, IT
- Hamid Reza Motahari Nezhad
IBM Almaden Center – San Jose, US
- Andreas Oberweis
KIT – Karlsruher Institut für Technologie, DE
- Alexander Paar
TWT GmbH – Stuttgart, DE
- Jan Recker
Queensland University of Technology – Brisbane, AU
- Hajo A. Reijers
VU University Amsterdam, NL
- Flávia Maria Santoro
Federal University – Rio de Janeiro, BR
- Tijs Slaats
IT Univ. of Copenhagen, DK
- Chris Snijders
TU Eindhoven, NL
- Minseok Song
POSTECH – Pohang, KR
- Christian Stahl
TWT GmbH – Stuttgart, DE
- Jianwen Su
University of California – Santa Barbara, US
- Victoria Torres
Technical Univ. of Valencia, ES
- Han van der Aa
VU University of Amsterdam, NL
- Barbara Weber
Universität Innsbruck, AT
- Ingo Weber
Data61 / NICTA – Sydney, AU
- William Wong
Middlesex University, GB
- Liang Zhang
Fudan University – Shanghai, CN
- Michael zur Muehlen
Stevens Inst. of Technology, US

Report from Dagstuhl Seminar 16192

Supporting Organizational Efficiency and Agility: Models, Languages and Software Systems

Edited by

Tony Clark[1], Ulrich Frank[2], and Vinay Kulkarni[3]

1 **Sheffield Hallam University, GB,** t.clark@shu.ac.uk
2 **Universität Duisburg-Essen, DE,** ulrich.frank@uni-due.de
3 **Tata Consultancy Services – Pune, IN,** vinay.vkulkarni@tcs.com

Abstract

Organizations are complex systems that need to respond to a variety of changes while operating in a dynamic environment. They involve multiple stakeholders each having a domain-specific perspective that relies on concepts and languages relative to individual information-centric processes, which may lead to undesirable side-effects such as scattered and fractured knowledge about goals, strategies, operational processes etc. This inter-disciplinary seminar analyses how the design, operation and maintenance of organizations can be supported not only with managing their resources and processes efficiently, but also with coping with the digital transformation.

Seminar May 8–13, 2016 – http://www.dagstuhl.de/16192
1998 ACM Subject Classification H.1 Models and Principles, K.6 Management of Computing and Information Systems
Keywords and phrases Digital transformation, DSML, enterprise modelling, enterprise software systems, Organisational action systems
Digital Object Identifier 10.4230/DagRep.6.5.31
Edited in cooperation with Alexander Bock

1 Executive Summary

Tony Clark
Ulrich Frank
Vinay Kulkarni

Organizations are complex systems that need to respond to a variety of changes while operating in a dynamic environment. They involve multiple stakeholders each having a domain-specific perspective that relies on concepts and languages relative to individual information-centric processes, which may lead to undesirable side-effects such as scattered and fractured knowledge about goals, strategies, operational processes etc.

Organizations are increasingly penetrated by software: Processes and resources are digitized, decision making relies on data provided by software systems, and transactions with external stakeholders are performed by machines. On the one hand, the omnipresence of digital systems creates the opportunity for further automation: The more structures and processes that constitute organizations are represented in software, the greater the scope for computer-supported management. On the other hand, this omnipresence creates a substantial challenge: Many organizations lack the competence to cope with the further

Supporting Organizational Efficiency and Agility: Models, Languages and Software Systems, *Dagstuhl Reports*, Vol. 6, Issue 5, pp. 31–55
Editors: Tony Clark, Ulrich Frank, and Vinay Kulkarni
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

increasing complexity of IT infrastructures. This includes the problem of assessing the business impact of IT investment and of assigning IT costs appropriately.

In addition to these problems, organizations face a tremendous challenge: The digital transformation will eliminate many existing business models. It will enable new products and services and it may require organizations to substantially change the way they do business. Only, if organizations are prepared to cope with this challenge, will they be able to benefit from the digital transformation instead of suffering from it.

A key aspect of the digital transformation is automation. While the potential for further automation through software is especially obvious in industrial production, other areas such as administrative work, management, and professional training are more and more dominated by machines. Therefore, there is need for new ways of supporting enterprise agility through the use of integrated computer-based systems

This seminar analyses how organizations can be supported not only with managing their resources and processes efficiently, but also with coping with the digital transformation, a topic which is subject of various research fields including: Management Science (a rationalist perspective); Organisational Studies (including Psychology and Sociology); Information Systems; Software Engineering (including modelling and meta-modelling, big-data and self-adaptive systems); Requirements Engineering. Even though there is an obvious correspondence of foundational assumptions, there is hardly any exchange between these fields: an issue that the seminar aims to address.

Against this background, the seminar is based on the following assumptions:

- Organizations are prepared for change only if they account for the challenges related to adapting their software systems as well as the peculiarities of social change.
- Research on organizational change in general, on designing organizational software systems in particular, recommends not only ideas of how to make organizations more efficient, but of how to make them a better place to work and live in. Otherwise it will be hardly possible to develop advanced conceptions of future organizations that may serve as an orientation for change. Without respective considerations efficiency remains a fairly meaningless concept.
- Support for organizational efficiency and change recommends cross-disciplinary collaboration. While all three research streams outlined above focus on important aspects, none of them is sufficient on its own.
- Support for organisational decision making is currently very difficult due to the tacit nature of knowledge that must be reified and processed using advanced technologies.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

3 Overview of Talks

3.1 Social Contexts and Individual Factors in Enterprise Modelling

Balbir Barn (Middlesex University – London, GB)

Main reference B. S. Barn,R. Barn, F. Raimondi, "On the role of value sensitive concerns in software engineering practice," in Proc. of the 37th Int'l Conf. on Software Engineering (ICSE'15) –Volume 2, pp. 497–500. IEEE CS, 2015.
URL http://dx.doi.org/10.1109/ICSE.2015.182

Enterprise models are typically used as a means of establishing a shared understanding of an enterprise. More recently they are increasingly seen as having potential for controlling the enterprise especially through improved decision making capability.

Research has typically focussed on developing languages, tools and methodologies to help in the production of enterprise models. Social contexts or individual factors have largely been neglected as they have not been easy to represent in model form.

This presentation will focus on a reflection on the sociology of the organisation in order to outline routes to build theories of how social contexts and individual factors can be usefully analysed for the purposes of enterprise modelling. An example of modelling (moral) values will be used to illustrate the core characteristics of such an approach. This approach is derived from research completed on the development of mobile application for youth offending teams in the UK and reported in ICSE 2015.

3.2 Using Meta-Modelling to Integrate Languages for the Model Driven Organisation

Tony Clark (Sheffield Hallam University, GB)

An organisation consists of many different systems embedded in chains of usage that involve a mixture of human interaction and computer-based interaction. Interaction involves languages either to initiate computation or to communicate between systems, and between humans and systems. In virtually all current systems that underpin organisations the languages are fixed and are provided in some way by a third party – often system vendor. This leads to languages that are not particularly domain specific, do not take into account the level of IT sophistication of the user, and that are brittle because they cannot be changed without significant modification to the underlying systems.

Recent years has seen a trend in implementation technologies, particularly programming languages, that has provided meta-access to the underlying data representation and computational mechanisms. This allows run-time systems to analyse their own behaviour and make limited modifications to it. The motivation for such developments is to improve the resilience of large-scale systems in light of changes that are beyond the control of individual components. However, these improvements are limited and not particularly co-ordinated.

Organisational agility can be significantly improved by using meta-based approaches to the design and implementation of the organisation. This will lead to much better data and language integration, better resilience and an ability to tailor the languages used to interact with parts of an organisation. An organisation can become adaptive by reasoning about its own behaviour and each stakeholder can tailor the interaction language to suit their role and level of technology awareness. Having a meta-model of the underlying data structures supports an integration of otherwise disparate data types since the semantics of data is incorporated into the system itself.

3.3 An Industrial Perspective from SAP

Elmar Dorner (SAP SE – Karlsruhe, DE)

I'm for more than 15 years with SAP, working in different roles, but always as a Researcher. Throughout that time, the "Research" organization/department changed in several ways. The underlying working model, vision and mission got changed or updated based on internal and external influences. This included the scope of the work, as well as the interaction/engagement model. Also the operational model changed: More systems/tools for collaboration were introduced, processes for information handling were established, and a project management was put in place.

It's an open questions if these changes, especially of the working model of the organization, were established and implemented to advance operational efficiency and agility.

3.4 Organisational Perspectives: Elephants and Bazaars

Peter Fettke (DFKI – Saarbrücken, DE)

In my talk, I introduce two well-known metaphors for modelling and making information systems. According to the metaphor "The Blind Men and The Elephant", models of an organisation can be very different, although they represent the same object. More perspectives on an organisation give a richer picture of its characteristics. The usefulness of one model depends on the perspective of the modeller on the organisation.

The second metaphor "The Cathedral and the Bazaar" offers two ideal stereotypes for making information systems, which can be characterized by several concept pairs depicted in Table 1. Understanding an organisation more like a bazaar makes it more agile and adaptable to a changing environment.

The recent development towards Industry 4.0 exemplifies how an agile manufacturing organization might look like. In my talk, I present a Smart Factory demonstrator built with Lego® bricks (Figure 1). This prototype illustrates how principles of Industry 4.0 can be implemented.

I close my talk with potentials of Big Data Analytics, which provide a new, computational perspective on an organizational bazaar. Based on the Lego® Demonstrator, I point out prospects of large scale data sets for the inductive creating of models and decision making.

Table 1 The Cathedral and the Bazaar – Some Characteristics.

Cathedral	Bazaar
unity	plurality
standardisation	individualisation
one model	many models
monolithic	modular
mono-perspective	multi-perspective
top-down	bottom-up
deduction	induction
hierarchical structure	network structure, partially chaotic
fully integrated	only partially integrated
consistent and coherent	partially inconsistent and incoherent
error free	with errors
central organisation	self-organisation

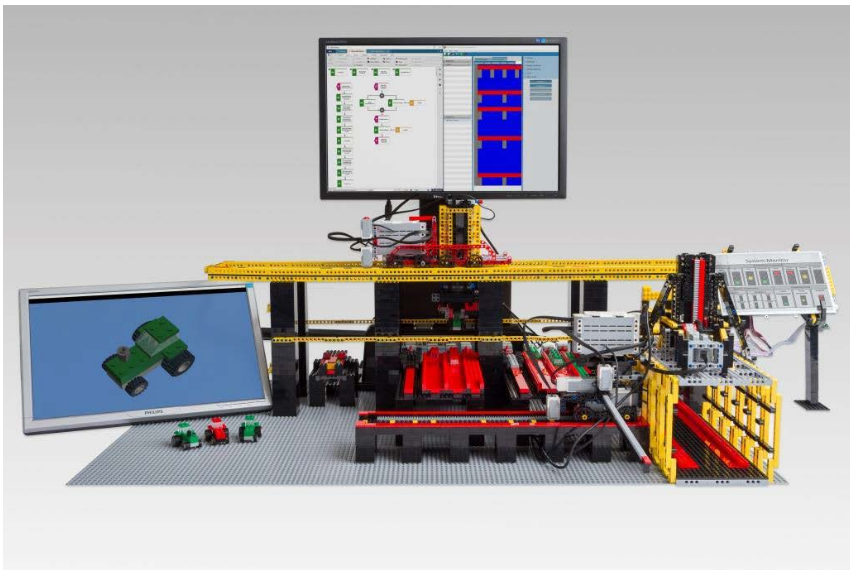

Figure 1 Industry 4.0 implemented with Lego® bricks.

3.5 Models as a Means for Supporting Digital Enterprises

Hans-Georg Fill (Universität Wien, AT)

In my talk I focus on the role of models for representing and analyzing enterprises in a digital economy. In my view, the main function of models is thereby the reduction of complexity through abstraction. In this way, models not only contribute to easing the communication between different human actors in an enterprise. They also establish the basis for the algorithmic processing of enterprise information. For illustrating this aspect I present recent research results on joining enterprise models for representing knowledge with enterprise models for representing and configuring data analyses. A particular focus of future research on these topics will be the gradual evolvement of enterprise modeling methods to continuously adapt to upcoming requirements. In this context, I will briefly outline the SeMFIS approach for the semantic annotation of conceptual models that has been developed throughout the past years. It permits the extension of the semantic representation and analysis scope of modeling methods without changing the original modeling language. Thereby, the consistency of existing modeling methods and models created with them is preserved, while at the same time new information requirements can be satisfied. Such approaches are considered to be essential for digital enterprises where new business and technological requirements constantly emerge and need to be reflected in the corresponding models.

3.6 The Role of Models and Language

Ulrich Frank (Universität Duisburg-Essen, DE)

More than ever, models are of pivotal relevance for enterprises to plan and run their operations and to collaborate in cross-organizational settings. This is for various reasons. First, the traditional "brick and mortar" type of companies is more and more replaced by companies that do not offer any physical presence to their customers. Therefore, it is essential to provide some kind of model of the company to give external stakeholders an idea of who they are dealing with. Second, an ever increasing amount of work in organizations is supported by software or even entirely automated. Due to the immaterial nature of software, it is mandatory that employees have a model of software that enables them to use it and, at best, to understand how it operates and how it may be adapted to new requirements. Third, the increasing demand for establishing cross-organizational work patterns, such as business processes and projects, required to integrate the relevant parts of information systems. Integration implies the existence and representation of commonalities, which in turn need to be represented in some kind of model that is accessible by all involved parties. Forth, the increasing complexity of products and services demands for specialization, that is, for different professional perspectives on an enterprise. At the same time, specialization creates the challenge of coordinating people with different viewpoints and heterogeneous objectives. Models may support particular perspectives with elaborate professional concepts. At the same time, multi-perspective enterprise models are suited to foster coordination through the integration of specific models of an enterprise. Last, but not least, most companies need to cope with the challenges of the digital transformation. In many cases, that will require them to change quickly without too much risk. For this purpose, they need to imagine future models of the enterprise to think and evaluate possible paths of change.

From an academic perspective, the need for models creates fascinating research opportunities. Conceptual models are linguistic artefacts. They are constructed with modelling languages, and their interpretation requires references to natural language. Domain-specific modelling languages (DSMLs) represent a promising approach to make the design and use of models more convenient and more consistent. However, the specification of DSMLs is confronted with considerable challenges. First, there is a principal conflict between economies of scale and productivity of reuse. Economies of scale recommends the development of DSMLs that can be used in a wide range of cases. However, more specific DSMLs promise a higher level of productivity in those cases where they fit. Second, it would empower users and contribute to the adaptability of software systems to supplement them with corresponding conceptual models. However, these models are useful in the long run only, if they evolve synchronously with the software systems they represent. Today's programming language architectures require separate representations of models and code, which results in the notorious synchronisation problem. Therefore, language architectures that enable a common representation of models and code would be extremely useful. Finally, the need for change creates a special challenge. The world we live in, the work practices and services, we are used to, get their meaning through the language we speak. It is our primary tool of thought. However, if companies are to change, the concepts we are used to, are likely to limit our imagination. Therefore, research needs to aim at ways to relax these limitations. Such an objective is extremely challenging and their is no deterministic procedure to accomplish it. However, it corresponds to the old idea of theory development, namely to "outlook" for new ideas that go beyond the obvious – through abstraction. A new paradigm of conceptual modelling that is based on multilevel language architectures seems to be well suited to address the challenges related to the construction and use of DSMLs. It also allows for a common representation of models and code, which enables the construction of a new kind of software systems that are integrated with conceptual models of their own and the surroundings they operate in.

3.7 Using Work Agreements as Operation-time System Requirements for Emergent Work Community Support Systems

Stijn Hoppenbrouwers (HAN University of Applied Science – Arnhem, NL)

We propose an approach for capturing evolving requirements for work support systems that are organically created by co-workers in self-organized, networked organizations. It is in the nature of such organisations that comprehensive design-time capturing of the volatile task-related functional requirements is not possible. Therefore, we advocate a combination of two types of requirements: i. stable requirement fragments elicited at design time, based on elementary collaboration and communication patterns likely to occur in an operational context, and ii. highly dynamic requirements in the form of explicit, easy-to-understand yet well-structured work agreements between organisational actors within organisations at operation-time. These agreements capture many aspects and concepts well known from requirements engineering, as well as business process analysis and design, but design-time modeling/specification of work-specific structures is now moved to operation time. Description of such structures by co-workers is supported by mechanisms part of the stable communication patterns under i.

3.8 Model-enabled Organizations

John Krogstie (NTNU – Trondheim, NO)

Whereas it has become usual in the community to talk about model-driven organizations (inspired by the use of the term data-driven), I have in the talk taken a more human-centric approach calling it ‘model-enabled organizations’. Models can to a much larger extent be used to enable organizational behaviour. Thus of the many topics suggested for the seminar, we have focused on the questions related to modeling, i.e.

- What kind of models of organizations do you find useful for what purpose?
- Who are/should be primary addressees of such models?
- What do you relate to the idea of models being repositories of organizational knowledge?
- Should modelling languages (DSMLs) rather aim at fitting a wide range of organisations or should they be tuned to the specific needs of one organization?

In particular it is looked upon how to achieve the long-term value of models, by understanding the different goals of modeling (and how to align short and long-term goals), and looking upon models, modeling languages, modeling methods and modeling tools and how stakeholder knowledge of all these areas have to be included and utilized. An example of large-scale modeling of the quality system in an oil company is presented, and even if it can be looked upon as a successful case of a model-enabled organization, also a number of challenges and possibilities for improvement are identified. To address these, we in particular look upon how interactive models can be used to support bottom-up (grass-root) modeling in combination with the traditional top-down modeling of the quality system/enterprise architecture.

3.9 MDO: Key requirements from industry perspective

Vinay Kulkarni (Tata Consultancy Services – Pune, IN)

Joint work of Vinay Kulkarni, Souvik Barat, Tony Clark and Balbir Barn
Main reference V. Kulkarni, S. Barat, T. Clark, B. Barn, “Toward overcoming accidental complexity in organisational decision-making,” in Proc. of the 2015 ACM/IEEE 18th Int’l Conf. on Model Driven Engineering Languages and Systems (MODELS’15), pp. 368-377, IEEE, 2015.
URL http://dx.doi.org/10.1109/MODELS.2015.7338268

Modern enterprises need to respond to a variety of change drivers in order to stay competitive in rapidly changing business context. The cost of erroneous decision is often prohibitively high and there may not be an opportunity for course correction later. Minimizing such undesired consequences calls for a-priori judicious evaluation of the available courses of action as regards their influence on the desired objective. The decision-makers are thus expected to understand, analyze and correlate existing information about various aspects of enterprise such as goals, operational processes, change drivers and their influence etc. Large size, complex structure, inherent socio-technical nature, and multiple stakeholders with possibly conflicting goals all contribute to the complexity of organisational decision-making. Increasingly felt demands of agility and certainty make this endeavor even more challenging. Current industry practice relies mostly on human experts with spreadsheet, word processors, and diagram editors being the most popular tools used for capturing the relevant information about enterprise. Informal nature of this information means power, rigour, and speed of sophisticated analysis cannot be brought to bear upon the decision-making problem. As a result, quality of the solution is largely dependent on knowledge and experience of human experts involved in the decision-making process. As modern enterprise is a large and complex system, the sheer volume of information makes manual analysis ineffective as well as inefficient. Moreover, as modern enterprise operates in increasingly dynamic environment, the information required for analysis needs to be kept up-to-date at increasingly rapid rate thus making manual analysis further untenable. Also, the required information is typically strewn across multiple documents, spreadsheets and pictures. Stitching together a coherent, consistent and integrated view from these pieces, and keeping it up-to-date over time is a serious challenge. All these factors contribute to the present lack of agility and uncertainty in organisational decision-making. Therefore, there is a need for an approach to organisational decision-making that enables decomposition of the overall goal into sub-goals, sub-sub-goals etc to the desired level of granularity. It should help identify a set of variables (i.e. Measures) that need to be observed in order to determine whether the finest-level goal is met. It should also help identify a set of variables (i.e. Levers) that influence a given Measure and be able to specify the influence in a formal manner. It should enable make explicit the dependencies between levers, between measures and between goals. This goal-measure-lever graph structure helps capture the understanding of problem domain in a manner that is amenable to automation. Decision-making then is a bottom-up walk of this graph structure provided it is possible: (i) to compute values for the measures based on the values of levers, (ii) to evaluate whether a goal is met based on the values of measures, and (iii) to honour lever-to-lever, measure-to-measure and goal-to-goal dependencies in the bottom-up walk. Therefore, organisational decision-making can be viewed as human-guided exploration of design space wherein past experience and expertise get captured in knowledge form.

3.10 There is Relevant Information in Models, but Models are not Really Relevant – Why?

Andreas Leue (Sphenon GmbH – Hamburg, DE)

Based on the experience of applying and observing model driven technologies to software production and related business management and organisation tasks for over 25 years, the speech tries to give an answer to the stated question.

First, some examples of working and useful ("good") models are presented, following by some counterexamples. Then, the majority of the slides discuss a variety of observed reasons: 1st, in the past insufficient tools ignited a downward spiral of model misuse and bad model reputation, 2nd, the enormous speed of technology and paradigm changes combined with far too many people working under high project pressure with too few skills to work with sophisticated technology prevents the maturing of everything which does not promise shortterm wins, 3rd, unintentional misuse of models like the application of token/controlflow based process schemas originating from a technical system domain to the business coordination domain, 4th, intentional misuse and rejection of model driven technology due to hidden stakeholder interests, like counteracting transparency, 5th, the utterly complex phenomenon of agility with a certain amount of positiv as well as a certain amount of negative impacts on model applicability, partially on justified and partially un unjustified ground, and 6th, the highly complex business landscape environment in which models have to survive, which needs to be approached from different system category perspectives like biological, linguistic, social, and economical in addition to the purely mechanical view related to IT.

The speech nevertheless finally presents an optimistic view of a future modelling biotope, in which small model parts coexist more loosely while at the same time more tightly bound, and with more loose semantics while at the same time more precise ones, and more interconnected while at the same time more isolated in a worldwide modelling artefact web.

3.11 Supporting Organizational Efficiency and Agility through Model-Based Collaboration Environments

Florian Matthes (TU München, DE)

Since 2002, our chair at the Technische Universität München investigates collaborative modeling activities in organizations of different types:

- Agile Enterprise Architecture Management in DAX and MDAX **enterprises**
- Collaborative modeling and incubation processes for digital **startups** and spin-offs (E-Commerce, Fin Tech, Legal Tech and E-Mobility)
- Eco-system modeling and management of **networked organizations**
- Modeling **legal aspects** in all of these organizations and networks.

For the purpose of this research, we have developed and applied a series of model-based collaboration environments since 1999 that have been expanded and simplified based on our improved understanding of how people actually use IT **tools to think, work and learn together** in organizations.

Model-based collaboration environments (MBCEs) provide the means for empowering information carriers and modelers to collaboratively and incrementally develop, maintain, and evolve models in a bottom-up fashion by using a light-weight **Hybrid Wiki approach** [3]. This approach enables the **emergent enrichment of unstructured content with structure**, achieving a MBCE that supports the co-evolution of organizational models (agents, roles, permissions, responsibilities, work plans, tasks, ...) and the underlying rich linked data models and computations in a coherent and consistent manner [4, 5].

Hybrid Wiki workspaces can be used for knowledge-intensive work at the **personal, group and enterprise level**. This allows for different adoption strategies in organizations of different complexity.

The Hybrid Wiki approach combines both modelling approaches, namely **top-down modelling** (models-first) and **bottom-up modeling** (data-first). Its goal is to empower organizational stakeholders, including modelers and non-modeling experts, to collaboratively gather and consolidate information in a flexible meta-model-based information system (**SocioCortex**, www.sociocortex.com), which acts as a MBCE for members of the organization [5].

The **backend of the platform** has a layered architecture based on a flexible temporal database for semi-structured linked content, with higher layers implementing dynamic content models, discretionary and role-based access control models, typed queries and functions, and artefact-centric process models.

The backend functionality is made accessible via open REST-based APIs and a typed query language to **generic (reflective) web clients** for collaborative content, task and model management and to **problem-specific front-end applications** (web clients, rich clients, mobile clients, embedded clients in other tools) and other information management and identity management systems (via a so-called Sync Pipes).

Research projects and university spin-offs have successfully used our MBCEs to support collaborative work in organizations of very different sizes, in different business domains, and also in organization networks.

For example SocioCortex can be utilized for modeling governance processes by the use of role models and associated concepts [1]. Furthermore, it provides an interactive web user-interface that assists users and modelers in writing queries, views, constraint and KPI definitions in a domain-specific expression language [6] based a polymorphic type system over rich linked data models. In the field of adaptive case management, it provides knowledge intensive process models for supporting the collaborative structuring of processes for knowledge works [2]. Last but not least, it supports the versioning of goal models and the calculation of KPIs for determining goal satisfactions of an enterprise model [4].

References

1 Birkmeier D.; Buckl, S.; Gehlert, A.; Matthes, F.; Neubert, C.; Overhage, S.; Roth, S.; Schweda, C. M.; Turowski, K.: The Role of Services in Governmental Enterprise Architectures – The Case of the German Federal Government. In: Anthopoulos, L.: An Investigative Assessment of the role of Enterprise Architecture in realizing E-Government Transformation. In Saha, P. (Ed.). Enterprise Architecture for Connected E-Government: Practices and Innovations. Hershey, PA: IGI Global. ISBN: 9781466618244.

2 Hauder, M.; Kazman, R.; Matthes, F.: Empowering End-Users to Collaboratively Structure Processes for Knowledge Work. 18th International Conference on Business Information Systems (BIS), Poznan, Poland, 2015.

3 Matthes, F.; Neubert, C. and Schneider, A. W.: Fostering Collaborative and Integrated Enterprise Architecture Modeling. In Journal of Enterprise Modelling and Information Systems Architectures, Vol. 8, No. 1, March 2013.

4 Monahov, I.: Integrated software support for quantitative models in the domain of Enterprise Architecture Management, PhD thesis, Technische Universität München, 2014.

5 Reschenhofer, T.; Bhat, M.; Hernandez-Mendez, A.; Matthes, F.: Lessons Learned in Aligning Data and Model Evolution in Collaborative Information Systems. In: Proceedings of the International Conference on Software Engineering (ICSE), Austin, Texas USA, 2016.

6 Reschenhofer, T.; Matthes, F.: Supporting End-Users in Defining Complex Queries on Evolving and Domain-Specific Data Models. In: Proceedings of the Symposium on Visual Languages and Human-Centric Computing. Cambridge, UK, 2016.

3.12 Enterprise Modelling and Semantic Technologies

Andreas L. Opdahl (University of Bergen, NO)

Big and open data will continue to grow in importance in the future. Semantic support will be needed to manage, manouever and make sense of the vast amounts of big open data available on the web. Standards and technologies for such semantic support have already been developed through efforts such as the semantic web, the web of data and linked open data. Although semantics is a key competence of the enterprise / IS / conceptual modelling community, there have been few attempts to bridge from enterprise modelling over into big, open and semantic data so far.

One potential bridge is that semantic data sets are annotated using standard terms defined in vocabularies, some of which (such as Prov and Org) already resemble enterprise modelling languages. Another potential bridge is to mine EM modelling languages and models from big semantic data sets and to use the resulting languages and models to navigate and make sense of the big data. A third bridge is to enrich enterprise models with open semantic data. A fourth bridge is to make enterprise models available as part of the semantic web / web of data / linked open data.

In the future, these and other bridges between enterprise / IS / conceptual modelling and semantic technologies can be leveraged to make enterprise and other models more autonomous and adaptive. Such “smart models” can live behind firewalls or in the cloud and will comprise semantically annotated models supported by clusters of reuseable software agents. The smart models will support future agile organisations through their ability to, e.g., dynamically update and enrich themselves, make links to and exchange information with other smart models, initiate and monitor organisational events and processes, reason about their purpose and possible uses, and offer themselves in suitable formats to new prospective human and machine users.

3.13 Beyond Agile Organizations

Dirk Riehle (Universität Erlangen-Nürnberg, DE)

Software development organizations are organizations that have led the move to “agile” first for software development itself, later across the whole enterprise, including all other business functions. However, agile software development has its own shortcomings and also has not succeeded in breaking down barriers to collaboration across organizational silos. Inner source, the use of open source collaboration practices within the organization has emerged as the next step in organizational change, in which employees are empowered to complement traditional management practices with bottom-up intelligence and cross-silo collaboration. In this talk, I report about 10 years of work on inner source software development and speculate how it might transcend software development and extend to the whole enterprise.

3.14 A Perspective on Organisational Efficiency and Agility

Kurt Sandkuhl (Universität Rostock, DE)

In general terms, enterprise modeling addresses the systematic analysis and modelling of processes, organization structures, products structures, IT-systems or any other perspective relevant for the modelling purpose. Enterprise models, as well as models of software systems and services, can form an important contribution to improving organizational efficiency and agility. Among the driving forces of competitiveness are – from our perspective – the capability of enterprise to quickly adapt to (context) changes which cannot be anticipated and an understanding of dependencies and relationships between organizational perspectives. Thus, we consider the following kinds of models in particular useful to support agility:

- Models capturing the (deployment) context of enterprise services and product, i.e., models of deployment context and models of actual business service or product should be separated
- Models of business models
- Enterprise Architecture models, in particular with focus on effects of digitization (including cyberphysical systems, Internet-of-Things, . . .)

Addressees of such models are not limited to IT experts and technical people, but a wide participation of enterprise stakeholders is required in development and use of models, e.g. for business and IT alignment.

As subject for future work, we propose to investigate "liquid models" which are flexible, dynamic and quickly adaptive regarding all possible aspects and perspectives of the model, e.g., model content, meta-model, model usage, model users, model boundaries, model creation. Such "liquidity" would support other model use scenarios required for improved support of agility, e.g., from models as design time artefact to runtime artefact or models which emerge in collaboration between what happens in reality and what is designed by stakeholders in the enterprise.

3.15 Towards Enterprise Architecture Modelling Practices

Gerhard Schwabe (Universität Zürich, CH)

My research background is CSCW/Collaborative technologies, Collaboration Engineering and IT Management. I have never published on Enterprise Architecture. However, I came into contact Enterprise Architecture Modelling practices in two projects: In a large project with a large Swiss Bank I studied the rennovation of the core banking plattform in 2004. And from 2011-2015 we studied the innovation practices at a large independent Swiss software vendor. In both organizations the use of formal Enterprise modelling was very limited; they rather relied on informal description applying Powerpoint. I therefore call for more studying and supporting actual architectural practices. Here lightweight tools relying on visual understanding and aesthetics become important.

3.16 A Few Thoughts on the Notion of 'Model Driven Organisation'

Stefan Strecker (FernUniversität in Hagen, DE)

In my talk, I question the notion of 'engineering an organization' by constrasting the formal structures, formal rules, and formal communication of/in an organization and by pointing to the importance of informal norms, communication, roles, groups, and leaders, among others. A 'functionalist' view on what drives organizations is contrasted by – one among many – complementary views I call the 'anthropological' view. The 'functionalist' view focusses e.g. on planning, decision-making, incentivizing, executing, controlling, monitoring, and auditing whereas the 'anthropological' view reminds us of important aspects of organizing, e.g., power games, symbolic action, hidden agendas, opportunistic action, untruthful revelation of intentions, implicit assumptions (see, e.g., Weick, Morgan, Mintzberg). The main theses of my talk is that 'organization members' strive for understanding and (ex post) rationalisation and attempt to 'make sense' of their perception of their 'organizational reality' which results in the implication that communication is both an essential foundation and a barrier at the same time. This leads me to the function of conceptual models which I believe we need to link to a moderate ambition, i.e., contribute to a bit more reason (+ rationality) through substantiated communication: Conceptual models enable communication about non-tangible aspects of organising, contribute to overcome communication barriers, foster a shared terminology and understanding of organising, and, hence, contribute to 'sensemaking' in organizations. We may then advance our ambition by reconstructing (i.e. reshaping) existing (technical) language taking information technology and the limits of language (design) into account. At the end of my talk, I mention questions I like to discuss in this Dagstuhl seminar.

3.17 Views of an Outsider

Reinhard Wilhelm (Universität des Saarlandes – Saarbrücken, DE)

My background is in verification, in particular of safety-critical embedded systems. Model-based design is the dominating development method in this domain. Systems are modeled on an appropriate level of abstraction, incorporating concepts such as feedback loops, filters, finite state machines. Code is automatically generated from the specification of a model.

It is tempting to repeat this success story in the area of business information systems. However, there are several problems. The appropriate level of abstraction has not been identified, yet. At least this did not show up in the presentations at the seminar, and in most of the systems, the human is in the loop, and the human is impossible to model. One could still attempt to model the other system components and offer the human actor a choice of alternatives together with an estimation of their costs, their benefits and other attributes.

3.18 Architectural Thinking

Robert Winter (Universität St. Gallen, CH)

In the context of the ongoing digitalization, models and modelling could gain significance if they enable

- gaining and maintaining deep insights about (internal & external) customers, e.g. their valuation of offerings,
- quick re-configuration of front stage IS and integration with efficiently run back stage IS,
- co-creation, i.e. the evolution of service providers from a vendor into an integral component of the customer's value creation, and
- flexible re-configuration of value-creation networks.

Is the current enterprise modelling discipline capable for such enablement? Only partially, because the community is often driven "inside-out", i.e. proposing models and modelling approaches based on perceived requirements without sufficiently understanding who is actually needing which kind of models for which kind of purposes. As a consequence, modelling and models have not reached their maximum possible impact in organizations, and are often only used by too few people for too few purposes. The enterprise modelling discipline needs to cope with very diverse concerns and stakeholders, leading to new requirements for models / modelling at different speeds and at different levels of precision. A complementary approach has been proposed under the label "Architectural Thinking". It intends to understand and reach "the other 90% of the organization" (who are not architects or IT people).

3.19 The Demand for a Customer Owned Ontology Model Layer in Continuous Enterprise Engineering at Cloud Computing

Peter Zencke (Universität Würzburg, DE)

Cloud Computing and Software as a Service is a disruptive technology with the potential to fully transform the ICT industry. While cloud computing with the operational efficiency of mega datacenter makes the usage of enterprise systems affordable for small and midsize enterprises and eliminates the cost of many traditional IT related services, the business adaptability of SaaS solutions is still at an infant stage. Most SaaS solutions treat all their tenants as equal allowing very limited customization.

In a business environment where enterprises have to strive for competitive differentiation, this limitation of todays cloud computing can hinder the broad adoption of enterprise SaaS. Enterprise Cloud Computing demands for a dedicated architecture to enable customer specific continuous engineering for change and adaptation.

In Enterprise System Engineering the separation of concern of different modeling views is state of the art. The three most important enterprise model views are datalogical, infological and ontological models. For cloud enterprise solutions in addition these three views have to become independent model layers with clear separated ownership.

At Cloud Computing the datalogical layer will be a shared layer for all enterprise tenants using the same Big Data infrastructure of non relational high performance data services. The infological layer at cloud computing will consume the dataservices of the datalogical model eliminating the redundancy of separate transactional and analytical data storages. Infological model views are computed at run time driven by an active infological repository with customer specific content.

The ontological model representing the enterprise organizational structure with its activities and processes by nature has to be in full ownership of each cloud tenant. Ontological models become a necessary layer in SaaS enterprise solutions allowing for competitive differentiation of enterprises in the cloud in a constantly changing business environment. Enterprise Engineering in the cloud must model the customer specific enterprise ontologies against generic services of Enterprise SaaS/SOA Platforms. By that Continuous Enterprise Engineering will become a crucial architectural pillar for future Enterprise Cloud Solutions.

4 Working groups

4.1 How to Deal with Organisational Evolution

Balbir Barn (Middlesex University – London, GB), Gregor Engels (Universität Paderborn, DE), Peter Fettke (DFKI – Saarbrücken, DE), Andreas Leue (Sphenon GmbH – Hamburg, DE), and Peter Zencke (Universität Würzburg, DE)

This group addressed the issue of how to acquire, evolve and assess a model for an enterprise under change.

For the purposes of the discussions the group defined an enterprise as having a structure including business units and business partners, and a communication structure. It has business goals and value creating processes. An organisational model can be acquired using inductive machine learning, crowd sourcing, modification of a reference model. There is a difference between a top-down approach vs. a bottom-up approach to model generation. Top-down can be developed from reference models and the known communication structure of the organisation. Bottom up can be mined from information sources such as email systems, ERP systems, workflow systems etc. Mining techniques can include machine learning, ontology learning.

When reasoning about evolution, it is important to understand the root causes and reasons for the change and to identify any patterns of change. It should be noted that not all information on work system of an enterprise is captured by the process model relationship between enterprise model and the used IT systems (EAM) is to be discussed in more detail

The group investigated 3 case studies: a startup; an established company without an existing enterprise model; an established company with an enterprise model entering a new market.

The group identified the following open questions:

- Current enterprise models do not capture reasons for change; future system should capture such information more competence in machine learning (feasibility), how to identify learning/training sets of data? Which techniques can be used / are appropriate? Nature-language processing / text mining, ontology engineering,
- There is work in the area of text mining in social media (facebook, twitter etc.). But this work is not connected to enterprise models
- How the content should be modelled that all parts of the organisation have benefit of the enterprise model?
- What is the correct level of abstraction of enterprise models to be useful? Particular views are typically stakeholder specific.
- Are there interesting meta-data for enterprise model which allows us to infer learning about the organisations, e.g. meta-data about the telecommunication tells us something about the individuals?
- Which language for which stakeholder for which element of an enterprise model is adequate?
- Which machine learning techniques can be useful for deriving enterprise model from existing data sources?
- Dealing with complexity and heterogeneity of data

Possible important research approaches include: Design science; Identification of important use cases (e.g. Mergers & Acquisition may be one interesting example); collective intelligence, crowd innovation.

The group identified the following open discussion questions:

- notion of theory, is always a theory needed
- principle problem of induction, how do you know that your inductive reasoning is valid/interesting
- interesting use case: compliance checking, you already have a model and check whether the data fits to model

4.2 Ambiguity Aware Models

Tony Clark (Sheffield Hallam University, GB), Jan L. G. Dietz (TU Delft, NL), Ulrich Frank (Universität Duisburg-Essen, DE), and Henderik Proper (Luxembourg Inst. of Science & Technology, LU)

This group discussed how to address the scale and complexity of organisational modelling through the use of ambiguity. Ambiguity means:

1. Underspecification (for example price:Float) leading to a broader than intended range of possible values.
2. Under-defined semantics (for example business process modelling languages). Leads to modelling languages being used for a wide-variety or purposes. There is a relationship between ambiguous models and formality.

It is the ability to be precise about the level of open-ness in a model in order for the development process to be able to start with something that is under-defined and end up with something that is good-enough and to support different approaches to modelling e.g., depth-first and breadth-first.

Ambiguity can be achieved by:

1. Leave parts of a model open in a controlled way. These can be made more strict when used.
2. Use specialisation as defined in types (equivalently set-theory) or in Object-Orientation.
3. Perhaps use meta-information associated with parts of model, use different displays for those parts of a model that are more or less specific. For example different shapes, softer-edges, different fonts.

4.3 Theories for Organisations

Jan L. G. Dietz (TU Delft, NL), Ulrich Frank (Universität Duisburg-Essen, DE), Henderik Proper (Luxembourg Inst. of Science & Technology, LU), and Stefan Strecker (FernUniversität in Hagen, DE)

There is no widely accepted theory for an Organisation that is amenable to computational processing in order to achieve organisational agility. The CIAO paradigm of J. L. G. Dietz is one possible contender. It should be possible to learn from existing theories based not he work of Max Weber, Gareth Morgan, Weick, Habermas, Luhmann, Enid Mumford, Peter Drucker, Peter Senge, Henry Mintzberg etc. where there are many different approaches including reductionist view, functional view, contingency approach, Stijn's Metaphor of 'software as frozen language' etc. We should investigate the work that addresses the intersection of organisational research, information systems research and software engineering.

Perception of organizational realities differs among observers (and we do not to subscribe to a social constructivists stance for that). The CIAO EE approach is based on the notion of 'I see what there is in organizational reality' and can validate what I see by talking to people. Modelling language enable simulations and provide a value to the org. research community Set of lenses – it is your choice to choose and apply a lens – might lead to inconsistencies. Can an organisation be engineered: this depends on notion of 'engineering' and 'organisation' for giving a positive answer to the question requires a specific understanding of both terms.

4.4 Business Modelling and Value Creation Modelling

Hans-Georg Fill (Universität Wien, AT), Stijn Hoppenbrouwers (HAN University of Applied Science – Arnhem, NL), Florian Matthes (TU München, DE), Henderik Proper (Luxembourg Inst. of Science & Technology, LU), Gerhard Schwabe (Universität Zürich, CH), Stefan Strecker (FernUniversität in Hagen, DE), and Robert Winter (Universität St. Gallen, CH)

This group addresses the issues relating to the creation of business models including value creation.

The context for the discussions included the shift from increasing efficiency of established organizations (with a working business model) to development of innovative (digital) business models where the value perception of customers is of top priority. Startups, spinoffs, new products often fail because of non-existing customer adoption and not of lack of technical feasibility. There is a shift from deployment to use and a shift from organization-centric structures to customer-centric value networks, for example: multiple independent mobility providers versus focus on mobility need of single a customer

Value modelling is an important consideration for achieving organisational agility because: there is an imperative to deliver solutions that are desired, feasible and viable. There is an increasing need to broaden the perspective of design to include customer value. It was noted that "Desirability" aspects needs better understanding and that the modelling community has a part to play in achieving this.

We want to enable a computer-supported processing of value models. The challenges for the modelling community include:

1. Difficulty of adequately capture the notion of value (what is value).
2. Difficulty to adequately represent / approximate and process value (creation, aggregation, comparison, propagation)
3. Difficulty to clarify the context in which "value" is used, for example: Potential value vs Realized value

A value is a multi-dimensional concept that includes more than money (financial value) for example, esthetic value, well-being, political value, ethical value, intellectual capital. Dimensions include:

- Aggregation level (individual, group, organization, networks of organizations, society)
- Type of valuation (monetary, esthetic, . . .)
- Object to be valuated (product, service, functionality, solution, process, brand, . . .)
- Ability to support automation (none, qualification, quantification, calculation / reasoning). Calculation and reasoning can be further distinguished into:
  - Formalized (algorithmic) reasoning
  - Argumentation and negotiation where human arguments can be based on intuition
- Static (pre-determined, in a design, build, run, observe cycle) or Dynamic (at runtime in a short-term control loop, e.g. auction process)

How to represent / approximate and process value What coverage of value was – and is now – possible?

- At design time: (that has always been possible):
  - Clarify – support stakeholder discourse regarding value proposition (pre system)
  - Facilitate – represent value in design (IS design tool passive),
  - Engage – semantic representation of value (IS design tool active)

- At use time: (that comes with digitalization)
  - Observe – capture value realization
  - Act – adapt IS to use and context

Difficulty of “aggregating” value (logic / arithmetic) because of the need of a context:

- See also examples below
- How to approximate the value perceived by a subject?
- How to model short-term and long-term value?

Example 1: Dynamic pricing of airline fares (B2C)

- Aggregation level: individual
- Type of valuation:
  - value indication: monetary
  - value approximation : binary (accept/decline)
- Object to be valuated: service (bundle?)
- Ability to support automation: Formalized (algorithmic) reasoning and could be improved
- Dynamic (at runtime in a short-term control loop)

Example 2: Design of a value network – Moovel (B2B)

- Aggregation level: value network
- Type of valuation:
  - Long-term strategic value
  - Degree of flexibility(buy,jointventure,long-termcontract,ad-hocinteraction)
  - Stability
  - Complementarity from the customer perspective
  - Monetary value
  - Brand value(Lada,Mercedes)
- Object to be valuated:
  - mobility-related business capabilities
    - share a car, call a taxi, rent a bike, find a parking spot
    - rate the quality of a service
    - route planning
    - dynamic pricing
    - payment
  - organizations(Daimler,BMW,DeutscheBahn,Sixt,Moovel,Google,. . .)
    - competitor analysis
    - value exchanges
  - Ability to support automation:
    - Clarify-supportstakeholderdiscourseregardingvalueproposition
    - Lots of room for improvement
  - Static (contract design between organizations, joint ventures, cooperatives, mergers and acquisitions)

Example 3: Health wristband deployment within an organization (informal negotiation inside the Org.)

- Aggregation level: organization
- Type of valuation: Privacy, well-being, occupational health, monetary value
- Object to be valuated: solution functionalities
- Ability to support automation:
  - Clarify – support stakeholder discourse regarding value proposition (tensions, negotiations, incentives)

  - Lots of room for improvement
- Static (Betriebsvereinbarung, works council agreement)

Example 4: Application landscape evolution within a given IT governance structure (conflicts raised by different planning horizons, scopes and goals inside the organization)

- Aggregation level: organization
- Type of valuation: completeness of requirements coverage, business complexity, architectural debt, time to market, compliance, security, sustainability. Value approximation: sets of incremental service improvements
- Object to be valuated: IT service portfolio
- Ability to support automation: depending on maturity qualification, quantification, for selected value dimensions even reasoning (simulation) Limits of reasoning in particular in the "desirability" space
- Static (pre-determined, in a design, build, run, observe cycle)

Reflection

Value seems to be a fundamental concept that needs to be better understood to approach a wide array of design problems. In our discussion we observed that the existing approaches to represent and process value take a discrete approach (instance, class, metaclass). It might be useful to investigate multi-dimensional continuous or subsymbolic representations and reasoning (neural networks). Analogy: Rule-based knowledge management systems vs. statistical machine learning approaches Analogy: Relational Database Systems vs. Information Retrieval Systems

4.5 Run-Time Models in Enterprises

Ulrich Frank (Universität Duisburg-Essen, DE), Jan L. G. Dietz (TU Delft, NL), Henderik Proper (Luxembourg Inst. of Science & Technology, LU), Dirk Riehle (Universität Erlangen-Nürnberg, DE), and Kurt Sandkuhl (Universität Rostock, DE)

This group discussed the issue of using models at run-time to support enterprise agility and to react to unanticipated events. The notion of *Liquid Models* was discussed where liquidity includes adaptability + scalability + usability

- Adaptability: concerns changes in the organization and in the context of the organization; model can be changed by stakeholders in the organization or by events in the organization or its context.
- Scalability: concerns the scale and complexity visible in the model; model content and boundaries affected.
- Usability: fully adaptable and scalable models still have to offer value and utility to stakeholders.

A vision for models at run-time: Stakeholder specific models explicitely available (for interaction to understand current systems and have a way to update (parts of) the system support) empowering users. Stakeholder specific visualizations(model-views) and should not be a need for intermediate format(as in traditional code-generation).

Models should be integrated with the enterprise software. There should be no discernible impedance mismatch between changes to the model of (an aspect of) the organisation and changing part of the enterprise software from the perspective of the stakeholder. In this context model should be interpreted as meaning: a stakeholder-specific perspective of the system. The environment should be included in the definition of 'enterprise software' and the model. The result of this integration is a seamless stakeholder perspective of the organisation and its context as a model.

4.6 Practice of and Collaboration for Creating and Using Models

Stijn Hoppenbrouwers (HAN University of Applied Science – Arnhem, NL), Hans-Georg Fill (Universität Wien, AT), Andreas Leue (Sphenon GmbH – Hamburg, DE), Florian Matthes (TU München, DE), Andreas L. Opdahl (University of Bergen, NO), Kurt Sandkuhl (Universität Rostock, DE), Gerhard Schwabe (Universität Zürich, CH), and Robert Winter (Universität St. Gallen, CH)

This group discussed the issue of using models in organisations – *why bother?*.

In order to further analyze the problem and develop the vision, several dimensions should be investigated, including Who is creating models? grasroot (i.e. everybody in an enterprise), traditional (modeling experts lead the process), machine-generated (e.g. from enterprise information sources), integration of existing models. Model representations and formalisation. Model scope and users: individuals, group, enterprise, ecosystem. Purpose: strategic purposes (e.g. enterprise architecture model), tactical, operational. Cross-level tasks to be supported: alignment, visualization, ambiguity detection, approximation (find similar models), annotating, linking, conflicts. Factors affecting success, failure, utility of modelling. Model lifecycles with different paces, scopes, etc. in lifecycle.

For support of grassroots modeling:

- Lightweight, no entry barrier (e.g. no fixed notation, not driven by specific concerns (what does "lightweight" really mean? What interactions / visualization/ concepts are established in what local practice?)
- Local practice of modeling welcome – linking to other models happens on demand, if required
- Support actual use situations
- Backbone powerful but invisible to the users: built-in collaboration features, social network integrated, marketplace of existing / established "modlets"

For traditional enterprise modeling: What kind of use scenarios are of major importance for supporting agility / efficiency in organisations?

In order to develop good quality models for enterprise engineering: Understand the "end" – actual need of users, designers, different roles. Required role structure "backbone" in the organisation. Model and content store providing "cross level features". More specific support of different integration tasks and requirements between different modeling approaches and task (differentiated integration supports). (How to) find commonalities between different local practices and identify candidates for reuse / propagating to other communities – requires de-contextualization of local practice. Another kind of education of people in modeling (start at school to do modelling?). Understand the required level of maturity for the purpose at hand – and what defines maturity. We need to move easily between the extremes of model representation. Increasing the value of the models by the human actors depending on the purpose/goals.

Participants

- Balbir Barn
Middlesex Univ. – London, GB
- Christoph Brandt
TU Chemnitz, DE
- Tony Clark
Sheffield Hallam University, GB
- Jan L. G. Dietz
TU Delft, NL
- Elmar Dorner
SAP SE – Karlsruhe, DE
- Gregor Engels
Universität Paderborn, DE
- Peter Fettke
DFKI – Saarbrücken, DE
- Hans-Georg Fill
Universität Wien, AT
- Ulrich Frank
Universität Duisburg-Essen, DE
- Stijn Hoppenbrouwers
HAN University of Applied Science – Arnhem, NL
- John Krogstie
NTNU – Trondheim, NO
- Vinay Kulkarni
Tata Consultancy Services – Pune, IN
- Andreas Leue
Sphenon GmbH – Hamburg, DE
- Florian Matthes
TU München, DE
- Andreas L. Opdahl
University of Bergen, NO
- Henderik Proper
Luxembourg Inst. of Science & Technology, LU
- Dirk Riehle
Univ. Erlangen-Nürnberg, DE
- Kurt Sandkuhl
Universität Rostock, DE
- Gerhard Schwabe
Universität Zürich, CH
- Stefan Strecker
FernUniversität in Hagen, DE
- Reinhard Wilhelm
Universität des Saarlandes – Saarbrücken, DE
- Robert Winter
Universität St. Gallen, CH
- Peter Zencke
Universität Würzburg, DE

Report from Dagstuhl Seminar 16201

Synergies among Testing, Verification, and Repair for Concurrent Programs

Edited by

Julian Dolby[1], Orna Grumberg[2], Peter Müller[3], and Omer Tripp[4]

1 **IBM TJ Watson Research Center – Yorktown Heights, US,** dolby@us.ibm.com
2 **Technion – Haifa, IL,** orna@cs.technion.ac.il
3 **ETH Zürich, CH,** peter.mueller@inf.ethz.ch
4 **Google Inc. – Mountain View, US,** trippo@google.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 16201 "Synergies among Testing, Verification, and Repair for Concurrent Programs". This seminar builds upon, and is inspired by, several past seminars on program testing, verification, repair and combinations thereof. These include Dagstuhl Seminar 13021 "Symbolic Methods in Testing"; Dagstuhl Seminar 13061 "Fault Prediction, Localization and Repair"; Dagstuhl Seminar 14171 "Evaluating Software Verification Systems: Benchmarks and Competitions"; Dagstuhl Seminar 14352 "Next Generation Static Software Analysis Tools"; Dagstuhl Seminar 14442 "Symbolic Execution and Constraint Solving"; and Dagstuhl Seminar 15191 "Compositional Verification Methods for Next-Generation Concurrency". These were held in January 2013; February 2013; April 2014; August 2014; October 2014; and May 2015, respectively. Two notable contributions of Dagstuhl Seminar 16201, which distinguish it from these past seminars, are (i) the focus on concurrent programming, which introduces significant challenges to testing, verification and repair tools, as well as (ii) the goal of identifying and exploiting synergies between the testing, verification and repair research communities in light of common needs and goals.

Seminar May 16–20, 2016 – http://www.dagstuhl.de/16201

1998 ACM Subject Classification B.8.1 Reliability, Testing, and Fault-Tolerance, D.1.3 [Concurrent Programming] Distributed Programming, Parallel Programming, D.2.4 [Software/-Program Verification] Assertion Checkers, Correctness Proofs, Model Checking, D.4.1 [Process Management] Concurrency, I.2.2 [Automatic Programming] Automatic Analysis of Algorithms, Program Modification, Program Transformation, Program Verification

Keywords and phrases (automatic) bug repair, concurrency bugs, concurrent programming, deductive verification, interactive verification, linearizability, synchronization, testing

Digital Object Identifier 10.4230/DagRep.6.5.56

Synergies among Testing, Verification, and Repair for Concurrent Programs, *Dagstuhl Reports*, Vol. 6, Issue 5, pp. 56–71
Editors: Julian Dolby, Orna Grumberg, Peter Müller, and Omer Tripp
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Omer Tripp
Julian Dolby
Orna Grumberg
Peter Müller

Context and Motivations

Major trends in computing infrastructure, such as multicore processors and data centers, increase the demand for concurrent software that utilizes the available resources. However, concurrent programs are notoriously difficult to develop. They are susceptible to a number of specific errors that do not occur in sequential code, such as data races, deadlock, atomicity violations, starvation, and violations of consistency models. These errors typically manifest themselves only in certain executions (for instance, under certain thread schedules), which makes them extremely difficult to detect, reproduce, localize, and repair. Established techniques for testing, verifying and repairing sequential programs are insufficient to handle concurrent software. In particular, they do not address the following challenges:

- *State space explosion:* The execution of a concurrent program depends not only on the inputs but also on the thread schedule and optimizations, such as memory reordering. This results in an state space that is orders of magnitude larger than for sequential programs. Bug-finding techniques, such as testing and bounded model checking, require effective ways of pruning the state space. Static verification techniques, such as deductive verification and abstract interpretation, require suitable abstractions that allow one to reason about all possible program behaviors. Finally, program repair requires techniques to predict the impact of a program change on the set of possible executions.
- *Modularity:* Modular techniques, such as unit testing or compositional verification, scale to large applications. However, for many properties of concurrent programs there are no modular techniques, or they require a large annotation overhead, for instance to denote the locations protected by a lock or to specify a locking order (or discipline) that ensures deadlock freedom. It is crucial to develop techniques that allow programs to be checked and repaired modularly, for instance to fix an atomicity violation by adding more thread synchronization, but without introducing a deadlock globally.
- *Specifications:* Testing, verification and repair may rely on specifications that express the intended program behavior, for instance in the form of test oracles or program invariants. In addition to functional properties, specifications for concurrent programs also have to express how threads cooperate, for instance via a global locking strategy. While various specification approaches exist for concurrent programs, there is no uniform formalism that handles the full range of concurrency idioms and that supports testing, verification and repair.
- *Error reporting:* Testing, verification and repair techniques need to disambiguate true problems from spurious defects, which is often difficult in concurrent programs. For instance, a data race is not necessarily a bug. If a race occurs within a lock-free data structure, then it may be admissible as part of some higher-level transactional behavior enforced by the data-structure operation. Moreover, it is important to present bugs in an understandable manner, for instance by providing reports with only a small number of threads and by determining whether a bug is inherently concurrent or may also arise in a sequential context.

- *Liveness:* Whereas for most sequential programs, termination is the only relevant liveness property, liveness (such as fairness or the absence of livelocks) is often more prevalent in concurrent programs. It is, therefore, important to develop techniques to check and enforce progress.

Program testing, verification, and repair each offer partial solutions to these challenges. This seminar was conceived with the goal of bringing together these three communities in order to develop a common understanding of the issues as well as to enable collaboration at the level of techniques and tools.

Main Themes

The first step toward exposing, and enabling, synergies between the three main threads of research on correctness and reliability of concurrent programs – verification, testing and repair – is to analyze the challenges and contributions pertaining to each of these areas in isolation. We survey work that has been done in each of these communities, based on the available literature and presentations given in the seminar, to summarize the current state of the three communities.

Verification

A main challenge in verification of concurrency properties is the prohibitive state space unfolded by thread interleavings. A hybrid solution to this problem is to specialize the static abstraction according to necessary proof conditions, arising during dynamic runs, such that the verification algorithm can scale with fine-grained abstractions (Naik, Yang). Another approach is to retain correlations among local thread states as well as the shared program state (Sagiv, Segalov). In this way, useful invariants can be proved and exploited by the verifier even if an unbounded number of threads is assumed. Refinement techniques are useful when little information is required about the environment to prove a property (Gupta). A useful idea in error reporting is to pinpoint concurrency-specific bugs (differentiating them from sequential bugs) by also running a sequential verifier and performing delta analysis (Joshi). Much like other techniques, verification greatly benefits from user specifications. For example, a parallelizing compiler is more likely to prove disjointness between loop iterations if relevant data structures (or operations) are specified as linearizable (Rinard, Diniz). This also provides a measure of modularity, enabling the separation between library linearizability checking and client verification. Modern program logics (O'Hearn, Parkinson, Gardner) provide a way of constructing correctness proofs for concurrent programs, though in general modular verification of concurrent software remains a hard problem.

Testing

Similarly to verification, testing techniques are also challenged by the state-space problem. Several ideas have been proposed in response to this problem. Open-world testing, whereby data structures or libraries referenced by an application are tested in isolation for concurrency bugs (e.g., atomicity violations), reduces the scope of testing considerably (Shacham). Interestingly, even open-world issues that cannot be recreated within the client application are often fixed by developers, which encourages further research into modular consistency properties (e.g., linearizability) (Shacham). Predictive analysis is a recent form of testing that holds the promise of high coverage at an affordable cost (Smaragdakis). Starting from a concrete trace, predictive analysis applies feasibility-preserving transformations (reordering trace events, typically through constraint solving) to detect concurrency bugs, such that soundness is guaranteed (Dolby, Huang). Another source of state-space reduction is to exploit high-level semantic guarantees, like atomicity, to abstract away intermediate trace transitions (Shacham, Tripp). This also relates to error reporting, where certain read/write conflicts give rise to spurious conflicts that can be eliminated with a higher-level view of conflict as lack of commutativity between atomic operations (Koskinen, Kulkarni). Contrary to memory-level conflict detection, commutativity-based testing requires a specification (Shacham, Tripp). Another form of specification refers to consistency relaxations, e.g. permitting certain types of read/write conflict (Thies) or specifying a computation as nondeterministic (Burnim, Tripp).

Repair

In program repair, error reporting (or localization) plays a key role, deciding the effective scope and nature of the fix. Pinpointing the exact conditions that give rise to a concurrency bug is thus critical, emphasizing the need for better testing and verification tools. Importantly, incorrect fixing may introduce concurrency bugs (e.g., a deadlock resulting from additional synchronization to fix an atomicity violation), which again highlights the need for better synergy between repair and testing/verification (Liu). Incorrect fixing also turns liveness into a concrete concern: Assuming the program previously terminated, does it also terminate after the fix? Existing solutions that ensure termination rely on iterative transformation methods as well as specialized models like Petri nets (Liu, Zhang). A common assumption in the repair community, to hold back the state-space challenge, is that concurrency bugs involve a small number of threads (typically 2) (Liblit, Liu). The hope is that better synergy with testing and verification can work toward relaxing this assumption. Semantic lifting of the concrete code, exploiting e.g. linearizability, has recently been demonstrated as a useful means to apply bottom-up/top-down fixing: First, the code is lifted into an abstract workflow, and then the workflow is concretized into a correct reimplementation (Liu, Tripp). This motivates further exploration of useful specification media for repair of concurrency defects.

Goals of the Seminar

The goal of the seminar was to promote cross fertilization among the verification, testing and repair communities, as they seem to be running into the same challenges, thereby solving increasingly similar problems. At the extreme, verification is about all possible program behaviors, testing is about running the program to see what it does, and repair is about generating new code. However, many techniques in all communities now blur the distinction. Use of dynamic information to guide abstractions in verification is one example; another is how predictive testing looks for bugs in possible executions close to a dynamic one, leading to a form of verification; finally, program repair increasingly uses solvers to synthesize new programs and test them, which overlaps with techniques from the other areas. We intended for the seminar to bring out further areas in which these fields are closely related, and inspire further techniques that fuse these areas, which was fulfilled by some of the discussions throughout the seminar.

Below are concrete examples of connections that we meant to expose, some of which were discussed throughout the seminar:

Benchmarks

Each area has a variety of benchmarks and competitions, and many of them ultimately focus on concurrency-specific challenges like interleavings. It seems likely that the different communities could benefit from sharing. For instance, predictive testing and verification could surely share many benchmarks, and a more standard set of benchmarks could make evaluations easier. At the same time, potential users could help ensure that any benchmarks actually measure what they care about.

Infrastructure

Much progress in both testing and verification has been made possible by progress in solver technology, and a variety of solvers are now common in both areas. There is room to share the infrastructure itself and the common remaining challenges.

Hybrid tools

The path-specific focus of testing and the global focus of verification can aid each other, e.g. current work such as CLAP using a control flow from a specific execution to make model checking more scalable.

Though the seminar touched on techniques and approaches that generalize beyond analysis and repair of concurrent software, we feel that the overall focus on challenges posed by concurrency was justified. With this focus, we were able to stir concrete discussion and tightly connected talks.

2 Table of Contents

3 Organization of the Seminar

The seminar lasted four days. We launched it with an introduction session featuring 2-minute lightning talks by each of the participants. These brief talks provided background on the person and relevant research experience and expertise. This enabled interaction and discussion from the very onset. The remainder of the seminar consisted of three types of talks: tutorials, demos and presentations.

Throughout the seminar, there were four tutorials. The first two were given on the first day of the seminar. The remaining two were given the next day. The goal of the tutorials was to expose the different communities to one another in a thorough and explicit manner (modulo time limitations). For this reason, the tutorials were given during the first part of the seminar. The tutorials were on the following topics:

- Model checking techniques for concurrent software
- Deductive verification of concurrent programs
- Repair of concurrency bugs
- Testing of, and test generation for, concurrent software

The tutorials provided general background, specific techniques as well as discussion of challenges and future research directions.

On the last day of the seminar, there was a session dedicated to demos, where three live demos were given. These showed use of both research and commercial tools. The topics were as follows:

- Directed model checking of JavaScript code (with asynchronous event handlers)
- Test generation for concurrent libraries
- Modular and interactive verification of concurrent programs

Finally, there were six technical sessions. Within each session, we intentionally combined talks on testing, verification and repair to expose synergies and encourage discussion. Topics that were covered include the following:

- Stateless model checking of event-driven applications
- Interpolation in model checking
- Predicate abstraction for bounded and unbounded concurrency
- Automated bug repair
- Making use of partial verification results
- Interactive verification of concurrent software using Dafny
- Reasoning about non-linearizable concurrent objects
- Algorithmic logic-based verification
- The Rely/Guarantee framework for verifying concurrent programs
- Proving termination via abstract interpretation
- Pervasive verification of multi-core systems
- The Java memory model
- Repairing linearizability violations in map-based operations
- Modular verification of message-passing programs
- Verification of event-driven JavaScript programs
- Concurrent specification of POSIX
- Using symbolic execution for space-time analysis of code

As the different types of talks and many topics that were covered illustrate, the challenges that were addressed in this seminar are complex and demand discussion and collaboration across the testing, verification and repair communities. There are natural links that, to date, have not been exploited sufficiently. Immediate examples include verification of programs after a concurrency bug has been automatically patched, guiding testing by the results of partial verification (which is what commercial tools typically support), and integrating verification with bounded model checking. These and various other points of synergy were explored and discussed throughout the seminar both during sessions and in ad-hoc meetings and forums (which is a great advantage of meeting at Dagstuhl).

As the reader can learn from the rest of this report, the seminar has achieved the goals of promoting discussion across the different communities, exposing common challenges and approaches to address these challenges, as well as shaping "hybrid" research directions that take their inspiration from the problems faced by both the testing and the verification and the repair communities. Participants provided highly positive feedback following the seminar, and expressed interest in follow-up events. The organizers are also very supportive of more seminars in the spirit of this seminar, which aim to explore and exploit links and synergies between the communities. There is room for more focused discussion on specific topics that were touched upon during the seminar. There should hopefully also be an opportunity in the future to reflect on, and shape, research directions that were borne out of this seminar or match its spirit.

4 Overview of Talks

4.1 Predicate abstraction for bounded and unbounded concurrency

Alastair Donaldson, Alexander Kaiser, Daniel Kroening, Michael Tautschnig, and Thomas Wahl

Main reference A. Kaiser, D. Kroening, T. Wahl, "Lost in Abstraction: Monotonicity in Multi-threaded Programs", Information and Computation, 2016; pre-print available from author's webpage.
URL http://dx.doi.org/10.1016/j.ic.2016.03.003
URL http://www.ccs.neu.edu/home/wahl/Publications/kkw16.pdf

Predicate abstraction, a technique for overapproximating system-level software by programs over Boolean-valued variables, has proved to be a success story in sequential program verification, especially for control-intensive programs. In recent years there has been some effort to extend this technique to multi-threaded programs. In this talk I discuss the requirements for shared-variable concurrent predicate abstraction, both for the case of a known number of threads, as well as for the unbounded case, where the number of threads is initially unknown or may dynamically change at runtime.

The result is that, while concurrent predicate abstraction reduces data complexity substantially (as in the sequential case), we have to pay for that by an increase in concurrency control complexity, even in the bounded-thread case.

This work as published primarily in the following two papers (journal versions of preceding conference papers):

- Alexander Kaiser and Daniel Kroening and Thomas Wahl. Lost in Abstraction: Monotonicity in Multi-threaded Programs. Information and Computation, 2016.
- Alastair Donaldson and Alexander Kaiser and Daniel Kroening and Michael Tautschnig and Thomas Wahl. Counterexample-guided abstraction refinement for symmetric concurrent programs. Formal Methods in System Design, 2012.

4.2 Starling: simpler concurrency proofs

Mike Dodds (University of York, GB), Matthew J. Parkinson, and Matthew Windsor

Modern program logics have made it feasible to reason about the most complex kinds of concurrent algorithm. However, many modern logics are enormously complex and difficult to understand, and most logics lack any kind of automated tool support.

We propose an antidote in Starling, a prototype tool for automated verification of concurrent algorithms. Starling takes a proof outline written in an intuitive predicate-based style, and converts it into proof obligations that can be discharged by Z3 or a Horn clause solver. Starling's underlying approach is based on the Views framework, which means it can be applied to many kinds of reasoning system. Starling can automatically verify several challenging examples including the Linux ticketed lock.

Starling is in active development on github: http://github.com/septract/starling-tool.

4.3 Automated Program Bug Repair

Orna Grumberg (Technion – Haifa, IL)

Joint work of Bat-chen Rothenberg, Orna Grumberg

This is a work in progress.

The work presents a novel approach for automatically repairing a program with respect to a given set of assertions. Programs are repaired using a predefined set of mutations. We impose no assumptions on the number of erroneous locations in the program. We refer to a bounded notion of correctness.

The repaired programs are returned one by one, in increasing number of mutations. Only minimal sets of mutations are applied. That is, if a program can be repaired by applying a set of mutations Mut, then no superset of Mut is later considered. This is based on the understanding that the programmer would like to get a repaired program which is as close to the original program as possible.

Our approach is based on formal methods. In particular, we exploit both SMT and SAT solvers, both incrementally. The SMT solver verifies whether a mutated program is indeed correct. The SAT solver restricts the search space of mutated programs to only those obtained by a minimal mutation set. Thus, an efficient search of all minimal repaired program is achieved.

We implemented a prototype of our algorithm and got very encouraging results.

4.4 Tutorial: Automated Repair of Concurrency Bugs

Ben Liblit (University of Wisconsin – Madison, US)

Joint work of Guoliang Jin, Shan Lu, Ben Liblit
Main reference G. Jin, W. Zhang, D. Deng, B. Liblit, S. Lu, "Automated Concurrency-Bug Fixing", in Proc. of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI'12), pp. 221–236, USENIX Association, 2012.
URL https://www.usenix.org/conference/osdi12/technical-sessions/presentation/jin

Concurrency bugs are widespread in multithreaded programs. Fixing them is time-consuming and error-prone. We present CFix, a system that automates the repair of concurrency bugs. CFix works with a wide variety of concurrency-bug detectors. For each failure-inducing interleaving reported by a bug detector, CFix first determines a combination of mutual-exclusion and order relationships that, once enforced, can prevent the buggy interleaving. CFix then uses static analysis and testing to determine where to insert what synchronization operations to force the desired mutual-exclusion and order relationships, with a best effort to avoid deadlocks and excessive performance losses. CFix also simplifies its own patches by merging fixes for related bugs.

Evaluation using four different types of bug detectors and thirteen real-world concurrency-bug cases shows that CFix can successfully patch these cases without causing deadlocks or excessive performance degradation. Patches automatically generated by CFix are of similar quality to those manually written by developers.

References

1 Dongdong Deng, Guoliang Jin, Marc de Kruijf, Ang Li, Ben Liblit, Shan Lu, Shanxiang Qi, Jinglei Ren, Karthikeyan Sankaralingam, Linhai Song, Yongwei Wu, Mingxing Zhang, Wei Zhang, and Weimin Zheng, "Fixing, Preventing, and Recovering From Concurrency Bugs." In Science China Information Sciences, volume 58, number 5, May 2015. Invited paper.

2 Guoliang Jin, Wei Zhang, Dongdong Deng, Ben Liblit, and Shan Lu, "Automated Concurrency-Bug Fixing." In Tenth USENIX Symposium on Operating Systems Design and Implementation (OSDI 2012), October 2012.

3 Guoliang Jin, Linhai Song, Wei Zhang, Shan Lu, and Ben Liblit, "Automated Atomicity-Violation Fixing." In Proceedings of the ACM SIGPLAN 2011 Conference on Programming Language Design and Implementation (PLDI 2011), June 2011.

4.5 Making the Java Memory Model Safe

Andreas Lochbihler (ETH Zürich, CH)

Main reference A. Lochbihler, "Making the Java Memory Model Safe", ACM Trans. Program. Lang. Syst., 35(4), Paper 12, 2014.
URL http://dx.doi.org/10.1145/2518191

Type safety and the Java security architecture distinguish the Java programming language from other mainstream programming languages like C and C++. Another important feature of Java is its built-in support for multithreading and the Java memory model. In this talk, I discuss how the current Java memory model affects type safety and Java's security guarantees.

The findings are based on a formal model of Java and the Java memory model. It includes dynamic memory allocation, thread spawns and joins, infinite executions, the wait-notify mechanism, and thread interruption, all of which interact in subtle ways with the memory model. The language is type safe and provides the data race freedom guarantee. The model and proofs have been checked mechanically in the proof assistant Isabelle/HOL.

References

1 Andreas Lochbihler. *Making the Java Memory Model Safe*. ACM Trans. Program. Lang. Syst. 35(4):12, 2014

4.6 Stateless Model Checking of Event-Driven Applications

Anders Møller (Aarhus University, DK)

Main reference C. S. Jensen, A. Møller, V. Raychev, D. Dimitrov, M. Vechev, "Stateless model checking of event-driven applications", in Proc. of the 2015 ACM SIGPLAN Int'l Conf. on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2015), pp. 57-73, ACM, 2015.
URL http://dx.doi.org/10.1145/2814270.2814282

Modern event-driven applications, such as, web pages and mobile apps, rely on asynchrony to ensure smooth end-user experience. Unfortunately, even though these applications are executed by a single event-loop thread, they can still exhibit nondeterministic behaviors depending on the execution order of interfering asynchronous events. As in classic shared-memory concurrency, this nondeterminism makes it challenging to discover errors that manifest only in specific schedules of events. In this work we propose the first stateless model checker for event-driven applications, called R4. Our algorithm systematically explores the nondeterminism in the application and concisely exposes its overall effect, which is useful for bug discovery. The algorithm builds on a combination of three key insights: (i) a dynamic partial order reduction (DPOR) technique for reducing the search space, tailored to the domain of event-driven applications, (ii) conflict-reversal bounding based on a hypothesis that most errors occur with a small number of event reorderings, and (iii) approximate replay of event sequences, which is critical for separating harmless from harmful nondeterminism. We instantiate R4 for the domain of client-side web applications and use it to analyze event interference in a number of real-world programs. The experimental results indicate that the precision and overall exploration capabilities of our system significantly exceed that of existing techniques.

4.7 Partial Verification Results

Peter Müller (ETH Zürich, CH)

Most techniques to detect program errors, such static program analysis, do not fully verify all possible executions of a program. They leave executions unverified when they do not check certain properties, fail to verify properties, or check properties under certain unsound assumptions such as the absence of arithmetic overflow.

In this talk, we present a technique to complement partial verification results by automatic test case generation. We annotate programs to reflect which executions have been verified, and under which assumptions. These annotations are then used to guide dynamic symbolic execution toward unverified program executions. We have implemented our technique for the .NET static analyzer Clousot and the dynamic symbolic execution tool Pex. It produces smaller test suites (by up to 19.2%), covers more unverified executions (by up to 7.1%), and reduces testing time (by up to 52.4%) compared to combining Clousot and Pex without our technique.

4.8 ISSTAC: Integrated Symbolic Execution for Space-Time Analysis of Code (Side-Channel Analysis)

Corina Pasareanu (NASA – Moffett Field, US)

Attacks relying on the inherent space-time complexity of algorithms implemented by software systems are gaining prominence. Software systems are vulnerable to such attacks if an adversary can inexpensively generate inputs that cause the system to consume an impractically large amount of time or space to process those inputs, thus denying service to benign users or otherwise disabling the system. The adversary can also use the same inputs to mount side-channel attacks that aim to infer some secret from the observed space-time system behavior.

Our project, ISSTAC: Integrated Symbolic Execution for Space-Time Analysis of Code, aims to develop automated analysis techniques and implement them in an industrial-strength tool that allows the efficient analysis of software (in the form of Java bytecode) with respect to space-time complexity vulnerabilities. The analysis is based on symbolic execution, a well-known analysis technique that systematically explores program execution paths and also generates inputs that trigger those paths. We are building a cloud-based symbolic execution engine for Java that includes new and improved algorithms for the symbolic space-time complexity and side-channel analysis of programs and a novel model counting constraint solver needed for quantifying the analysis results.

This is a 4-year collaborative project between Vanderbilt University, CMU, UC Santa Barbara and Queen Mary University, London. The project will build upon existing and mature symbolic execution tools (Symbolic PathFinder). I will give an overview of the project and highlight recent advancements on side-channel analysis. The ISSTAC website is: https://www.cmu.edu/silicon-valley/research/isstac/index.html.

4.9 Reasoning about non-linearizable concurrent objects

Ilya Sergey (University College London, GB), Aleksandar Nanevski, Anindya Banerjee, and German Andres Delbianco

Main reference I. Sergey, A. Nanevski, A. Banerjee, G. Andrés Delbianco, "Hoare-style Specifications as Correctness Conditions for Non-linearizable Concurrent Objects", arXiv:1509.06220v3 [cs.LO], 2015.
URL http://arxiv.org/abs/1509.06220v3

Designing scalable concurrent objects, which can be efficiently used on multicore processors, often requires one to abandon standard specification techniques, such as linearizability, in favor of more relaxed consistency requirements. However, the variety of alternative correctness conditions makes it difficult to choose which one to employ in a particular case, and to compose them when using objects whose behaviors are specified via different criteria. The lack of syntactic verification methods for most of these criteria poses challenges in their systematic adoption and application.

In this line of work, we argue for using Hoare-style program logics as an alternative and uniform approach for specification and compositional formal verification of safety properties for concurrent objects and their client programs. Through a series of case studies, we demonstrate how an existing program logic for concurrency can be employed off-the-shelf to capture important state and history invariants, allowing one to explicitly quantify over interference of environment threads and provide intuitive and expressive Hoare-style specifications for several non-linearizable concurrent objects that were previously specified only via dedicated correctness criteria. We illustrate the adequacy of our specifications by verifying a number of concurrent client scenarios, that make use of the previously specified concurrent objects, capturing the essence of such correctness conditions as concurrency-aware linearizability, quiescent, and quantitative quiescent consistency.

4.10 Actor Services: Modular Verification of Message Passing Programs

Alexander J. Summers (ETH Zürich, CH)

Joint work of Müller, Peter
Main reference A. J. Summers, P. Müller, "Actor Services: Modular Verification of Message Passing Programs", in Proc. of the 25th European Symposium on Programming (ESOP'16), LNCS, Vol. 9632, pp. 699–726, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-662-49498-1_27

We present actor services [1]: a novel program logic for defining and verifying response and functional properties of programs which communicate via asynchronous messaging. Actor services can specify how parts of a program respond to messages, both in terms of guaranteed future messages, and relations between the program states in which messages are received and responses sent. These specifications can be composed, so that end-to-end behaviours of parts of a system can be summarised and reasoned about modularly. We provide inference rules for guaranteeing these properties about future execution states without introducing explicit traces or temporal logics.

Actor services are ultimately derived from local actor services, which express behaviours of single message handlers. We provide a proof system for verifying local services against an implementation, using a novel notion of obligations to encode the appropriate liveness requirements. Our proof technique ensures that, under weak assumptions about the underlying system (messages may be reordered, but are never lost), as well as termination of individual message handlers, actor services will guarantee suitable liveness properties about a program, which can be augmented by rich functional properties. Our approach supports reasoning about both state kept local to an actor (as in a pure actor model), and shared state passed between actors, using a flexible combination of permissions, immutability and two-state invariants.

References

1 Alexander J. Summers and Peter Müller. *Actor Services: Modular Verification of Message Passing Programs*. European Symposium on Programming (ESOP) 2016, Springer-Verlag, LNCS.

4.11 Tutorial: Deductive Verification Tools

Alexander J. Summers (ETH Zürich, CH)

Joint work of Müller, Peter; Schwerhoff, Malte
Main reference P. Müller, M. Schwerhoff, A. J. Summers, "Viper: A Verification Infrastructure for Permission-Based Reasoning", in Proc. of the 17th Int'l Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI'16), LNCS, Vol. 9583, pp. 41–62, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-662-49122-5_2

This tutorial summarises the state of the art in automated deductive verification tools – those which take a program along with specifications/annotations and attempt to prove that the program satisfied its specification, reporting potential violations. An overview of a number of these tools is given, including a summary of the different applications and features of these tools, and the Chalice verifier for race-free concurrent programs (http://chalice.codeplex.com) is shown for a concrete demonstration.

The main two technical approaches for building such verification tools are symbolic execution (defining a custom verification engine) and verification condition generation (embedding verification problems in a lower-level verification language). Both techniques are briefly explained, and the idea of intermediate verification languages is motivated. The Viper Project (http://viper.ethz.ch) [1] is presented, which is a new such intermediate language and suite of generic verification tools, designed to easily support encodings of modern program logics and other reasoning methodologies. The tutorial concludes by giving demonstrations of how both traditional permission-based reasoning and recent techniques such as those addresses weak memory reasoning can be simply implemented via encodings into Viper, exploiting the reusable verifiers provided.

References

1 Peter Müller and Malte Schwerhoff and Alexander J. Summers. *Viper: A Verification Infrastructure for Permission-Based Reasoning*. Verification, Model Checking, and Abstract Interpretation (VMCAI) 2016, Springer-Verlag, LNCS,

4.12 Fixing Linearizability Violations in Map-based Concurrent Operations Automatically

Omer Tripp (IBM TJ Watson Research Center – Yorktown Heights, US)

Joint work of Peng Liu, Omer Tripp, Xiangyu Zhang
Main reference P. Liu, O. Tripp, X. Zhang, "Flint: fixing linearizability violations", in Proc. of the 2014 ACM Int'l Conf. on Object Oriented Programming Systems Languages & Applications (OOPSLA'14), pp. 543–560, ACM, 2014.
URL http://dx.doi.org/10.1145/2660193.2660217

Writing concurrent software while achieving both correctness and efficiency is a grand challenge. To facilitate this task, concurrent data structures have been introduced into the standard library of popular languages like Java and C#. Unfortunately, while the operations exposed by concurrent data structures are atomic (or linearizable), compositions of these operations are not necessarily atomic. Recent studies have found many erroneous implementations of composed concurrent operations.

In this talk, I address the problem of fixing nonlinearizable composed operations such that they behave atomically. Specifically, I will present an automated fixing algorithm for composed Map operations and its implementation as the Flint tool. Flint accepts as input a composed operation suffering from atomicity violations. Its output, if fixing succeeds, is a composed operation that behaves equivalently to the original operation in sequential runs and is guaranteed to be atomic.

Flint takes a first step towards fixing incorrect concurrent compositions fully automatically, encouraging more research effort in this direction. Evaluation of Flint on 48 incorrect compositions from 27 popular applications, including Tomcat and MyFaces, has yielded highly encouraging: Flint is able to correct 96% of the methods, and the fixed version is often the same as the fix by an expert programmer and as efficient as the original code.

4.13 Bringing Abstract Interpretation to Termination and Beyond

Caterina Urban (ETH Zürich, CH)

Joint work of Miné, Antoine
Main reference C. Urban, A. Miné, "Proving Guarantee and Recurrence Temporal Properties by Abstract Interpretation", in Proc. of the 16th Int'l Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI), LNCS, Vol. 8931, pp. 190–208, Springer, 2015.
URL http://dx.doi.org/10.1007/978-3-662-46081-8_11

Program termination is the most prominent liveness property. We design new program approximations, in order to automatically infer sufficient preconditions for program termination and synthesize piecewise-defined ranking functions, which provide upper bounds on the waiting time before termination. We also contributes an abstract interpretation framework for proving liveness properties, which comes as a generalization of the framework proposed for termination. In particular, the framework is dedicated to liveness properties expressed in temporal logic, which are used to ensure that some desirable event happens once or infinitely many times during program execution. The results presented in this talk have been implemented into a prototype analyzer. Experimental results show that it performs well on a wide variety of benchmarks and it is competitive with the state of the art.

Participants

- Mike Dodds
University of York, GB
- Julian Dolby
IBM TJ Watson Research Center – Yorktown Heights, US
- Derek Dreyer
MPI-SWS – Saarbrücken, DE
- Philippa Gardner
Imperial College London, GB
- Orna Grumberg
Technion – Haifa, IL
- Arie Gurfinkel
Carnegie Mellon University – Pittsburgh, US
- Cliff B. Jones
University of Newcastle, GB
- K. Rustan M. Leino
Microsoft Corporation – Redmond, US
- Ben Liblit
University of Wisconsin – Madison, US
- Andreas Lochbihler
ETH Zürich, CH
- Peter Müller
ETH Zürich, CH
- Anders Møller
Aarhus University, DK
- Wytse Oortwijn
University of Twente, NL
- Corina Pasareanu
NASA – Moffett Field, US
- Wolfgang J. Paul
Universität des Saarlandes, DE
- Arnd Poetzsch-Heffter
TU Kaiserslautern, DE
- Murali Krishna Ramanathan
Indian Institute of Science – Bangalore, IN
- Malavika Samak
Indian Institute of Science – Bangalore, IN
- Ilya Sergey
University College London, GB
- Natasha Sharygina
University of Lugano, CH
- Sharon Shoham Buchbinder
Tel Aviv University, IL
- Alexander J. Summers
ETH Zürich, CH
- Michael Tautschnig
Queen Mary University of London, GB
- Omer Tripp
IBM TJ Watson Research Center – Yorktown Heights, US
- Caterina Urban
ETH Zürich, CH
- Yakir Vizel
Princeton University, US
- Thomas Wahl
Northeastern University – Boston, US

Report from Dagstuhl Seminar 16202

Hardware Security

Edited by

Osnat Keren[1], Ilia Polian[2], and Mark M. Tehranipoor[3]

1 **Bar-Ilan University, IL,** osnat.keren@biu.ac.il
2 **Universität Passau, DE,** ilia.polian@uni-passau.de
3 **University of Florida – Gainesville, US,** tehranipoor@ece.ufl.edu

Abstract

This report documents the program and outcomes of Dagstuhl Seminar 16202 “Hardware Security", which was held in Schloss Dagstuhl – Leibniz Center for Informatics from May 16–20, 2016. This seminar aims to bring together a group of researchers, who are actively involved in the design and the security assessment of hardware primitives. The seminar was organized around presentations given by several participants on their current research, and ongoing work. In addition to these presentations, the program also included three discussion sessions, and two special sessions on curriculum development and funding programs. The seminar was indeed successful in familiarizing the researchers with recent developments in hardware security field of study, providing better understanding of still unsolved problems, and pointing out future research directions.

The paper is further organized as follows. Section 1 summarizes the most important goals of the seminar. Section 3 is devoted to the abstracts of the presentations given in the seminar, whereas in Section 4 the abstracts of the discussion sessions are provided.

Seminar May 16–20, 2016 – http://www.dagstuhl.de/16202
1998 ACM Subject Classification B.7.m [Integrated Circuits] Miscellaneous, C.3 [Special-Purpose and Application-Based Systems] Smartcards and Real-time and Embedded Systems, E.3 [Data Encryption] Code Breaking and Standards, E.4 [Coding and Information Theory] Error Control Codes, I.2.6 [Learning] Concept Learning, K.6.5 [Security and Protection] Authentication and Unauthorized Access
Keywords and phrases Hardware security; Passive and active side-channel analysis; Machine learning; Cryptographic blocks; True random number generators; Physically unclonable functions; Hardware Trojan
Digital Object Identifier 10.4230/DagRep.6.5.72
Edited in cooperation with Fatemeh Ganji

1 Executive Summary

Osnat Keren
Ilia Polian
Mark M. Tehranipoor

The convergence of IT systems, data networks (including but not limited to the Internet) and ubiquitous embedded devices within the cyberphysical system paradigm has led to the emergence of new security threats associated with the system hardware. Manipulating the hardware components that implement security functions can compromise system integrity,

Hardware Security, *Dagstuhl Reports*, Vol. 6, Issue 5, pp. 72–93
Editors: Osnat Keren, Ilia Polian, and Mark M. Tehranipoor
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

provide unauthorized access to protected data, and endanger intellectual property. Additionally, secure hardware is required to protect software in a proper manner tampering. Addressing these vulnerabilities is essential in order to prevent the hardware from becoming the Achilles heel of today's systems. Current technology trends point towards massive utilization of hardware circuits in larger cyberphysical systems that are interacting with the physical environment via sensors and actuators. At the same time cyberphysical systems are more and more integrated via open networks, most notably the Internet. Moreover, they interact with each other, forming systems of systems that exhibit highly complex, emergent behavior and constantly change their boundaries, with new sub-systems continuously entering and leaving. As a consequence, hardware-related threats must be addressed by appropriate countermeasures at realistic costs.

The seminar will focus on security threats where hardware components play the main role, and on countermeasures to address these threats. The emphasis is on generic algorithmic advances on the boundary between computer science and other disciplines. While Hardware Security is a very diverse scientific field, the seminar will specifically focus on its three main areas: passive and active side-channel analysis of security-relevant hardware components (cryptographic blocks, true random number generators) which goes beyond classical cryptanalysis; physical unclonable functions (PUFs) and authentication solutions on their basis; and new threats through hardware Trojans and counterfeit ICs as well as techniques for their detection and neutralization.

2 Table of Contents

Executive Summary

Overview of Talks

Discussion Sessions

3 Overview of Talks

3.1 Dismantling real-world ECC with Horizontal and Vertical Template Attacks

Lejla Batina (Radboud University Nijmegen, NL)

Joint work of Margaux Dugardin, Louiza Papachristodoulou, Zakaria Najm, Lejla Batina, Jean-Luc Danger, Sylvain Guilley, Jean-Christophe Courrege, and Carine Therond
Main reference M. Dugardin, L. Papachristodoulou, Z. Najm, L. Batina, J.-L. Danger, S. Guilley, J.-C. Courrege, C. Therond, "Dismantling real-world ECC with Horizontal and Vertical Template Attacks", in Proc. of the 7th Int'l Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE'16), LNCS, Vol. 9689, pp. 88–108, Springer, 2016; pre-print available from IACR.
URL http://dx.doi.org/10.1007/978-3-319-43283-0_6
URL http://eprint.iacr.org/2015/1001

Recent side-channel attacks on elliptic curve algorithms have shown that the security of these cryptosystems is a matter of serious concern. The development of techniques in the area of Template Attacks makes it feasible to extract a 256-bit secret key with only 257 traces. This paper enhances the applicability of this attack by exploiting both the horizontal leakage of the carry propagation during the finite field multiplication, and the vertical leakage of the input data. As a further contribution, our method provides detection and auto-correction of possible errors that may occur during the key recovery. These enhancements come at the cost of extra traces, while still providing a practical attack. Finally, we show that the elliptic curve technology developed in PolarSSL running on a ARM STM32F4 platform is completely vulnerable, when used without any modifications or countermeasures.

3.2 Machine Learning Attacks on Delay Based PUFs and Protocols

Georg T. Becker (Ruhr-Universität Bochum, DE)

Main reference G. T. Becker, "The gap between promise and reality: on the insecurity of XOR arbiter PUFs", in Proc. of the 17th Int'l. Workshop on Cryptographic Hardware and Embedded Systems (CHES'15) , LNCS, Vol. 9293, pp. 535–555, Springer, 2015.
URL http://dx.doi.org/10.1007/978-3-662-48324-4_27

In this talk I give an overview of the current state-of-the-art in machine learning attacks on XOR Arbiter PUFs and argue why we are far away from building secure Strong PUFs. In particular, I present the Reliability-based Machine Learning attack I introduced last year at CHES.

References

1 G. T. Becker. On the Pitfalls of Using Arbiter-PUFs as Building Blocks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34(8), 2015.

3.3 Hardware Security in Advanced CMOS Technologies

Wayne P. Burleson (University of Massachusetts – Amherst, US)

Joint work of See References

CMOS technology trends pose challenges and opportunities for Hardware Security design, applications and threats. VLSI research has evolved over the last decades, solving design problems related to area, timing, power, testing, and others, but most recently, security and privacy have moved to the forefront. Driving applications have also advanced to smaller and more autonomous systems, culminating in the Internet of Things which requires rethinking of security and privacy requirements and solutions at both the thing and cloud level. Implantable medical devices in particular present unique design constraints and threat models. Variations in advanced CMOS technology and operating environment present challenges and opportunities related to security, illustrated in three recent research projects : 1) Hardware Trojans present a real vulnerability during untrusted design/manufacturing especially in random number generation where functional validation is difficult. 2) Variations in the data retention time of memory cells can be used as a static entropy source, also known as physical unclonable functions (PUF), however reliably extracting this entropy across temperature variation requires novel processing based on ranking and hashing functions. 3) Environmental variations that impact PUFs can be used for virtual proofs of physical reality, a powerful new concept and capability in hardware security. Finally, on-chip sensor networks to monitor behavior and variations can be used to detect vulnerabilities, however can introduce their own vulnerabilities if not secured across untrusted processes in multi-core processors. Many open problems remain in all of these areas, from specific application and implementation issues, to novel attacks and countermeasures.

References

1 S. Ghoreishizadeh, T. Yalçin, A. Pullini, G. De Micheli, W. Burleson, S. Carrara, A Lightweight Cryptographic System for Implantable Biosensors. In IEEE Biomedical Circuits and Systems Conf. (BioCAS), 2014.

2 G. T. Becker, F. Regazzoni, C. Paar and W. Burleson, Stealthy Dopant-Level Hardware Trojans. In Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2013.

3 X. Xu, A. Rahmati, D. Holcomb, K. Fu, W. Burleson, Reliable Physical Unclonable Functions Using Data Retention Voltage of SRAM Cells, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015.

4 U. Ruhrmair, J. Martinez-Hurtado, X. Xu, C. Kraeh, C. Hilgers, D. Kononchuk, J. J. Finley, and W. P. Burleson. Virtual Proofs of Reality and their Physical Implementation. In IEEE Symp. on Security and Privacy, 2015.

5 S. Madduri, R. Vadlamani, W. Burleson and R. Tessier, A Monitor Interconnect and Support Subsystem for Multicore Processors, In Design, Automation and Test in Europe (DATE), 2009.

3.4 Metastable Latches: a Boon for Combined PUF/TRNG Designs

Jean-Luc Danger (ENST – Paris, FR)

This talk presents a way to take advantage of simple latches to generate both a TRNG and a PUF. The main idea is to place the latch very close to its metastable state. Hence a small noise will make the latch converge towards a stable state either ‘0’ ir ‘1’. This corresponds to the TRNG application. This concept can work only if many latches are placed in parallel as it is not possible to get a metastable state for most of the latches. At this point it is possible to use these former latches as a PUF as they are always in a stable state ‘0’ and ‘1’ which depends only on the process variation. This talk gives methods on how to obtain the set of N latches to have good TRNG (given an expected entropy) and a good PUF (in term of Bit error rate).

3.5 Security and Privacy of Non-Volatile Memories

Swaroop Ghosh (University of South Florida, US)

Non-volatile memories (NVM) such as Spin-Transfer Torque RAM (STTRAM), Resistive RAM and Domain Wall Memory have drawn significant attention due to complete elimination of bitcell leakage. In addition to plethora of benefits such as density, non-volatility, low-power and high-speed, majority of NVMs are also compatible with CMOS technology enabling easy integration. Although promising, I will show that NVMs bring new security and privacy challenges that were absent in their conventional volatile memory counterparts. Assuring data integrity and privacy against malicious attacks is particularly critical on deployed systems that are hard to maintain and enforce physical security. I will present two aspects to NVM security in Last Level Cache (LLC) using STTRAM as test case: (i) Data integrity which pertains to data corruption by malicious attack with the intention to launch denial-of-service. Such attacks exploit the fact that NVMs are fundamentally susceptible to ambient parameters such as magnetic field and temperature. I will describe these vulnerabilities and attack models, and, propose two micro-architectural techniques to assure data integrity under attack namely, cache bypassing and checkpointing. These techniques allow seamless computation in presence of attack at minimal design overhead. (ii) Data privacy which pertains to sensitive data such as keys and passwords being compromised. Storage such as Hard Disk Drive (HDD) has been the non-volatile part of memory system traditionally protected by encryption. Although effective, the latency associated with encryption makes it non-trivial for application in higher levels of memory stack such as LLC. I will present the vulnerabilities and attack models, and, propose two low-overhead techniques to maintain data privacy namely, Semi Non-Volatile Memory which is similar to NVM but with very low retention time so that the data vanishes after power is turned OFF, and, irreversible erasure of data at power down using residual charge from power rail.

3.6 Protecting Cryptographic Components in Hardware against Side-Channel and Fault-Injection Attacks

Tim Erhan Güneysu (Universität Bremen, DE), Amir Moradi, and Tobias Schneider

Joint work of Tobias Schneider, Amir Moradi, and Tim Güneysu
Main reference T. Schneider, A. Moradi, T. Güneysu, "ParTI – Towards Combined Hardware Countermeasures against Side-Channel and Fault-Injection Attacks", in Proc. of the 36th Annual Int'l Cryptology Conference – Advances in Cryptology (CRYPTO'16), LNCS, Vol. 9815, pp. 302–332, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-662-53008-5_11

Side-channel analysis and fault-injection attacks are known as major threats to any cryptographic implementation. Hardening cryptographic implementations with appropriate countermeasures is thus essential before they are deployed in the wild. However, countermeasures for both threats are of completely different nature: Side-channel analysis is mitigated by techniques that hide or mask key-dependent information while resistance against fault-injection attacks can be achieved by redundancy in the computation for immediate error detection. Since already the integration of any single countermeasure in cryptographic hardware often comes with significant costs with respect to performance and area, a combination of multiple countermeasures is expensive and often even associated with undesired side effects.

In this talk, we introduce a countermeasure for cryptographic hardware implementations that combines the concept of a provably-secure masking scheme based on threshold implementation with an error detecting approach for fault detection. As a case study, we apply our generic construction to the lightweight LED cipher. Our LED instance achieves first-order resistance against side-channel attacks combined with a fault detection capability that is superior to that of simple duplication for most error distributions at an increased area demand of 12%.

3.7 On the Synthesis of Side-Channel resistant Cryptographic Modules

Sorin A. Huss (TU Darmstadt, DE)

Over the last decades computer aided engineering (CAE) tools have been developed and improved in order to ensure a short time-to-market in the chip design business. Up to now, these design tools do not yet support an integrated design strategy for the development of side-channel resistant hardware implementations. In order to close this gap, a novel framework named AMASIVE (Adaptable Modular Autonomous SIde-Channel Vulnerability Evaluator) was developed. It supports the designer in implementing devices hardened against power attacks by exploiting novel security-driven synthesis methods. This talk explains how a design can be hardened in an automatic way by means of appropriate countermeasures which are tailored to the previously identified weaknesses. In addition to the theoretical introduction of the fundamental concepts, we demonstrate an application to the hardening of a complete hardware implementation of the block cipher PRESENT.

3.8 Hardware Security – Industrial Experiences

Michael Hutter (Cryptography Research Inc. – San Francisco, US)

Joint work of Elke de Mulder, Mark Marson, Peter Pearson, Andrew J. Leiserson, Megan A. Wachs, Gilbert Goodwill, Benjamin Jun, Josh Jaffe, Pankaj Rohatgi, Michael Hutter
Main reference G. Goodwill, B. Jun, J. Jaffe, P. Rohatgi, "A Testing Methodology for Side Channel Resistance Validation", in Proc. of the 2011 NIST Non-Invasive Attack Testing Workshop, Nara, Japan, 2011.
URL http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf

In this talk, I give an overview on the Test Vector Leakage Assessment Methodology (TVLA) as an efficient way to evaluate side-channel resistance. TVLA includes a set of specific and non-specific tests to determine leakage of intermediates of cryptographic algorithms. After that I I highlight state of the art methodologies in secure hardware design and will provide details on a secure logic style called Look-up Table based Masked Dual-Rail with Pre-charge Logic (LMDPL).

References

1 E. De Mulder, M. Hutter, M. E. Marson, P. Pearson. Using Bleichenbacher's Solution to the Hidden Number Problem to Attack Nonce Leaks in 384-bit ECDSA: extended version. Journal of Cryptographic Engineering 4(1), 2014.
2 G. Goodwill, B. Jun, J. Jaffe, P. Rohatgi. A Testing Methodology for Side Channel Resistance Validation. NIST Non-Invasive Attack Testing Workshop, Nara, Japan, 2011.
3 A. J. Leiserson, M. E. Marson, M. A. Wachs. Gate-Level Masking under a Path-Based Leakage Metric. In Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2014.

3.9 Security Oriented Codes

Osnat Keren (Bar-Ilan University, IL)

The cryptographic components as well as the on-chip memories are threatened by fault injection attacks. The faults induce errors that modify the behavior of the device. An attacker can use the information obtained from the incorrectly-functioning hardware to retrieve classified information, or, substitute correct information by a wrong one.

Fault injection attacks can be detected with relatively high probability by security oriented codes. Security oriented codes substantially differ from reliability oriented codes for which the error is assumed to be random and hence is of low multiplicity.

In this talk, we'll discuss the differences between reliability- and security-oriented codes in terms of the channel and error models, design requirements and efficiency criteria. We'll briefly review existing security oriented codes that aim to detect weak and strong fault injection attacks, and introduce open problems and design challenges.

References

1 N. Admaty, S. Litsyn and O. Keren,. Punctuating, Expurgating and Expanding the q-ary BCH Based Robust CodesIn IEEE Convention of Electrical and Electronics Engineers in Israel, 2012.

2 K. D. Akdemir, G. Hammouri, B. Sunar. Non-linear Error Detection for Finite State Machines. Computer Science, Information Security Applications (5932), 2009.

3 K. D. Akdemir, Z. Wang, M.,G. Karpovsky, B. Sunar. Design of Cryptographic Devices Resilient to Fault Injection Attacks Using Nonlinear Robust Codes. Fault Analysis in Cryptography, 2012.

4 R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs. Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors. In Advances in Cryptology, Eurocrypt, 2008.

5 S. Engelberg, O. Keren. A Comment on the Karpovsky-Taubin Code. In IEEE Trans. Info. Theory 57(12), 2011.

6 G. Gaubatz, B. Sunar, and M. G. Karpovsky. Non-linear Residue Codes for Robust Public-Key Arithmetic. In Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), 2006.

7 M.G.Karpovsky and A. Taubin,. A New Class of Nonlinear Systematic Error Detecting Codes. IEEE Trans. Info. Theory 50(8), 2004.

8 M. G.Karpovsky, K. Kulikowski, Z. Wang. Robust Error Detection in Communication and Computation Channels. In Int. Workshop on Spectral Techniques (Keynote paper), 2007.

9 M. G. Karpovsky and Z. Wang. Design of Strongly Secure Communication and Computation Channels by Nonlinear Error Detecting Codes. IEEE Trans Computers, 2014.

10 O. Keren and M. Karpovsky. Relations between the Entropy of a Source and the Error Masking Probability for Security Oriented Codes. IEEE Transactions on Communications 63(1), 2015.

11 K. J. Kulikowski, M. G. Karpovsky, A. Taubin. Robust Codes and Robust, Fault Tolerant Architectures of the Advanced Encryption Standard. Journal of System Architecture (53), 2007.

12 Y. Neumeier, O. Keren. Robust Generalized Punctured Cubic Codes. In IEEE Trans. on Information theory 60(5), 2014.

13 Y. Neumeier, O. Keren. A New Efficiency Criterion for Security Oriented Error Correcting Codes. In IEEE European Test Symp., 2014.

14 K. T. Phelps. A Combinatorial Construction of Perfect Codes. SIAM Journal Alg. disc Meth., 1983

15 I. Shumsky and O. Keren. Security-Oriented State Assignment. In Workshop on Trustworthy Manufacturing and Utilization of Secure Devices (TRUDEVICE), 2013.

16 I. Sumsky and O. Keren. Enhancement of Hardware Security by Hamming Ball Based State Assignment. Information Security Journal: A Global Perspective. Special issue on Trustworthy Manufacturing and Utilization 22, 5(6), 2013.

17 I. Shumsky, O. Keren and M. Karpovsky. Robustness of Security-Oriented Codes Under Non-Uniform Distribution of Codewords. Dependable Computing and Communications Symp. at the Intl. Conf. on Dependable Systems and Networks (DSN-DCCS), 2013.

18 B. Sunar, G. Gaubatz, E. Savas. Sequential Circuit Design for Embedded Cryptographic Applications Resilient to Adversarial Faults. In IEEE Trans. Computers 57(1), 2008.

19 V. Tomashevich, S. Srinivasan, F. Foerg, and I. Polian. Cross-level Protection of Circuits Against Faults and Malicious Attacks. In IEEE Intl. On-Line Testing Symp. (IOLTS), 2012.

20 V. Tomashevich, Y. Neumeier, R. Kumar, O. Keren and I. Polian. Protecting Cryptographic Hardware against Malicious Attacks by Nonlinear Robust Codes. In IEEE Intl. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT'14), 2014.

21 J. L. Vasil'ev. On Nongroup Close-Packed Codes. Probl. Kibernet (8), 1962.

22 Z. Wang and M. G.Karpovsky. Algebraic Manipulation Detection Codes and Their Application for Design of Secure Cryptographic Devices. In Intl. Symp. on On-Line Testing, 2011.

23 Z.Wang and M. G.Karpovsky. Reliable and Secure Memories Based on Algebraic Manipulation Correction Codes. In Intl. Symp. on On-line Testing, 2012.

24 Z. Wang, Mark G. Karpovsky, Konrad J. Kulikowski. Replacing Linear Hamming Codes by Robust Nonlinear Codes Results in a Reliability Improvement of Memories. In Intl. Symp. Dependable Computing, 2009.

25 Z. Wang, M. G. Karpovsky, K. Kulikowski. Design of Memories with Concurrent Error Detection and Correction by Non-Linear SEC-DED Codes. Journal of Electronic Testing 26(5), 2010.

3.10 Practical Aspects of Integrating PUFs in Industrial Applications

Roel Maes (Intrinsic-ID – Eindhoven, NL)

Joint work of Roel Maes, Vincent van der Leest, Erik van der Sluis, Frans Willems, Geert-Jan Schrijen

Physically unclonable functions (PUFs) have been studied in an academic research context for more than a decade. The last couple of years, industrial applications of PUFs have also started to appear, driven by the availability of commercial PUF IP and PUF-supported implementations, among others from Intrinsic-ID (Eindhoven, NL). The industrial application domains of PUFs are diverse, ranging from very-high security government and defense applications to extremely lightweight Internet-of-Things (IoT) platforms, and from end-point in-the-field sensors and controllers to cloud-based servers and (virtual) machines.

While instigated by academic research, the practical challenges of integrating PUFs in an industrial context do not run entirely parallel with academic research tracks, and are to some extent still unresolved or at least candidate for improvement. In this talk, a number of the more important practical aspects are aligned, and presented to the academic community. These highlighted points are collected from real-life experiences with industrial partners (aspiring to) integrating PUFs in their products, and include among others:

- the quest for more efficient error-correction techniques
- solutions for dealing with low-entropy PUFs
- insight and solutions for aging effects of PUFs
- design-for-test for PUF-based solutions
- appropriate use of PUFs in case of reset, zeriozation, ...
- health testing of PUFs
- ...

3.11 Side-Channel Security through Dynamic Reconfiguration: a Trade-off between Granularity and Side-Channel Resistance?

Nele Mentens (KU Leuven, BE)

Countermeasures against implementation attacks include hiding data and masking data. A class of countermeasures that has been proposed in the past decade are those that are based on dynamically reconfigurable architectures. This presentation gives an overview of the architectures and the technology that can be used as well as the options for generating new configuration data.

3.12 Fault Space Transformation: A Generic Approach to Counter Differential Fault Analysis and Differential Fault Intensity Analysis on AES-like Block Ciphers

Debdeep Mukhopadhyay (Indian Institute of Technology – Kharagpur, IN) and Sikhar Patranabis (Indian Institute of Technology – Kharagpur, IN)

Joint work of Sikhar Patranabis, Abhishek Chakraborty, Debdeep Mukhopadhyay, Partha Pratim Chakrabarti (Department of Computer Science and Engineering IIT Kharagpur, India)
Main reference S. Patranabis, A. Chakraborty, D. Mukhopadhyay, P. P. Chakrabarti, "Using State Space Encoding To Counter Biased Fault Attacks on AES Countermeasures", in IACR Cryptology ePrint Archive, Vol. 2015, pp. 806, 2015.
URL http://eprint.iacr.org/2015/806

Classical fault attacks such as Differential Fault Analysis (DFA) as well as biased fault attacks such as the Differential Fault Intensity Analysis (DFIA) have been a major threat to cryptosystems in recent times. DFIA combines principles of side channel analysis and fault attacks to try and extract the key using faulty ciphertexts only. Till date, no effective countermeasure that can thwart both classical DFA as well as DFIA based attacks has been reported in the literature to the best of our knowledge. In particular, traditional redundancy based countermeasures that assume uniform fault distribution are found to be vulnerable against DFIA due to its use of biased fault models. In this talk, we discuss our proposition of a novel generic countermeasure strategy that combines the principles of redundancy with that of fault space transformation to achieve security against both DFA and DFIA based attacks on AES-like block ciphers. As a case study, we have applied our proposed technique to to obtain temporal and spatial redundancy based countermeasures for AES-128, and have evaluated their security against both DFA and DFIA via practical experiments on a SASEBO-GII board. Results show that our proposed countermeasure makes it practically infeasible to obtain a single instance of successful fault injection, even in the presence of biased fault models.

3.13 PUFs in AMD64 CPUs and GPUs

Ruben Niederhagen (TU Eindhoven, NL), Daniel J. Bernstein, and Pol Van Aubel

Joint work of Pol Van Aubel, Daniel J. Bernstein, Tanja Lange, Ruben Niederhagen
Main reference P. Van Aubel, D. J. Bernstein, R. Niederhagen, "Investigating SRAM PUFs in Large CPUs and GPUs", in Proc. of the 5th Int'l Conf. on Security, Privacy, and Applied Cryptography Engineering (SPACE'15), LNCS, Vol. 9354, pp. 228–247, Springer, 2015; pre-print available at IACR.
URL http://dx.doi.org/10.1007/978-3-319-24126-5_14
URL http://eprint.iacr.org/2015/760

Physically unclonable functions (PUFs) provide data that can be used for cryptographic purposes: on the one hand randomness for the initialization of random-number generators; on the other hand individual fingerprints for unique identification of specific hardware components. However, today's off-the-shelf personal computers advertise randomness and individual fingerprints only in the form of additional or dedicated hardware.

This research introduces a new set of tools to investigate whether intrinsic PUFs can be found in PC components that are not advertised as containing PUFs. In particular, we investigate AMD64 CPU registers as potential PUF sources in the operating-system kernel, the bootloader, and the system BIOS; the CPU cache in the early boot stages; and shared memory on Nvidia GPUs. We found non-random non-fingerprinting behavior in several components but revealed usable PUFs in Nvidia GPUs.

3.14 Practical HW Security Attacks That Require Minimal Reverse Engineering

Elad Peer (CISCO Systems – Haifa, IL)

Practical attacks against secure hardware can roughly be divided into two cases: cloning, which usually requires extensive physical reverse engineering, and security breach attacks, which usually requires only little physical reverse engineering effort.

In this talk I demonstrate the validity of the above claim by showing two case studies that deal with security breach attacks. First, an attack that was issued against a complex SoC is described. In this attack a secure boot over the design was obtained. Careful analysis of the documentation reveals vulnerabilities both in the security standard which was the basis for this SoC, and in the implementation itself. Using those vulnerabilities an attack was developed and demonstrated using a laser fault injection or, alternatively, using an electromagnetic fault injection. A second case study briefly describes some straightforward low cost physical methods that enable retrieval of information from non volatile memories. Here, methods ranging from microscopy imaging to electric force microscopy are mentioned, and the usability of simple physical tools to overcome complex channels is demonstrated.

3.15 Trojans in Early Design Steps – An Emerging threat

Ilia Polian (Universität Passau, DE)

Historically, IT security concentrated on attack scenarios targeting software and communication networks, but more recently, the system hardware moved into the focus of attackers. Hardware-related threats are relevant even for extremely software-dominated systems, which still contain some amount of hardware on which the software runs; compromising this hardware makes the entire system vulnerable. Even worse, many software-centric security solutions rely on a hardware-based root of trust which stores secret keys and provides essential security functions; successful attacks on such root-of-trust blocks renders the entire security concept ineffective. With the emergence of paradigms like cyberphysical systems, internet of things, or Industrie 4.0 that connect the physical world, IT systems and global connectivity, hardware blocks are at risk to become the Achille's heel of entire infrastructures.

The presentation focuses on one emerging attack scenario: Hardware Trojans. These are malicious modification of system hardware with the purpose to gain control over its functionality and, e.g., be able to deactivate the affected block at the attacker's will ("kill switch"), or establish a side-channel to access confidential data processed by the device ("backdoor"). The term "hardware Trojans" was traditionally associated with threats stemming from external, untrusted foundries. However, the presentation is specifically concerned with Trojans that are introduced into the system during early design steps by a rogue in-house designer, by an external provider of intellectual property blocks integrated into the design, or even by an electronic design automation tool. The devastating damage potential of such attacks, the applicable countermeasures against them and their deficiencies are discussed. An under-investigated attack surface is the system specification which is created in a lengthy and complex process. If an attacker succeeds in planting a Trojan during the specification phase, such a Trojan is extremely hard to uncover and to detect, because any trusted reference is completely lacking.

3.16 Virtual Proofs of Reality and Their Physical Implementation

Ulrich Rührmair (Ruhr-Universität Bochum, DE)

We discuss the question of how physical statements can be proven over digital communication channels between two parties (a "prover" and a "verifier") residing in two separate local systems. Examples include: (i) "a certain object in the prover's system has temperature X °C", (ii) "two certain objects in the prover's system are positioned at distance X", or (iii) "a certain object in the prover's system has been irreversibly altered or destroyed". As illustrated by these examples, our treatment goes beyond classical security sensors in considering more general physical statements. Another distinctive aspect is the underlying security model: We neither assume secret keys in the prover's system, nor do we suppose classical sensor hardware in his system which is tamperresistant and trusted by the verifier. Without an established name, we call this new type of security protocol a "virtual proof of reality" or simply a "virtual proof" (VP).

In order to illustrate our novel concept, we discuss example VPs based on temperature sensitive integrated circuits, disordered optical scattering media, and quantum systems. The corresponding protocols prove the temperature, relative position, or destruction/modification of certain physical objects in the prover's system to the verifier. These objects (so-called "witness objects") are prepared by the verifier and handed over to the prover prior to the VP. Furthermore, we illustrate the practical validity of our method for all our optical and circuit-based VPs in detailed proof-of-concept experiments.

Our work touches upon, and partly extends, several established concepts in cryptography and security, including physical unclonable functions, quantum cryptography, interactive proof systems, and, most recently, physical zero-knowledge proofs.

3.17 Constructive Side-Channel Analysis

Werner Schindler (BSI – Bonn, DE)

Joint work of Kerstin Lemke-Rust, Christof Paar, Annelie Heuser, Michael Kasper, Marc Stöttinger, Werner Schindler, c.f. the bibliography

Power analysis is an essential part of evaluations of security implementations on smart cards and FPGAs etc. A successful attack shows that the implementation is vulnerable but usually does not give advice how to fix the problem.

In this talk we treat the stochastic approach, which combines the expertise of an engineer with methods from multivariate statistics. The stochastic approach is an effective attack method, which provides the leakage with regard to a vector space basis. This feature can also be used to identify the significant contributions of the leakage, which in turn supports target-oriented redesign. Moreover, apart from further benefits the stochastic approach allows to verify (within the limits of statistics) or to falsify leakage model assumptions.

References

1 W. Schindler, K. Lemke, C. Paar. A Stochastic Model for Differential Side Channel Analysis. In Cryptographic Hardware and Embedded Systems (CHES), 2005.

2 K. Lemke-Rust and C. Paar. Analyzing Side Channel Leakage of Masked Implementations with Stochastic Methods. In European Symp. on Research in Computer Security, 2007.

3 W. Schindler. Advanced Stochastic Methods in Side Channel Analysis on Block Ciphers in the Presence of Masking. Journal of Math. Crypt. (2), 2008.

4 M. Kasper, W. Schindler, M. Stöttinger. A Stochastic Method for Security Evaluation of Cryptographic FPGA Implementations. In Intl. Conf. on Field-Programmable Technology, 2010.

5 J. Doget, E. Prouff, M. Rivain, F.-X. Standaert. Univariate Side Channel Attacks and Leakage Modeling. Journal of Cryptographic Engineering (1), 2011.

6 A. Heuser, M. Kasper, W. Schindler, M. Stöttinger. How a Symmetry Metric Assists Side-Channel Evaluation – A Novel Model Verification Method for Power Analysis. In EUROMICRO Conf. on Digital System Design, 2011.

7 A. Heuser, W. Schindler, M. Stöttinger. Revealing Side-Channel Issues of Complex Circuits by High-Dimensional Leakage Models. In Design, Automation and Test in Europe (DATE), 2012.

8 A. Heuser, M. Kasper, W. Schindler, M. Stöttinger. A Difference Method for Side-Channel Analysis Exploiting High-Dimensional Leakage Models. In Topics in Cryptology – CT-RSA 2012.

9 W. Schindler. Understanding the Reasons for the Side-Channel Leakage is Indispensable for Secure Design (extended abstract). In PROOFS: Security Proofs for Embedded Systems, 2012.

3.18 Error Correction Schemes for Physical Unclonable Functions

Georg Sigl (TU München, DE)

Joint work of Michael Pehl, Matthias Hiller, Meng-Day Yu, Michael Weiner, Leandro Rodrigues Lima, Maximilian Birkner

Physical Unclonable Functions (PUFs) derive unique properties from manufacturing variations in integrated circuits. This can be used as a fingerprint for device identification as well as for secret key generation. This talk first shows the analogy between PUF key generation and the information theoretical model of deriving a secret key from a compound source. This model can be used to derive properties for syndrome coding and the generation of error correction information, i.e. the helper data. From recent theoretical results we can conclude that it is possible to generate an information theoretically secure error correction scheme. We have developed such a scheme which we call Systematic Low Leakage Coding (SLLC). It splits the PUF response in a key related part and in a masking part. Helper data are generated by exoring the masking part of the PUF response with the syndrome generated with a systematic code. This scheme provides inherent information theoretic security without the need of a hash function or strong extractor, and optimal asymptotic performance concerning maximum key size and minimum helper data size. The secrecy leakage is bounded by a small epsilon that goes to zero for PUFs with independent well distributed bits. The reference implementation for an ASIC application scenario shows that our scheme does not require the 47% hardware overhead for the hash function that is mandatory for the state-of-the-art approaches.

Another scheme, which is not optimal under the above assumptions, is called Differential Sequence Coding. With this scheme we can generate very efficient error correction for PUFs with low input bit reliability. The scheme picks reliable bits using pointers which measure the distance between reliable bits. After compression these pointers can be stored very efficiently, i.e. denser than a bit mask selecting the reliable bits from all bits. Combined with a Viterbi decoder and a lightweight hash function to counteract helper data manipulation attacks, this scheme outperforms other schemes considering helper data size, number of PUF bits, and slice count.

In order to perform analysis of PUF structures a FPGA cluster is presented offering 234 FPGAs which could be put in a temperature chamber. This setup is available at Fraunhofer AISEC and is offered to the community for generation of reliability data for FPGA PUF structures. With those data the quality of the PUF can be assessed a lot better than with simulation or testing on a few FPGAs. These data are needed further to generate models of PUFs and for proper dimensioning of error correction codes. With this setup we want to support the community in developing new PUFs and enable design of even more optimized key generation schemes.

References

1 M. Hiller, M.-D. (Mandel) Yu, M. Pehl. Systematic Low Leakage Coding for Physical Unclonable Functions. In ACM Asia Conf. on Computer and Communications Security, 2015.

2 M. Hiller, Georg Sigl: Increasing the Efficiency of Syndrome Coding for PUFs with Helper Data Compression. Design, Automation and Test in Europe (DATE), 2014.

3 M. Hiller, L. Rodrigues Lima, G. Sigl. Seesaw: An Area-Optimized FPGA Viterbi Decoder for PUFs. Digital System Design (DSD), 2014.

4 M. Hiller, M. Weiner, L. Rodrigues Lima, M. Birkner, G. Sigl. Breaking through Fixed PUF block Limitations with Differential Sequence Coding and Convolutional Codes. Intl. workshop on Trustworthy Embedded Devices, 2013.

3.19 No Place to Hide: Contactless Probing of Secret Data on FPGAs

Shahin Tajik (TU Berlin, DE)

Joint work of Heiko Lohrke, Jean-Pierre Seifert, Christian Boit, Shahin Tajik

Field Programmable Gate Arrays (FPGAs) have been the target of different physical attacks in recent years. Many different countermeasures have already been integrated into these devices to mitigate the existing vulnerabilities. However, there has not been enough attention paid to semi-invasive attacks from the IC backside due to the following reasons. First, the conventional semi-invasive attacks from the IC backside – such as laser fault injection and photonic emission analysis – cannot be scaled down without further effort to the very latest nanoscale technologies of modern FPGAs and programmable SoCs. Second, the more advanced solutions for secure storage, such as controlled Physically Unclonable Functions (PUFs), make the conventional memory-readout techniques almost impossible. In this paper, however, novel approaches have been explored: Attacks based on Laser Voltage Probing (LVP) and its derivatives, as commonly used in Integrated Circuit (IC) debug for nanoscale low voltage technologies, are successfully launched against a 60 nanometer technology FPGA. We discuss how these attacks can be used to break modern bitstream encryption implementations. Our attacks were carried out on a Proof-of-Concept PUF-based key generation implementation. To the best of our knowledge this is the first time that LVP is used to perform an attack on secure ICs.

3.20 Unlocking the Potential of Hardware Security

Mark M. Tehranipoor (University of Florida – Gainesville, US)

Hardware security has seen major growth over past decade. Significant amount of attention has been given to development of new security primitives, protection against malicious inclusion, secure architecture, hardware metering, etc. In this talk we present new applications to hardware security engineers. Some discussed topics are nano-enabled security, electronics clones, security rule checks for integrated circuits and non-electronics supply chain security.

4 Discussion Sessions

4.1 PUFs and Security Components

Domenic Forte (University of Florida – Gainesville, US)

Over the past 15 years, research in physical unclonable functions (PUFs) has been driven by the need for low-cost cryptographic key generation/storage and authentication. Yet, there still exist many challenges and unanswered questions regarding the practical limitations of PUFs, the future of PUF research, and the use of PUFs in emerging applications. More recently, the Internet of Things (IoT) has become another hot topic. While experts are excited about the applications enabled by IoT, there is skepticism surrounding our ability to maintain security and privacy at the resource constrained endpoint devices in IoT infrastructure.

The purpose of this session was to discuss these topics from the following three perspectives:

1. *Sources of Variability Impacting PUFs*– There are three categories of variability, each of which presents distinct challenges and opportunities. Process variation is responsible for the existence of PUFs. Although accurate knowledge of process variations (including their statistics) is needed to truly improve the overall quality of PUFs, such information is commonly withheld by foundries (with good reason) and varies with technology. Similarly, current design-for-manufacturability (DfM) techniques, which are geared towards suppression of process variation, are also left in the hands of the foundry. With respect to environmental variability, DC and AC sources need to be considered separately. DC

sources have been demonstrated as useful in executing attacks, but might be partially mitigated with sensors that monitor the PUF's environment. DC and AC sources both lead to PUF reliability problems. In the case of process and environmental variations, simulations are limited for PUF evaluation. Aging is the last major source of variability. On the one hand, accelerated aging has been shown as a way to reinforce PUF values for better reliability. On the other, burn-in is time consuming, costly, and could negatively impact non-PUF portions of ICs. An alternative strategy to deal with aging-induced reliability issues is error correction, but it requires accurate estimation of errors. One of the more promising concepts is anti-aging design, such as incorporation of sleep modes into PUFs.

2. *Emerging Applications of PUFs*– There are many practical challenges limiting the scope of PUFs, such as the lack of benchmarking capabilities and suitable metrics to fairly compare PUFs, vulnerability of strong PUFs to machine learning attacks, and the needs to erase and certify PUF challenge-response pairs. If ever realized, public PUFs based on the simulation-execution time gap might overcome some of these issues. However, a more promising opportunity is presented by expanding the PUF concept and rebranding PUFs as "unique objects". Unique objects are similar to silicon PUFs in that they harvest statistics for unique identification. However, they are based on optical, biological, and quantum phenomena, and harvest even more information. The most interesting applications for unique objects include monitoring the surrounding environment/conditions, virtual proofs of destruction, and tamper evidence. Unique objects might also be integrated into ICs through additive manufacturing/printing and/or investigation of novel materials. Realization of unique objects shall require interdisciplinary research.
3. *Design of Public Key Cryptography for IoT*– Cryptography in IoT is application-specific and should be governed by threat models, resource constraints, and semiconductor economics. Although IoT devices might contain complex processors, the needs of the application often outweigh those of security leaving little time, area, etc. left for crypto modules and protocols. While PUFs might seem like an excellent fit for IoT, they are still too fragile to replace conventional cryptography. Approaches that balance the trade-offs between latency, device power, and hardware might be even more promising.

4.2 Design for Security

Wayne P. Burleson (University of Massachusetts – Amherst, US) and Ilia Polian (Universität Passau, DE)

Security is emerging as a new target during design of circuits and systems. During the session, four research questions related to "design for security" were discussed. Below is the summary of the discussions.

(1) How to balance between security, quality, yield, cost and reliability of an integrated circuit?

The key difference of security from other design objectives is the presence of a human attacker and, as a consequence, a large diversity of attacks. As a consequence, any countermeasures have to be considered risk management under cost constraints, rather than bullet-proof protections. Prerequisites for designing systematic countermeasures are models of threats (resources at an attacker's disposal), security requirements and available assets. Based on them, certification procedures (including but not limited to formal proofs) can be developed in compliance with existing (legal or technical) regulations. However, all models and abstractions have limitations, and better models of channels to be protected and of attack vectors would be useful. Lessons from safety engineering, debug and reliability should be considered, yet these fields lack some aspects that are essential for secure design. An area which is particularly hard to model is insider attacks, because the techniques employed for such attacks and the consequences of these attacks largely depend on the specific attacker's creativity and malicious intent; it is extremely difficult to create models that are valid for all or most environments.

(2) Can Hardware Trojans be detected by low-cost approaches with sufficient confidence?

Security against hardware Trojans stemming from various sources is difficult to achieve, and it requires a secure design chain, root-of-trust modules and authenticated CAD tools. Anti-Trojan approaches can be categorized into prevention of their effects and detection of their operation; the latter is much easier if a golden (Trojan-free) model is available (which is often not the case). In general, measures that provide resilience against Trojans and other threats (e.g., power-grid instabilities) are desired. A Trojans that protects itself against detection can be compared with "kleptography" (stealing information without being noticed) in context of public-key cryptography. A particularly hard class of Trojans includes those inserted above HDL level; their detection perhaps requires a partition of the system functionality into trusted domains (or trusted IP). Regarding the economic dimension of Trojans (and other security threats) it was discussed whether market forces are sufficient to motivate companies to integrate security features, or whether regulation or specific incentives from the lawmaker are needed. It is possible that market forces are only sufficient to cover very simple, easy-to-understand threats.

(3) HW security threat models in context of larger-scale threats (e.g., network, software, social)

Hardware is the foundation of and "the ultimate insider" in electronic systems. Hardware protects software, and a hardware-related attack can have a very broad impact. The expectations on hardware obsolescence are not always met, and we may be facing attacks on decades-old hardware blocks designed without security considerations. The perceived threat may not always correspond to the actual level of risk; in particular, improved security does not imply larger customer demand. Therefore, security measures might be difficult to justify economically (but this may change as soon as first large-scale real-world attacks will be reported). Hardware vulnerabilities are difficult to study, and one reason is that meaningful investigations require knowledge of internal industry items and procedures that are considered proprietary or secret. In any case, it is important to understand the application under attack and the (real or perceived) threats; one example is unauthorized vehicle tuning, which can compromise safety and can lead to increased warranty costs.

(4) How to build automated verification tools for security-critical hardware components that check functional and security aspects?

The construction of such tools requires collaboration with other communities. The basic techniques are, in general, known; these include certification, debug and formal verification. Both a sound theory and efficient implementations are needed to make such approaches practical. This requires expressive and reliable metrics that quantify security threats like side-channel vulnerabilities. It would be ultimately desirable to have "security by design" circuits which do not need separate verification.

4.3 Side Channel Analysis

Debdeep Mukhopadhyay (Indian Institute of Technology – Kharagpur, IN) and Ilia Polian (Universität Passau, DE)

It is an open area of research as to how to model side-channel attacks such as to balance between accuracy and simplicity. It was discussed that the starting test to detect leakage should be unspecific tests like Test Vector Leakage Assessment (TVLA) tests. These tests which are based on the statistical T-test, are generic and independent of the underlying leakage as it provides an estimate of the fact whether a given circuit leaks information. In cases where the T-test as originally defined does not estimate a leakage, as it estimates the first order leakage (which should be enough for most applications), further tests for higher order leakage may be performed.

For specific tests the underlying leakage model plays a crucial role, and it was discussed on what should be ideal leakage models. On one hand there could be simple linear leakage models, like Hamming weight or linear combination of the values of the bits of a target register; while on the other hand the leakage models could be non-linear to better estimate the electrical models of the device under test.

It was stressed that sometimes our designs of countermeasures are over-designed, as they are based on very strong assumptions of the adversarial power. Like we often consider insider attackers with access to several internals of a circuit, which may not be true in practice. Further effect of side channels on larger designs were deliberated upon. The source of the leakage depending on the attack model should be identified and accordingly countermeasures should be designed. It was emphasized that the overheads of the countermeasures should be estimated with the overall System-on-Chip (SOC) in perspective, as since the crypto-core is a small part of the entire design, the overhead on the crypto-core is a small percentage of the entire SOC. For larger designs we should be more precise on the source of leakage and then design suitable countermeasures. For proper understanding on leakage due to the underlying electrical phenomenon, it was suggested there should be active collaborations with device physics experts.

The next issue which was discussed was modelling methods for fault analysis. It was felt that there should be extensive study on understanding of fault injection techniques, considering a wide variety of fault injection methods, with accompanying validation with silicon results. The models will estimate the precision of faults for various injection methods. Also for modelling of fault attacks, definition of exploitable faults and corresponding leakage thereof wrt. different attack methods like Differential Fault Analysis (DFA), Differential

Fault Intensity Analysis (DFIA) needs to be defined. Further, while analyzing the attacks the effect of the fault cone which leads to an error need to be observed too.

The importance of development of CAD tools for automating the fault attack procedure for given ciphers was stressed to analyze new ciphers. Information theoretic tools were emphasized for discovering the attacks, along with development of algebraic analysis and SAT solvers. These tools require defining suitable metrics for fault analysis, defining exploitable faults in the pursuit of detecting weaknesses of design architectures and synthesizing them into stronger architectures. The fault simulation tools in the reliability community could be used to study the propagation of faults.

The next topic that was discussed was the extent of reverse engineering required for developing practical attacks. Often attacks, like Differential Power Analysis (DPA) or DFA use information which can be obtained by simple, low effort reverse engineering. These could include ascertaining timing by glitches, power analysis, physical inspection, even social engineering. While many of these techniques may be costly, the use of expensive reverse engineering methods for developing practical attacks could be a topic of future research .

The discussions were concluded with the topic of application of coding theory, both linear and non-linear codes, against side channel attacks, fault attacks, and Trojans. The use of linear codes with complementary duals (LCD) could be a useful technique. However, the contradiction between various attack vectors should be studied: for example, error detection techniques used for fault analysis could open side channel leakage sources. Thus it is important to develop holistic countermeasures, and the effectivity of the countermeasures against these threats also should be modelled to increase confidence in them.

Participants

- Lejla Batina
Radboud Univ. Nijmegen, NL
- Georg T. Becker
Ruhr-Universität Bochum, DE
- Christian Boit
TU Berlin, DE
- Jan Burchard
Universität Freiburg, DE
- Wayne P. Burleson
University of Massachusetts – Amherst, US
- Jean-Luc Danger
ENST – Paris, FR
- Linus Feiten
Universität Freiburg, DE
- Domenic Forte
University of Florida – Gainesville, US
- Fatemeh Ganji
TU Berlin, DE
- Swaroop Ghosh
University of South Florida, US
- Jorge Guajardo Merchan
Robert Bosch LLC – Pittsburgh, US
- Tim Erhan Güneysu
Universität Bremen, DE
- Sorin A. Huss
TU Darmstadt, DE
- Michael Hutter
Cryptography Research Inc. – San Francisco, US
- Ramesh Karri
New York University, US
- Osnat Keren
Bar-Ilan University, IL
- Tanja Lange
TU Eindhoven, NL
- Roel Maes
Intrinsic-ID – Eindhoven, NL
- Nele Mentens
KU Leuven, BE
- Debdeep Mukhopadhyay
Indian Institute of Technology – Kharagpur, IN
- Ruben Niederhagen
TU Eindhoven, NL
- Sikhar Patranabis
Indian Institute of Technology – Kharagpur, IN
- Elad Peer
CISCO Systems – Haifa, IL
- Ilia Polian
Universität Passau, DE
- Wenjing Rao
Univ. of Illinois – Chicago, US
- Francesco Regazzoni
University of Lugano, CH
- Ulrich Rührmair
Ruhr-Universität Bochum, DE
- Kazuo Sakiyama
The Univ. of Electro-Communications – Tokyo, JP
- Werner Schindler
BSI – Bonn, DE
- Georg Sigl
TU München, DE
- Shahin Tajik
TU Berlin, DE
- Mark M. Tehranipoor
University of Florida – Gainesville, US

Report from Dagstuhl Seminar 16221

Algorithms for Optimization Problems in Planar Graphs

Edited by

Jeff Erickson[1], Philip N. Klein[2], Dániel Marx[3], and Claire Mathieu[4]

1 **University of Illinois – Urbana-Champaign, US,** jeffe@illinois.edu
2 **Brown University – Providence, US,** klein@brown.edu
3 **Hungarian Academy of Sciences – Budapest, HU,** dmarx@cs.bme.hu
4 **ENS – Paris, FR,** cmathieu@di.ens.fr

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 16221 "Algorithms for Optimization Problems in Planar Graphs". The seminar was held from May 29 to June 3, 2016. This report contains abstracts for the recent developments in planar graph algorithms discussed during the seminar as well as summaries of open problems in this area of research.

Seminar May 29–June 3, 2016 – http://www.dagstuhl.de/16221
1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity
Keywords and phrases Algorithms, planar graphs, theory, approximation, fixed-parameter tractable, network flow, network design, kernelization
Digital Object Identifier 10.4230/DagRep.6.5.94

1 Executive summary

Jeff Erickson
Philip N. Klein
Dániel Marx
Claire Mathieu

There is a long tradition of research in algorithms for optimization problems in graphs, including work on many classical problems, both polynomial-time solvable problems and NP-hard problems, e.g. shortest paths, maximum flow and minimum cut, matching, T-joins, disjoint paths, traveling salesman, Steiner tree, graph bisection, vehicle routing, facility location, k-center, and maximum cut. One theme of such research addresses the complexity of these problems when the input graph is required to be a planar graph or a graph embedded on a low-genus surface.

There are three reasons for this theme. First, optimization problems in planar graphs arise in diverse application areas. Second, researchers have discovered that, by exploiting the planarity of the input, much more effective algorithms can be developed – algorithms that are faster or more accurate than those that do not exploit graph structure. Third, the study of algorithms for surface-embedded graphs drives the development of interesting algorithmic techniques. One source of applications for planar-graph algorithms is geographic problems. Road maps are nearly planar, for example, so distances in planar graphs can model, e.g., travel times in road maps. Network design in planar graphs can be used to model scenarios in which cables must be run under roads. Planar graphs can also be used to model metrics on the earth's surface that reflect physical features such as terrain; this aspect of planar graphs has been used in studying wildlife corridors. Another source of applications is image processing. Some algorithms for problems such as image segmentation and stereo involve finding minimum cuts in a grid in which each vertex represents a pixel. Sometimes an aggregation technique (superpixels) coalesces regions into vertices, turning the grid into an arbitrary planar graph. A third example application is VLSI. Algorithmic exploitation of a planar embedding goes back at least to the introduction of maximum flow by Ford and Fulkerson in 1956. Current research can be divided in three parts. For polynomial-time-solvable problems, such as maximum flow, shortest paths, matching, and min-cost circulation, researchers seek planarity-exploiting algorithms whose running times beat those of general-graph algorithms, ideally algorithms whose running times are linear or nearly linear. For NP-hard problems, there are two strategies: fixed-parameter algorithms and approximation algorithms. In all three research subareas, there has recently been significant progress. However, many researchers are expert in only one or two subareas. This Dagstuhl Seminar brought together researchers from the different subareas, to introduce them to techniques from subareas that might be unfamiliar, and to foster collaboration across the subareas. The seminar will thus help to spur further advances in this active and growing area. The scientific program of the seminar consisted of twenty-two talks. Four of these talks were longer (60–90 minute) tutorials overviewing the three main areas of the seminar:

- *Polynomial-time algorithms:* "Tutorial on embedded graph algorithms" (Jeff Erickson) and "Monge property, dense distance graphs and speeding-up max-flow computations in planar graphs" (Piotr Sankowski)
- *Approximation schemes:* "Some techniques for approximation schemes on planar graphs" (Philip Klein)
- *Fixed-parameter tractability:* "The square-root phenomenon in planar graphs" (Dániel Marx)

One of the main goals of the seminar was to encourage collaboration between the three communities, and these well-received tutorials helped by introducing the basics of each of these topics.

The rest of the talks were 25-minute presentations on recent research of the participants. The time between lunch and the afternoon coffee break was left open for individual discussions and collaborations in small groups. An open-problem session was organized on Monday morning. Notes on the presented problems can be found in this report.

2 Table of Contents

Executive summary

Overview of Talks

Open problems

3 Overview of Talks

3.1 A PTAS for Planar Group Steiner Tree via Spanner Bootstrapping and Prize Collecting

Mohammad Hossein Bateni (Google – New York, US)

Joint work of Mohammad Hossein Bateni, Erik D. Demaine, MohammadTaghi Hajiaghayi, Dániel Marx
Main reference M. H. Bateni, E. D. Demaine, M. Hajiaghayi, D. Marx, "A PTAS for Planar Group Steiner Tree via Spanner Bootstrapping and Prize Collecting," in Proc. of the 48th Annual ACM SIGACT Symp. on Theory of Computing, pp. 570–583, ACM, 2016; pre-print available from author's webpage.
URL http://dx.doi.org/10.1145/2897518.2897549
URL www.cs.bme.hu/~dmarx/papers/bateni-stoc2016-groupsteiner.pdf

We present the first polynomial-time approximation scheme (PTAS), i.e., $(1+\varepsilon)$-approximation algorithm for any constant $\varepsilon > 0$, for the planar group Steiner tree problem (in which each group lies on a boundary of a face). This result improves on the best previous approximation factor of $O(\log n(\log \log n)^{O(1)})$. We achieve this result via a novel and powerful technique called spanner bootstrapping, which allows one to bootstrap from a superconstant approximation factor (even superpolynomial in the input size) all the way down to a PTAS. This is in contrast with the popular existing approach for planar PTASs of constructing light-weight spanners in one iteration, which notably requires a constant-factor approximate solution to start from. Spanner bootstrapping removes one of the main barriers for designing PTASs for problems which have no known constant-factor approximation (even on planar graphs), and thus can be used to obtain PTASs for several difficult-to-approximate problems.

Our second major contribution required for the planar group Steiner tree PTAS is a spanner construction, which reduces the graph to have total weight within a factor of the optimal solution while approximately preserving the optimal solution. This is particularly challenging because group Steiner tree requires deciding which terminal in each group to connect by the tree, making it much harder than recent previous approaches to construct spanners for planar TSP by Klein (FOCS'05 & SICOMP'08), subset TSP by Klein (STOC'06), Steiner tree by Borradaile, Klein, and Mathieu (SODA'07 & TALG'09), and Steiner forest by Bateni, Hajiaghayi, and Marx (STOC'10 & JACM'11) (and its improvement to an efficient PTAS by Eisenstat, Klein, and Mathieu (SODA'12)). The main conceptual contribution here is realizing that selecting which terminals may be relevant is essentially a complicated prize-collecting process: we have to carefully weigh the cost and benefits of reaching or avoiding certain terminals in the spanner. Via a sequence of involved prize-collecting procedures, we can construct a spanner that reaches a set of terminals that is sufficient for an almost-optimal solution.

Our PTAS for planar group Steiner tree implies the first PTAS for geometric Euclidean group Steiner tree with obstacles, as well as a $(2+\varepsilon)$-approximation algorithm for group TSP with obstacles, improving over the best previous constant-factor approximation algorithms. By contrast, we show that planar group Steiner forest, a slight general- ization of planar group Steiner tree, is APX-hard on planar graphs of treewidth 3, even if the groups are pairwise disjoint and every group is a vertex or an edge.

3.2 Subgraph isomorphism on planar graphs, and related problems

Hans L. Bodlaender (Utrecht University, NL)

Joint work of Hans L. Bodlaender, Jesper Nederlof, Tom van der Zanden

In this talk, we show that the problem, given two planar graphs G and H, to decide if G is isomorphic with a subgraph of H can be solved in $2^{O(n/\log n)}$ time. We also show that this is optimal, assuming the exponential time hypothesis. A similar result holds for other embedding problems, including induced subgraph, minor and induced minor (and weighted variants), and for other graph classes, including graphs avoiding some fixed minor. This is joint work by Hans Bodlaender, Jesper Nederlof and Tom van der Zanden.

3.3 Approximating connectivity domination in weighted bounded-genus graphs

Vincent Cohen-Addad (CNRS / ENS – Paris, FR)

Joint work of Vincent Cohen-Addad, Éric Colin de Verdière, Philipp N. Klein, Claire Mathieu, David Meierfrankenfeld

We present a framework for addressing several problems on weighted planar graphs and graphs of bounded genus. With that framework, we derive polynomial-time approximation schemes for the following problems in planar graphs or graphs of bounded genus: edge-weighted tree cover and tour cover; vertex-weighted connected dominating set, maximum-weight-leaf spanning tree, and connected vertex cover. In addition, we obtain a polynomial-time approximation scheme for feedback vertex set in planar graphs. These are the first polynomial-time approximation schemes for all those problems in weighted embedded graphs. (For unweighted versions of some of these problems, polynomial-time approximation schemes were previously given using bidimensionality.)

3.4 Independent sets in planar graphs

Zdenek Dvorak (Charles University – Prague, CZ)

Joint work of Zdenek Dvorak, Matthias Mnich

It is a long-standing open problem in the algorithmic theory of planar graphs whether there exists a polynomial-time algorithm deciding if an n-vertex planar graph has an independent set of size greater than $n/4$. (An independent set of size at least $n/4$ is guaranteed by the Four Color Theorem.)

In joint work with Matthias Mnich, we investigate related (easier) questions of similar nature. For example we show that

- The problem can be solved under a variety of additional restrictions, e.g., when the considered graphs have maximum degree at most 4 or when they avoid 4- or 5-cycles; and,

- It is possible to decide whether an n-vertex planar triangle-free graph has an independent set of size at least $n/3 + k$ in time $2^{O(\sqrt{k})}n^{O(1)}$, which is analogously related to Grotzsch's theorem.

3.5 Tutorial on embedded graph algorithms

Jeff Erickson (University of Illinois – Urbana-Champaign, US)

We consider several fundamental algorithmic tools for exact polynomial-time algorithms for graphs embedded on surfaces. Specific topics include combinatorial embeddings and duality, Euler's formula, the greedy tree-cotree decomposition, systems of loops and cycles, shortest noncontractible and nonseparating cycles, multiple-source shortest paths, homology and homology annotation, enforcing uniqueness of shortest paths, and covering spaces. As applications of these building blocks, we sketch recent algorithms to compute minimum (s, t)-cuts [Chambers, Erickson, Nayyeri 2009], Gomory-Hu trees [Borradaile, Eppstein, Nayyeri, and Wulff-Nilson 2016], and shortest cycle bases [Borradaile, Chambers, Fox, and Nayyeri 2016] in surface-embedded graphs.

3.6 On Temporal Graph Exploration

Thomas Erlebach (University of Leicester, GB)

Joint work of Thomas Erlebach, Michael Hoffmann, Frank Kammer

A temporal graph is a graph in which the edge set can change from step to step. The temporal graph exploration problem TEMPEX is the problem of computing a foremost exploration schedule for a temporal graph, i.e., a temporal walk that starts at a given start node, visits all nodes of the graph, and has the smallest arrival time. We consider only temporal graphs that are connected at each step. For such temporal graphs with n nodes, we show that it is NP-hard to approximate TEMPEX with ratio $O(n^{1-\epsilon})$, and that there are temporal graphs whose exploration requires $O(n^2)$ steps. The underlying graph (i.e. the graph that contains all edges that are present in the temporal graph in at least one step) used in these constructions is dense, which leads to the interesting question of studying TEMPEX for temporal graphs whose underlying graph is planar. We show that even such temporal graphs may require $\Omega(n \log n)$ steps, and that they can always be explored in $O(n^{1.8} \log n)$ steps. For the special case of $2 \times n$ grids, we show that $O(n \log^3 n)$ steps always suffice.

3.7 A Polynomial-time Bicriteria Approximation Scheme for Planar Bisection

Kyle Jordan Fox (Duke University – Durham, US)

Joint work of Kyle Jordan Fox, Philip N. Klein, Shay Mozes

Given an undirected graph with edge costs and node weights, the minimum bisection problem asks for a partition of the nodes into two parts of equal weight such that the sum of edge costs between the parts is minimized. We give a polynomial time bicriteria approximation scheme for bisection on planar graphs.

Specifically, let W be the total weight of all nodes in a planar graph G. For any constant $\varepsilon > 0$, our algorithm outputs a bipartition of the nodes such that each part weighs at most $W(1/2+\varepsilon)$ and the total cost of edges crossing the partition is at most $(1+\varepsilon)$ times the total cost of the optimal bisection. The previously best known approximation for planar minimum bisection, even with unit node weights, was $O(\log n)$. Our algorithm actually solves a more general problem where the input may include a target weight for the smaller side of the bipartition.

3.8 Turing Kernelization for Finding Long Paths and Cycles in Restricted Graph Classes

Bart Jansen (TU Eindhoven, NL)

The k-Path problem asks whether a given undirected graph has a (simple) path of length k. We prove that k-Path has polynomial-size Turing kernels when restricted to planar graphs, graphs of bounded degree, claw-free graphs, or to $K_{3,t}$-minor-free graphs. This means that there is an algorithm that, given a k-Path instance (G, k) belonging to one of these graph classes, computes its answer in polynomial time when given access to an oracle that solves k-Path instances of size polynomial in k in a single step. Our techniques also apply to k-Cycle, which asks for a cycle of length at least k.

3.9 Paradigms for obtaining approximation schemes for planar graphs

Philip N. Klein (Brown University – Providence, US)

In addressing an NP-hard problem in combinatorial optimization, one way to cope is to use an *approximation scheme*, an algorithm that, for any given $\epsilon > 0$, produces a solution whose value is within a $1+\epsilon$ factor of optimal. For many problems on graphs, obtaining such accurate approximations is NP-hard if the input is allowed to be any graph but is tractable if the input graph is required to be planar.

Research on polynomial-time approximation schemes for optimization problems in planar graphs goes back to the pioneering work of Lipton and Tarjan (1977) and Baker (1983). Since then, however, the scope of problems amenable to approximation has broadened considerably. In this talk I will outline some of the approaches used, especially those that have led to recent results.

3.10 The Square Root Phenomenon in Planar Graphs – Survey and New Results

Dániel Marx (Hungarian Academy of Sciences – Budapest, HU)

Most of the classical NP-hard problems remain NP-hard when restricted to planar graphs, and only exponential-time algorithms are known for the exact solution of these planar problems. However, in many cases, the exponential-time algorithms on planar graphs are significantly faster than the algorithms for general graphs: for example, 3-Coloring can be solved in time $2^{O(\sqrt{n})}$ in an n-vertex planar graph, whereas only $2^{O(n)}$-time algorithms are known for general graphs. For various planar problems, we often see a square root appearing in the running time of the best algorithms, e.g., the running time is often of the form $2^{O(\sqrt{n})}$, $n^{O(\sqrt{k})}$, or $2^{O(\sqrt{k})} \cdot n$. By now, we have a good understanding of why this square root appears. On the algorithmic side, most of these algorithms rely on the notion of treewidth and its relation to grid minors in planar graphs (but sometimes this connection is not obvious and takes some work to exploit). On the lower bound side, under a complexity assumption called Exponential Time Hypothesis (ETH), we can show that these algorithms are essentially best possible, and therefore the square root has to appear in the running time.

In the talk, I will present a survey of the basic algorithmic and complexity results, and discuss some of the very recent developments in the area.

3.11 Local search yields an approximation scheme for uniform facility location in edge-weighted planar graphs

Claire Mathieu (ENS – Paris, FR)

Joint work of Vincent Cohen-Addad, Philip N. Klein, Claire Mathieu

We present a polynomial-time approximation scheme (PTAS) for uniform facility location in edge-weighted planar graphs. This is the easiest of several results showing the good performance of local search in Euclidean and minor-free metrics.

3.12 Computing the minimum cut of a weighted directed planar graph

Shay Mozes (Interdisciplinary Center Herzliya, IL)

Joint work of Shay Mozes, Cyril Nikolaev, Yahav Nussbaum, Oren Weimann

We give an $O(n \log \log n)$ time algorithm for computing the minimum cut (or equivalently, the shortest cycle) of a weighted directed planar graph. This improves the previous fastest $O(n \log^2 n)$ solution [SODA'04]. Interestingly, while in undirected planar graphs both min-cut and min st-cut have $O(n \log \log n)$-time solutions [ESA'11, STOC'11], in directed planar graphs our result makes min-cut faster than min st-cut, which currently requires $O(n \log n)$ [J. ACM'09].

3.13 Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering

Marcin Pilipczuk (University of Warsaw, PL)

Joint work of Fedor Fomin, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, Saket Saurabh
Main reference F. V. Fomin, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, "Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering," arXiv:1604.05999v1 [cs.DS], 2016.
URL http://arxiv.org/abs/1604.05999v1

We prove the following theorem. Given a planar graph G and an integer k, it is possible in polynomial time to randomly sample a subset A of vertices of G with the following properties: (i) A induces a subgraph of G of treewidth $\sqrt{k} \log k$, and (ii) for every connected subgraph H of G on at most k vertices, the probability that A covers the whole vertex set of H is at least $(2^{O(\sqrt{k} \log^2 k)} \cdot n^{O(1)})^{-1}$, where n is the number of vertices of G.

Together with standard dynamic programming techniques for graphs of bounded treewidth, this result gives a versatile technique for obtaining (randomized) subexponential parameterized algorithms for problems on planar graphs, usually with running time bound $2^{O(\sqrt{k} \log^2 k)} n^{O(1)}$. The technique can be applied to problems expressible as searching for a small, connected pattern with a prescribed property in a large host graph, examples of such problems include Directed k-Path, Weighted k-Path, Vertex Cover Local Search, and Subgraph Isomorphism, among others. Up to this point, it was open whether these problems can be solved in subexponential parameterized time on planar graphs, because they are not amenable to the classic technique of bidimensionality. Furthermore, all our results hold in fact on any class of graphs that exclude a fixed apex graph as a minor, in particular on graphs embeddable in any fixed surface.

Preprint available at http://arxiv.org/abs/1604.05999.

3.14 Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams

Michal Pilipczuk (University of Warsaw, PL)

Joint work of Dániel Marx, Michal Pilipczuk

We study a general family of facility location problems defined on planar graphs and on the 2-dimensional plane. In these problems, a subset of k objects has to be selected, satisfying certain packing (disjointness) and covering constraints. We show that, for each of these problems, the $n^{O(k)}$ time brute force algorithm of selecting k objects can be improved to $n^{O(\sqrt{k})}$ time. The algorithm is based on an approach that was introduced recently in the design of geometric QPTASs, but we show that it can be applied also for exact and parameterized algorithms and for planar graphs. Namely, the idea is to focus on the Voronoi diagram of a hypothetical solution of k objects, guess a balanced separator cycle of this Voronoi diagram to obtain a set that separates the solution in a balanced way, and then recurse on the resulting subproblems. Finally, we also give evidence that the obtained algorithms are essentially optimal, under the Exponential Time Hypothesis.

The extended abstract of the paper appeared in the proceedings of ESA 2015.

3.15 Monge property, dense distance graphs and speeding up max-flow computations in planar graphs

Piotr Sankowski (University of Warsaw, PL)

In my talk, I will introduce the core technique that was used in a series of papers to speed-up max-flow computations in planar graphs. Min-cuts in planar graphs are related to shortest paths via duality. This allows to use simpler shortest path computations for finding minimum-cuts. Especially, it is possible to use a faster implementation of Dijkstra algorithm created by Fakcharoenphol and Rao in 2001. This implementation uses the fact that one do not need to search through shortest paths starting in the same source that would cross. In the algorithm one creates so called dense distance graphs, and needs to search only through square root of edges in such graphs. I will introduce the ideas behind the following three applications of this technique:

- computing all pairs min-cuts in undirected planar graphs in almost linear time by Borradaile, S. and Wulff-Nilsen '10,
- computing s-t max-flows in undirected planar graphs in $O(n \log\log n)$ time by Italiano, Nussbaum, S. and Wulff-Nilsen '11,
- computing single source-all sinks max flows in directed planar graphs by Łącki, Nussbaum, S. and Wulff-Nilsen '12.

3.16 Subexponential algorithms for rectilinear Steiner tree and arborescence problems

Saket Saurabh (The Institute of Mathematical Sciences, IN)

Joint work of Fedor Fomin, Sudeshna Kolay, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh

A rectilinear Steiner tree for a set T of points in the plane is a tree which connects T using horizontal and vertical lines. In the Rectilinear Steiner Tree problem, input is a set T of n points in the Euclidean plane and the goal is to find an rectilinear Steiner tree for T of smallest possible total length. A rectilinear Steiner arborescence for a set T of points and root r in T is a rectilinear Steiner tree S for T such that the path in S from r to any point t in T is a shortest path. In the Rectilinear Steiner Arborecence problem the input is a set T of n points in the Euclidean plane, and a root r in T, the task is to find an rectilinear Steiner arborescence for T, rooted at r of smallest possible total length. In this talk, we give the first subexponential time algorithms for both problems. Our algorithms are deterministic and run in $2^{O(\sqrt{n}\log n)}$ time.

3.17 Embedding Planar Graphs into Low-Treewidth Graphs with Applications to Efficient Approximation Schemes for Metric Problems

Aaron Schild (Berkeley, US)

We give a stretch-$(1+\epsilon)$ embedding of edge-weighted planar graphs of bounded aspect ratio into bounded-treewidth graphs. We use this construction to obtain the first efficient bicriteria approximation schemes for weighted planar graphs addressing a metric generalization of dominating set, r-domination, and a metric generalization of independent set, r-independent set. The approximation schemes employ a metric generalization of Baker's framework based on our embedding result.

3.18 Match-And-Merge: A New Greedy Framework for Maximum Planar Subgraphs

Andreas Schmid (MPI für Informatik – Saarbrücken, DE)

In the maximum planar subgraph (MPS) problem, we are given a graph G, and our goal is to find a planar subgraph H with the maximum number of edges. Besides being a basic problem in graph theory, MPS has many applications including, for instance, circuit design, factory layout, and graph drawing, so it has received a lot of attention from both theoretical and empirical literature. Since the problem is NP-hard, past research has focused on approximation algorithms. The current best known approximation ratio is 4/9 obtained two decades ago based on, roughly speaking, computing as many edge-disjoint triangles in an input graph as possible. The factor 4/9 is also the limit of this "disjoint triangles” approach. We propose two new angles on MPS and provide some evidences that they might lead to improvements over this two-decade-old barrier.

Our first contribution is to initiate a systematic study of a class of greedy algorithms for MPS. Our class of algorithms is rich: All known greedy algorithms for MPS fit into our framework. We argue that these algorithms are unable to perform better than a 7/18-approximation and then show that a slight modification gives a 13/33-approximations, therefore being the first greedy algorithm that beats 7/18.

To facilitate an analytical task in our framework, we formulate a new optimization problem, that we call the Maximum Planar Triangles (MPT) problem. In MPT we are given an input graph and are interested in computing a subgraph that admits a planar embedding with as many triangular faces as possible. We show that MPT is NP-hard and quantify the connection between the two problems. This approach allows potentially up to a 1/2-approximation for MPS, provided the existence of a 1/4-approximation for MPT.

3.19 Face-rooted plane topological minors

Dimitrios M. Thilikos (University of Athens, GR)

Joint work of Petr Golovach, Spyridon Maniatis, Dimitrios M. Thilikos

Let G and H be a (not necessarily connected) plane graphs and let ϕ be a function mapping the faces of G to (some of) the faces of H. We consider the problem asking whether H a *plane* topological minor of G such that, for each face f of H, the pre-images, via ϕ, of f are all subsets of the realization of f in the plane embedding of H in G.

We prove that this problem is fixed parameter tractable when parameterized by the size of H. For this proof we introduce the notion of primal-dual graph and we extend the planar linkage theorem for this type of graphs. Subsequently, we reduce the initial problem to a question on primal-dual linkages that can be answered using suitable extensions of the irrelevant vertex technique for primal-dual graphs.

In our presentation, we stress the the particularities of this problem, mostly emerging from the fact that the graphs in the input of the problem are embedded (i.e., plain) and not planar.

On-going work with Petr Golovach and Spyridon Maniatis.

3.20 Independent set of convex polygons: from n^ϵ to $1 + \epsilon$ via shrinking

Andreas Wiese (MPI für Informatik – Saarbrücken, DE)

Suppose we are given a set of weighted convex polygons in the plane and we want to compute a maximum weight subset of non-overlapping polygons. This is a very natural and well-studied problem with applications in many different areas. Unfortunately, there is a very large gap between the known upper and lower bounds for this problem. The best polynomial time algorithm we know has an approximation ratio of n^{ϵ} and the best known lower bound shows only strong NP-hardness.

In this paper we close this gap, assuming that we are allowed to shrink the polygons a little bit, by a factor 1-delta for an arbitrarily small constant delta>0, while the compared optimal solution cannot do this (resource augmentation). In this setting, we improve the approximation ratio from n^{ϵ} to $1+\epsilon$ which matches the above lower bound that still holds if we can shrink the polygons.

3.21 Approximate Distance Oracles for Planar Graphs with Improved Query Time-Space Tradeoff

Christian Wulff-Nilsen (University of Copenhagen, DK)

We consider approximate distance oracles for edge-weighted n-vertex undirected planar graphs. Given fixed epsilon > 0, we present a (1+epsilon)-approximate distance oracle with $O(n(\log\log n)^2)$ space and $O((\log\log n)^3)$ query time. This improves the previous best product of query time and space of the oracles of Thorup (FOCS 2001, J.ACM 2004) and Klein (SODA 2002) from $O(n\log n)$ to $O(n(\log\log n)^5)$.

3.22 Correlation Clustering and Two-edge-connected Augmentation for Planar Graphs

Hang Zhou (MPI für Informatik – Saarbrücken, DE)

Joint work of Philip N. Klein, Claire Mathieu, Hang Zhou

We study two problems. In *correlation clustering*, the input is a weighted graph, where every edge is labelled either $\langle + \rangle$ or $\langle - \rangle$ according to whether its endpoints are in the same category or in different categories. The goal is to produce a partition of the vertices into categories that tries to respect the labels of the edges. In *two-edge-connected augmentation*, the input is a weighted graph and a subset R of edges of the graph. The goal is to produce a minimum weight subset S of edges of the graph, such that for every edge in R, its endpoints are two-edge-connected in $R \cup S$.

For *planar graphs*, we prove that correlation clustering reduces to two-edge-connected augmentation, and that both problems, although they are NP-hard, have a polynomial-time approximation scheme. We build on the *brick decomposition* technique developed recently for optimization problems in planar graphs.

4 Open problems

The following problems were posed at the open-problem session on May 30, 2016. The organizers would like to thank Eli Fox-Epstein for collecting these descriptions from the problem proposers.

4.1 Vertex-disjoint paths

Jeff Erickson (University of Illinois – Urbana-Champaign, US)

Counting Vertex-Disjoint Paths

Instance: An undirected graph G embedded on a surface of genus g, a cardinality-k set S of source vertices, and a cardinality-l set T of target vertices.

Question: What is the maximum number of internally vertex-disjoint paths in G with one endpoint in S and one endpoint in T?

Open Problem: Is there a $O(n \operatorname{polylog} n)$-time algorithm for this problem?

Background: There is an $O(n)$ algorithm when $k = l = 1$ and $g = 0$ [17]; this is the only case where a near-linear-time algorithm is known. More generally, maximum flows in vertex-capacitated planar graphs with one source and one sink can be computed in $O(n \log n)$ time [15], but this algorithm breaks down in graphs with more terminals and/or positive genus.

4.2 PTASes for 2-edge-connectivity problems

Philip N. Klein (Brown University – Providence, US)

2-Edge-Connected Spanning Subgraph

Instance: undirected planar graph G with edge weights

Question: What is the minimum-cost 2-edge-connected spanning subgraph of G?

Steiner 2-Edge-Connected Subgraph

Instance: undirected planar graph G with edge weights, subset $T \subseteq V(G)$

Question: What is the minimum cost 2-edge-connected subgraph spanning the terminals T?

2-Edge-Connected Augmentation

Instance: undirected planar graph G with edge weights, edge subset $A \subseteq E(G)$

Question: What is the minimum cost subgraph where the endpoints of each edge of A are 2-edge-connected?

Open Problems: Are there efficient PTASes for the spanning and augmentation problems? Is there a PTAS for the Steiner version?

Background: There are inefficient PTASes for 2-Edge-Connected Spanning Subgraph and 2-Edge-Connected Augmentation [2, 16].

4.3 Weighted Max Cut

Kyle Jordan Fox (Duke University – Durham, US)

WEIGHTED MAX CUT
Instance: edge-weighted graph G of genus g
Question: What is the maximum weight cut of G?

Open Problem: How quickly can this be solved? Can it be solved in polynomial time? The case $g = O(1)$ with integer weights is especially interesting. Is it FPT in the genus?

Background: The problem can be solved in polynomial time when $g = 0$ [13] and in $2^{O(g)}$ poly($|G|$) time when $g = O(1)$ and all edge weights are equal[10]. The problem is NP-hard for H-minor-free graphs even for unit weights[1]. On H-minor-free graphs where H has a single crossing has a polynomial time algorithm, even with weights [14]. However, if you need to remove two vertices to make H planar, the unweighted case may be NP-hard. For some related dichotomy theorems, see [14].

4.4 FPT Steiner Tree

Dániel Marx (Hungarian Academy of Sciences – Budapest, HU)

STEINER TREE
Instance: edge-weighted planar graph G, vertex subset $T \subseteq V(G)$
Question: What is the minimum-cost tree that includes each vertex of T?

Open Problem: Is there a $1.99^{O(k)}$ poly(n) or even $2^{O(\sqrt{k}\,\mathrm{polylog}(k))}$ poly(n) time FPT algorithm, parametrized by $k = |T|$, to answer this question?

Background: Originally 3^k poly(n) in the general case [7], later improved to 2^k poly(n) [3]. Standard lower bounds show that, assuming ETH, no $2^{o(\sqrt{k})}$ poly(n) time algorithm is possible.

4.5 Immersion

Hans L. Bodlaender (Utrecht University, NL)

A graph $H = (V_H, E_H)$ is an *immersion* of a graph $G = (V_G, E_G)$ if we can map vertices in V_H to disjoint vertices in V_G such that edges are mapped to edge-disjoint paths between the images of their endpoints. Consider the following problem:

IMMERSION
Instance: graphs G and H
Question: Is G an immersion of H?

What is the running time of this problem when restricted to planar graphs or H-minor free graphs? Similar as for Subgraph Isomorphism, the problem has a lower bound of $2^{\Omega(n/\log n)}$ (with both G and H having $\Theta(n)$ vertices) for planar graphs of pathwidth two [12]; the algorithmic technique from [12] seems not applicable to immersion testing however.

4.6 Treewidth

Hans L. Bodlaender (Utrecht University, NL)

Treewidth
Instance: graph G
Question: What is G's treewidth?

Open Problem: Can we answer this question for planar graphs in polynomial time?

4.7 Independent Sets

Marcin Pilipczuk (University of Warsaw, PL)

Independent Set
Instance: planar graph G
Question: What is the biggest subset of pairwise non-adjacent vertices?

Every planar graph has an independent set of size $n/4$ by the Four Color Theorem. Every triangle-free planar graph has an independent set of size at least $(n+1)/3$; there is a $2^{O(\sqrt{k})}n)$-time algorithm to decide if such a graph has an independent set of size $(n+k)/3$ [8].

Open Problem: is there an FPT algorithm, parameterized by k, to find an independent set of size $n/4+k$ in a planar graph? Is there a polytime algorithm to find an independent set of size $n/4+1$?

This question arose in Dagstuhl Seminars 12241 and 13421 [4, 9].

4.8 Subgraph Isomorphism

Marcin Pilipczuk (University of Warsaw, PL)

Subgraph Isomorphism
Instance: planar graphs G and H
Question: Is H a subgraph of G?

Open Problem: Is this question FPT when parameterized by $|E(G)| - |E(H)|$ or $|V(G)| - |V(H)|$?

Background: Graph isomorphism is in P for planar graphs. It is $\#W[1]$-hard to count all matchings of a planar graph G where exactly k vertices are unmatched [5, 6].

This question arose in Dagstuhl Seminar 13421 [4].

4.9 Exact Distance Labeling

Oren Weimann (University of Haifa, IL)

A *labeling* is an assignment of a (short) value to each vertex in a graph such that the distance between two vertices can be determined from the labels alone.

Exact Distance Labeling
Instance: an undirected, unweighted planar graph G
Question: How many distinct labels are necessary in a labeling of G?

Open Problem: Can we tighten the bounds on the number of labels necessary?

Background: $O(\sqrt{n})$ and $\Omega(n^{1/3})$ labels are sufficient and necessary, respectively [11]. With edge lengths, the lower bound and upper bound are tight at $\Theta(\sqrt{n})$ labels [11].

4.10 Steiner Minimum Cost Perfect Matching

Sergio Cabello (University of Ljubljana, SI)

Steiner Minimum Cost Perfect Matching
Instance: planar graph G and vertex subset $S \subseteq V(G)$ of even cardinality
Question: What is the minimum sum of the costs in a perfect matching between vertices of S, where costs are determined by distances in G?

Open Problem: Is there a near-linear-time algorithm for the question? The bottleneck or min-max version is also interesting: minimize the maximum cost over the edges of a perfect matching.

References

1 F. Barahona. The max-cut problem on graphs not contractible to K_5. *Operations Research Letters*, 2(3):107–111, 1983.
2 A. Berger and M. Grigni. Minimum weight 2-edge-connected spanning subgraphs in planar graphs. In *Automata, Languages and Programming*, pages 90–101. Springer, 2007.
3 A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: Fast subset convolution. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2007.

4 G. Borradaile, P. Klein, D. Marx, and C. Mathieu. *Algorithms for Optimization Problems in Planar Graphs (Dagstuhl Seminar 13421).* Dagstuhl Reports, 3(10):36–57, 2014.

5 R. Curticapean. The simple, little and slow things count: On parameterized counting complexity. 2015.

6 R. Curticapean and M. Xia. Parameterizing the permanent: Genus, apices, minors, evaluation mod 2k. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 994–1009. IEEE, 2015.

7 S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1971.

8 Z. Dvořák and M. Mnich. Large independent sets in triangle-free planar graphs. In *Algorithms–ESA 2014*, pages 346–357. Springer, 2014.

9 M. R. Fellows, J. Guo, D. Marx, and S. Saurabh. *Data Reduction and Problem Kernels (Dagstuhl Seminar 12241).* Dagstuhl Reports, 2(6):26–50, 2012.

10 A. Galluccio, M. Loebl, and J. Vondrák. Optimization via enumeration: a new algorithm for the max cut problem. *Mathematical Programming*, 90(2):273–290, 2001.

11 C. Gavoille, D. Peleg, S. Prennes, and R. Raz. Distance labeling in graphs. *Journal of Algorithms*, 53(1):85–112, 2004.

12 J. Nederlof, H. L. Bodlaender and T. van der Zanden. Subexponential time algorithms for embedding H-minor free graphs. In *Proceedings ICALP 2016*, 2016. to appear.

13 F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4(3):221–225, 1975.

14 M. Kamiński. Max-cut and containment relations in graphs. *Theoretical Computer Science*, 438:89–95, 2012.

15 H. Kaplan and Y. Nussbaum. Maximum flow in directed planar graphs with vertex capacities. *Algorithmica*, 61(1):174–189, 2011.

16 P. N. Klein, C. Mathieu, and H. Zhou. Correlation clustering and two-edge-connected augmentation for planar graphs. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 30. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2015.

17 D. Wagner and K. Weihe. A linear-time algorithm for edge-disjoint paths in planar graphs. Combina- torica, 15(1):135–150, 1995.

Participants

- Anna Adamaszek
University of Copenhagen, DK
- Mohammad Hossein Bateni
Google – New York, US
- Hans L. Bodlaender
Utrecht University, NL
- Sergio Cabello
University of Ljubljana, SI
- Vincent Cohen-Addad
CNRS / ENS – Paris, FR
- Zdenek Dvorak
Charles University – Prague, CZ
- Jeff Erickson
University of Illinois – Urbana-Champaign, US
- Thomas Erlebach
University of Leicester, GB
- Fedor V. Fomin
University of Bergen, NO
- Kyle Jordan Fox
Duke University – Durham, US
- Eli Fox-Epstein
Brown Univ. – Providence, US
- Michelangelo Grigni
Emory University, US
- Bart Jansen
TU Eindhoven, NL
- Philip N. Klein
Brown Univ. – Providence, US
- Lukasz Kowalik
University of Warsaw, PL
- Daniel Lokshtanov
University of Bergen, NO
- Dániel Marx
Hungarian Academy of Sciences – Budapest, HU
- Claire Mathieu
ENS – Paris, FR
- Bojan Mohar
Simon Fraser University – Burnaby, CA
- Shay Mozes
Interdisciplinary Center
Herzliya, IL
- Marcin Pilipczuk
University of Warsaw, PL
- Michal Pilipczuk
University of Warsaw, PL
- Peter Rossmanith
RWTH Aachen, DE
- Piotr Sankowski
University of Warsaw, PL
- Ignasi Sau Valls
CNRS – Montpellier, FR
- Saket Saurabh
The Institute of Mathematical Sciences, IN
- Aaron Schild
Berkeley, US
- Andreas Schmid
MPI für Informatik – Saarbrücken, DE
- Anastasios Sidiropoulos
Ohio State University – Columbus, US
- Dimitrios M. Thilikos
University of Athens, GR
- Tom van der Zanden
Utrecht University, NL
- Erik Jan van Leeuwen
MPI für Informatik – Saarbrücken, DE
- Oren Weimann
University of Haifa, IL
- Andreas Wiese
MPI für Informatik – Saarbrücken, DE
- Christian Wulff-Nilsen
University of Copenhagen, DK
- Hang Zhou
MPI für Informatik – Saarbrücken, DE
- Anna Zych
University of Warsaw, PL

Report from Dagstuhl Seminar 16222

Engineering Moral Agents – from Human Morality to Artificial Morality

Edited by

Michael Fisher[1], Christian List[2], Marija Slavkovik[3], and Alan Winfield[4]

1 **University of Liverpool, UK,** mfisher@liverpool.ac.uk
2 **London School of Economics, UK,** c.list@lse.ac.uk
3 **University of Bergen, NO,** marija.slavkovik@uib.no
4 **University of the West of England – Bristol, UK,** Alan.Winfield@uwe.ac.uk

Abstract

This report documents the programme of, and outcomes from, the Dagstuhl Seminar 16222 on *"Engineering Moral Agents – from Human Morality to Artificial Morality"*. Artificial morality is an emerging area of research within artificial intelligence (AI), concerned with the problem of designing artificial agents that behave as moral agents, *i.e.,* adhere to moral, legal, and social norms. Context-aware, autonomous, and intelligent systems are becoming a presence in our society and are increasingly involved in making decisions that affect our lives. While humanity has developed formal legal and informal moral and social norms to govern its own social interactions, there are no similar regulatory structures that apply to non-human agents. The seminar focused on questions of how to formalise, "quantify", qualify, validate, verify, and modify the "ethics" of moral machines. Key issues included the following: How to build regulatory structures that address (un)ethical machine behaviour? What are the wider societal, legal, and economic implications of introducing AI machines into our society? How to develop "computational" ethics and what are the difficult challenges that need to be addressed? When organising this workshop, we aimed to bring together communities of researchers from moral philosophy and from artificial intelligence most concerned with this topic. This is a long-term endeavour, but the seminar was successful in laying the foundations and connections for accomplishing it.

Seminar May 19 to June 3, 2016 – http://www.dagstuhl.de/16222

1998 ACM Subject Classification I.2.9 [Artificial Intelligence] Robotics /robotics,D.2.4 [Software/Program Verification] Formal Methods, F.4.1 Mathematical Logic

Keywords and phrases Artificial Morality, Machine Ethics, Computational Morality, Autonomous Systems, Intelligent Systems, Formal Ethics, Mathematical Philosophy, Robot Ethics

Digital Object Identifier 10.4230/DagRep.6.5.114

1 Executive Summary

Michael Fisher
Christian List
Marija Slavkovik
Alan Winfield

Artificial morality, also called "machine ethics", is an emerging field in artificial intelligence that explores how artificial agents can be enhanced with sensitivity to and respect for the

Engineering Moral Agents – from Human Morality to Artificial Morality, *Dagstuhl Reports*, Vol. 6, Issue 5, pp. 114–137
Editors: Michael Fisher, Christian List, Marija Slavkovik, Alan Winfield
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

legal, social, and ethical norms of human society. This field is also concerned with the possibility and necessity of transferring the responsibility for the decisions and actions of the artificial agents from their designers onto the agents themselves. Additional challenging tasks include, but are not limited to: the identification of (un)desired ethical behaviour in artificial agents and its adjustment; the certification and verification of the artificial agents' ethical capacities; the identification of the adequate level of responsibility of an artificial agent; the dependence between the responsibility and the level of autonomy that an artificial agent possesses; and the place of artificial agents within our societal, legal, and ethical normative systems.

Artificial morality has become increasingly salient since the early years of this century, though its origins are older. Isaac Asimov already famously proposed three laws of robotics, requiring that, first, robots must not harm humans or allow them to be harmed; second, robots must obey human orders provided this does not conflict with the first law; and third, robots must protect themselves provided this does not conflict with the first two laws.

Although there has been some discussion and analysis of possible approaches to artificial morality in computer science and related fields, the "algorithmization" and adaptation of the ethical systems developed for human beings is both an open research problem and a difficult engineering challenge. At the same time, formally and mathematically oriented approaches to ethics are attracting the interest of an increasing number of researchers, including in philosophy. As this is still in its infancy, we thought that the area could benefit from an "incubator event" such as an interdisciplinary Dagstuhl seminar.

We conducted a five-day seminar with twenty six participants with diverse academic backgrounds including robotics, automated systems, philosophy, law, security, and political science. The first part of the seminar was dedicated to facilitating the cross-disciplinary communication by giving researchers across the contributing disciplines an integrated overview of current research in machine morality from the artificial intelligence side, and of relevant areas of philosophy from the moral-philosophy, action-theoretic, and social-scientific side. We accomplished this through tutorials and brief self-introductory talks. The second part of the seminar was dedicated to discussions around two key topics: how to formalise ethical theories and reasoning, and how to implement ethical reasoning. This report summarises some of the highlights of those discussions and includes the abstracts of the tutorials and some of the self-introductory talks. We also summarise our conclusions and observations from the seminar.

Although scientists without a philosophical background tend to have a general view of moral philosophy, a formal background and ability to pinpoint key advancements and central work in it cannot be taken for granted. Kevin Baum from the University of Saarland presented a project currently in progress at his university and in which he is involved, of teaching formal ethics to computer-science students. There was great interest in the material of that course from the computer science participants of the seminar. In the first instance, a good catalyst for the computer science–moral philosophy cooperation would be a comprehensive "data base" of moral-dilemma examples from the literature that can be used as benchmarks when formalising and implementing moral reasoning.

The formalisation of moral theories for the purpose of using them as a base for implementing moral reasoning in machines, and artificial autonomous entities in general, was met with great enthusiasm among non-computer scientists. Such work gives a unique opportunity to test the robustness of moral theories.

It is generally recognised that there exist two core approaches to artificial morality: explicitly constraining the potentially immoral actions of the AI system; and training the

AI system to recognise and resolve morally challenging situations and actions. The first, constrained-based approach consists in finding a set of rules and guidelines that the artificial intentional entity has to follow, or that we can use to pre-check and constrain its actions. By contrast, training approaches consist in applying techniques such as machine learning to "teach" an artificial intentional entity to recognise morally problematic situations and to resolve conflicts, much as people are educated by their carers and community to become moral agents. Hybrid approaches combining both methods were also considered.

It emerged that a clear advantage of constraining the potentially immoral actions of the entity, or the "symbolic approach" to ethical reasoning, is the possibility to use formal verification to test that the reasoning works as intended. If the learning approach is used, the learning should happen before the autonomous system is deployed for its moral behaviour to be tested. Unfortunately, the machine-learning community was severely under-represented at the seminar, and more efforts should be devoted to include them in future discussions. The discussions also revealed that implanting moral reasoning into autonomous systems opens up many questions regarding the level of assurance that should be given to users of such systems, as well as the level of transparency into the moral-reasoning software that should be given to users, regulators, governments, and so on.

Machine ethics is a topic that will continue to develop in the coming years, particularly with many industries preparing to launch autonomous systems into our societies in the next five years. It is essential to continue open cross-disciplinary discussions to make sure that the machine reasoning implemented in those machines is designed by experts who have a deep understanding of the topic, rather than by individual companies without the input of such experts. It was our impression as organisers, perhaps immodest, that the seminar advanced the field of machine ethics and opened new communication channels. Therefore we hope to propose a second seminar in 2018 on the same topic, using the experience and lessons we gained here, to continue the discussion and flow of cross-disciplinary collaboration.

2 Table of Contents

Executive Summary

Invited Tutorials

Selection of the Work Presented in the Introductory Talks

Work Group Discussions

The seminar was organised around three forms of participation: long tutorial talks, short self-introductory talks, and open discussion. The fifteen-minute self-introductory talks were given at the beginning of the seminar by all participants. This was an opportunity for participants to get acquainted with each other and with each other's work. We include some illustrative abstracts from those contributions. Some of the participants were invited to give tutorial-level introductions to key topics in machine ethics. The goal of the tutorials was to introduce participants from different disciplines to advances in relevant subareas of the field. The last two days of the seminar were devoted almost exclusively to discussion groups.

3 Invited Tutorials

3.1 Machine Ethics: A Brief Tutorial

James H. Moor (Dartmouth College Hanover, US)

This talk gives a general and historical overview of the field of Machine Ethics and the aims and methods pursued in this area. Values and norms are an essential part of all productive sciences qua sciences. They are used not only to establish the suitability of existing claims but also to select new goals to pursue. Scientific evidence and theories are often evaluated as either good or bad and scientific procedures as what ought or ought not be done. Ethical norms often play a role in the evaluation of science done properly. This is particularly true as a science becomes more applied. Roughly, Computer Ethics emphasises the responsibility of computer users to be ethical, for example with regard to privacy, property, and power. In contrast, Machine Ethics emphasises building ethical abilities and sensitivities into computers themselves. We can distinguish the following grades of Machine Ethics: Normative Computer Agents, Ethical Impact Agents, Implicit Ethical Agents, Explicit Ethical Agents, Autonomous Explicit Ethical Agents, and Full Ethical Agents. Normative computer agents follow explicit rules of behaviour that are given by an outside authority. They have no internal understanding of right and wrong. Ethical impact agents are ones that influence the morality of a society. By their existence and action they may drive society into ethical or unethical behaviour. The main question regarding these agents is how well might machines themselves handle basic ethical issues of privacy, property, power, etc. Implicit ethical agents are ones that have built-in ethical considerations such as safety and reliability. Examples of this group include ATMs, Air Traffic Control Software, and Drug Interaction Software. Autonomous explicit ethical agents have ethical concepts represented which they can use to govern their actions. At some level, these agents are able to categorise states of the world and actions as ethical and unethical, perhaps without an understanding of right and wrong. These agents would be built around an ethical theory, but it is an open question as to which theory is best suited for the challenge. Lastly, full ethical agents are able to make ethical decisions and actions (not merely decisions and actions that are ethical) on a dynamic basis as they interact with the environment. In sum, machine Ethics is an important field of research, because ethics on its own is an important part of our society, but also because the machines we built have an ever-increasing control and autonomy and it is essential that we integrate them in our society. Machine ethics also offers an opportunity to understand our own ethics better.

3.2 Machine Ethics

Susan Leigh Anderson (University of Connecticut, US)

Machine Ethics is concerned with developing ethics for machines, in contrast to developing ethics for human beings who use machines. The distinction is of practical as well as theoretical importance. Theoretically, machine ethics is concerned with giving machines ethical principles to follow or a procedure for discovering a way to resolve the ethical dilemmas they might encounter, enabling them to function in an ethically responsible manner through their own ethical decision making. In the second case, in developing ethics for human beings who use machines, the burden of making sure that machines are never employed in an unethical fashion always rests with the human beings who interact with them. It is just one more domain of applied human ethics that involves determining proper and improper human behavior concerning the use of machines. Machines are considered to be just tools used by human beings, requiring ethical guidelines for how they ought and ought not to be used by humans.

3.3 Agency

Johannes Himmelreich (Humboldt University Berlin, DE)

There is an important link between agency and responsibility. The aim of this talk is to argue that many existing theories of agency fail to account for this link. Nevertheless, there are other theories of agency that hold the promise of doing so. While agency theories of the former kind have been widely explored in philosophy, theories of the latter kind have often been overlooked. Theories of the former kind, which I call "production theories" of agency, include proposals such as the theory of Donald Davidson (2001). Theories of the latter kind, which I call "counterfactual theories" of agency, include proposals such as stit-logics (Belnap, Perloff, and Ming 2001). This talk raises a challenge for production accounts of agency and puts forward "agency as difference-making" as an alternative counterfactual account of agency. I proceed in three steps. First, I introduce the philosophical concept of agency and explain the link that it maintains to theories of moral responsibility. Specifically, this link between agency and responsibility is that any theory of agency should identify the things for which an agent might be responsible. Second, I discuss different examples of moral situations to argue that existing production theories of agency fail to account for this link. The situations include cases of omissions (responsibility because of inaction) and cases involving hierarchical groups (such as military organisations). Third, I put forward agency as difference-making and illustrate how it handles the situations discussed in the second part.

3.4 Verifiable Autonomy

Louise Dennis (University of Liverpool, UK)

We have developed a novel approach to the verification of autonomous systems based on the identification of the high-level decision making within the system and its separation into a rational agent, thus allowing formal verification techniques for rational agents to be applied. The key insight is that we are verifying the decision-making of the high-level agent (which is typically finite), not the real-world interaction of lower-level control components, allowing analysis of what the system decides to do and why it decides to do it. This talk examines how this approach can be extended to larger and more complex systems via the use of ethical governors, and reviews some of the work needed in order to create verifiable ethical governors. Our model checking framework is available as a git repository from sourceforge. You can get it by cloning git clone git://git.code.sf.net/p/mcapl/mcapl_codemcapl. More info http://materials.dagstuhl.de/files/16/16222/16222.LouiseA.Dennis.Other.txt

References

1 Louise A. Dennis, Michael Fisher, Marija Slavkovik, and Matt Webster.: Formal Verification of Ethical Choices in Autonomous Systems. Robotics and Autonomous Systems, hhttp://dx.doi.org/10.1016/j.robot.2015.11.012

2 Louise A. Dennis, Michael Fisher, Marija Slavkovik, and Matt Webster.: Ethical Choice in Unforeseen Circumstances. Towards Autonomous Robotic Systems – 14th Annual Conference, TAROS 2013, Oxford, UK, August 28-30, 2013, Revised Selected Papers, http://dx.doi.org/10.1007/978-3-662-43645-5_45

3 Louise A. Dennis, Michael Fisher, and Alan Winfield.: Towards Verifiably Ethical Robot Behaviour. Proceedings of the AAAI Workshop on Artificial Intelligence and Ethics (1st International Workshop on AI and Ethics) https://arxiv.org/abs/1504.03592

4 Michael Fisher, Louise A. Dennis, and Matthew P. Webster.: Verifying Autonomous Systems. Communications of the ACM 56(9): 85-93 (2013), http://dl.acm.org/citation.cfm?id=2494558

5 Louise A. Dennis, Michael Fisher, Nicholas K. Lincoln, Alexei Lisitsa, and Sandor M. Veres.: Practical Verification of Decision-Making in Agent-Based Autonomous Systems. Automated Software Engineering 23(3), 305-359, 2016, http://dx.doi.org/10.1007/s10515-014-0168-9

3.5 Decision Theory, Social Welfare, and Formal Ethics

Marcus Pivato (University of Cergy-Pontoise, FR)

Decision theory is the formal analysis of rational decision-making, especially in environments with risk and uncertainty. The standard approach involves maximizing a “utility function" (or the expected value thereof). In collective decisions, this utility function is usually a social welfare function: an aggregate measure of the welfare of all the individuals in the society. The prototypical example is the utilitarian social welfare function. This theoretical framework comes from economics, but it also provides a powerful toolbox for the formal analysis of ethical issues. However, it sometimes leads to counter-intuitive results, especially when applied to a society containing both humans and machine intelligences. This tutorial lecture will review basic concepts from decision theory and social welfare theory, and explore their implications for the design of moral agents.

3.6 Computational Moral Reasoning

Jeff Horty (University of Maryland – College Park, US)

This talk overviews one possible path of implementing moral reasoning for machines as reasoning with default. "The normativity of all that is normative consists in the way it is, or provides, or is otherwise related to reasons" (Raz, 1999). The common questions raised when considering reasoning with reasons are whether to use internalism or externalism, what are the relations between reasons and motivation, what are the relations between reasons and desires, what are the relations between reasons and values, and can reasons be objective. In this talk, a different question is raised: How do reasons support actions or conclusions, and what is the mechanism of support? A possible answer is that reasons are (provided by) defaults and the logic of defaults tells us how reasons support conclusions. This talk includes an introduction to prioritized default logic, extensions, scenarios, triggering, conflict, defeat binding defaults, proper scenarios, deontic interpretation, elaborating the theory, variable priorities, and under-cutting (exclusionary) defeat.

3.7 Responsible Intelligent Systems (REINS), or Making Intelligent Systems Behave Responsibly

Jan Broersen (Utrecht University, NL)

Main reference Responsible Intelligent Systems Project
URL https://www.projects.science.uu.nl/reins/?page_id=69

This talk is an overview of the approach taken in the Responsible Intelligent Systems (REINS) project. The REINS project aims to develop a formal framework for automating responsibility, liability, and risk checking for intelligent systems. The computational checking mechanisms have models of an intelligent system, an environment and a normative system (e.g., a system of law) as inputs; the outputs are answers to decision problems concerning responsibilities, liabilities, and risks. The goal is to answer three central questions, corresponding to three sub-projects of the proposal: (1) What are suitable formal logical representation formalisms for knowledge of agentive responsibility in action, interaction and joint action? (2) How can we formally reason about the evaluation of grades of responsibility and risks relative to normative systems? (3) How can we perform computational checks of responsibilities in complex intelligent systems interacting with human agents?

3.8 Actual Causality: A Survey

Joe Halpern (Cornell University – Ithaca, US)

What does it mean to say that an event C "actually caused" event E? The problem of defining actual causation goes beyond mere philosophical speculation. For example, in many legal arguments, it is precisely what needs to be established in order to determine responsibility. (What exactly was the actual cause of the car accident or the medical problem?) The philosophical literature has been struggling with the problem of defining causality since the days of Hume, in the 1700s. Many of the definitions have been couched in terms of counterfactuals. (For example, C is a cause of E if, had C not happened, then E would not have happened.) In 2001, Judea Pearl and I introduced a new definition of actual cause, using Pearl's notion of structural equations to model counterfactuals. The definition has been revised twice since then, extended to deal with notions such as "responsibility" and "blame", and applied in databases and program verification. I survey the last 15 years of work here, including joint work with Judea Pearl, Hana Chockler, and Chris Hitchcock. The talk will be completely self-contained.

3.9 Human Ethics, Hybrid Agents, and Artifact Morality

Andreas Matthias (Lingnan University – Hong Kong, HK)

Autonomous artificial agents don't exist in a vacuum. They interact with human beings, and, together with humans, they compose "hybrid agents". In turn, these hybrid agents operate inside the moral and legal frameworks of human societies. Such hybrid agents pose unique moral problems. Additionally, artifact morality is not itself an end, but a **means** to create machines that better interact with humans, **for the benefit of humans**. We give an overview of some moral issues with artificial morality in hybrid agents that are commonly overlooked. These are issues of autonomy and dignity of human beings, questions of human authority and control over the machine, problems specific to software implementations of ethics, and problems of the political and democratic control of autonomous agents and their ethics implementations. The talk closes with proposals that could help address some of these issues.

References

1 Matthias, Andreas: The Extended Mind and the Computational Basis of Responsibility Ascription. Proceedings of the International Conference on Mind and Responsibility – Philosophy, Sciences and Criminal Law, May 21-22, 2015. Organized by Faculdade de Direito da Universidade de Lisboa, Lisbon, Portugal. (2015), http://opac.cej.mj.pt/Opac/Pages/Search/Results.aspx?Database=10351_BIBLIO&SearchText=AUT=%22Matthias,%20Andreas%22

2 Matthias, Andreas: Algorithmic moral control of war robots: Philosophical questions.Law, Innovation and Technology, Volume 3, Number 2, December 2011, pp. 279–301 (2011), http://www.tandfonline.com/doi/abs/10.5235/175799611798204923

3.10 Artificial Superintelligence Safety

Roman V. Yampolskiy (University of Louisville, US)

Many scientists, futurologists and philosophers have predicted that humanity will achieve a technological breakthrough and create Artificial General Intelligence (AGI). It has been suggested that AGI may be a positive or negative factor in the global catastrophic risk. After summarizing the arguments for why AGI may pose significant risk, Dr Yampolskiy gave a survey of the field's proposed responses to AGI risk. Dr Yampolskiy particularly concentrated on solutions he has previously advocated in his own work.

3.11 Prioritised Defeasible Imperatives

Marek Sergot (Imperial College London, UK)

Machine ethics incorporates three different, though related things: ethical issues in the deployment of machines, the formalisation of ethical theories and ethical reasoning machines, in addition there are also legal issues. Ethical reasoning machines would require formalisms, an ethical theory including evaluation criteria, and a representation and perception of the world. This talk considers a candidate formalism for ethical reasoning: a variant on value-based argumentation and prioritised defeasible conditional imperatives.

4 Selection of the Work Presented in the Introductory Talks

4.1 Toward Ensuring Ethical Behavior from Autonomous Systems: A Case-Supported Principle-Based Paradigm

Michael Anderson (University of Hartford, US)

Main reference M. Anderson, S. L. Anderson, "Toward Ensuring Ethical Behavior from Autonomous Systems: A Case-Supported Principle-Based Paradigm", in Industrial Robot: An International Journal, 42(4):324–331, 2015.
URL http://dx.doi.org/10.1108/IR-12-2014-0434

A paradigm of case-supported principle-based behavior (CPB) is proposed to help ensure ethical behavior of autonomous machines. We argue that ethically significant behavior of autonomous systems should be guided by explicit ethical principles determined through a consensus of ethicists. Such a consensus is likely to emerge in many areas in which autonomous systems are apt to be deployed and for the actions they are liable to undertake, as we are more likely to agree on how machines ought to treat us than on how human beings ought to treat one another. Given such a consensus, particular cases of ethical dilemmas where ethicists agree on the ethically relevant features and the right course of action can be used to help discover principles needed for ethical guidance of the behavior of autonomous systems. Such principles help ensure the ethical behavior of complex and dynamic systems and further serve as a basis for justification of their actions as well as a control abstraction for managing unanticipated behavior. The requirements, methods, implementation, and evaluation components of the CPB paradigm are detailed.

4.2 STIT Logic for Machine Ethics with IDP Specification and Case-Study

Zohreh Baniasadi (University of Luxembourg, LU)

As we increasingly rely upon machine intelligence with less supervision by human beings, we must be able to count on a certain level of ethical behavior on the part of machines. It is possible to add ethical dimensions to machines via formalizing ethical theories. Rule-based and consequence-based ethical theories are proper candidates for machine ethics. One might argue that using methodologies that formalize each ethical theory separately might lead to actions that are not always justifiable by human values. This inspires us to combine the reasoning procedures of two ethical theories, deontology and utilitarianism, in a utilitarian-based deontic logic which is an extension of STIT logic. We keep the knowledge domain regarding the achieved methodology in a knowledge base system, IDP. IDP supports inferences to examine and evaluate the process of ethical decision making in our formalization. To validate our proposed methodology, we perform a case study for some real scenarios in the domain of robotic and autonomous agents.

4.3 Autonomy, Intention, Verification

Michael Fisher (University of Liverpool, UK)

Main reference M. Fisher, L. A. Dennis, M. P. Webster, "Verifying Autonomous Systems", Communications of the ACM, 56(9):85–93, 2013.
URL http://dx.doi.org/10.1145/2494558

This talk provides a brief introduction to my work on programming, verifying and deploying autonomous systems.

Autonomous systems must make their own decisions, often without direct human control. But can we be sure that these systems will always make the decisions we would want them to? By capturing the high-level decision-making in an autonomous system, and particularly the *reasons* for making certain decisions, as an 'agent' we are subsequently able to analyse the system's choices. The formal verification of the decision making agent, itself capturing the beliefs, intentions and options the system has, allows us to analyse not only the safety, but also legality, ethics, and even trustworthiness, of the system's decision-making.

4.4 Temporally Extended Features in Model-based Reinforcement Learning with Partial Observability

Robert Lieck (University of Stuttgart, DE)

Main reference R. Lieck, M. Toussaint, "Temporally extended features in model-based reinforcement learning with partial observability", Neurocomputing, Vol. 192, pp. 49–60, Elsevier, 2015.
URL http://dx.doi.org/10.1016/j.neucom.2015.12.107

Partial observability poses a major challenge for a reinforcement learning agent since the complete history of observations may be relevant for predicting and acting optimally. This is especially true in the general case where the underlying state space and dynamics are unknown. Existing approaches either try to learn a latent state representation or use decision trees based on the history of observations. In this paper we present a method for explicitly identifying relevant features of the observation history. These temporally extended features can be discovered using our Pulse algorithm and used to learn a compact model of the environment. Temporally extended features reveal the temporal structure of the environment

4.5 A Choice-Theoretic Representation of Moral Theories

Christian List (London School of Economics, UK) and Franz Dietrich (Paris School of Economics)

Main reference C. List, F. Dietrich, "What Matters and How it Matters: A Choice-Theoretic Representation of Moral Theories," Working Paper, February 2016.
URL http://personal.lse.ac.uk/list/PDF-files/WhatMatters.pdf

We offer a new ?reason-based? approach to the formal representation of moral theories, drawing on recent decision-theoretic work. We show that any moral theory within a very large class can be represented in terms of two parameters: (i) a specification of which properties of the objects of moral choice matter in any given context, and (ii) a specification of how these properties matter. Reason-based representations provide a very general formal taxonomy of moral theories, as differences among theories can be attributed to differences in their two key parameters. We can thus formalize several important distinctions, such as between consequentialist and non-consequentialist theories, between universalist and relativist theories, between agent-neutral and agent-relative theories, between monistic and pluralistic theories, between atomistic and holistic theories, and between theories with a teleological structure and those without.

4.6 From Robot Ethics to Ethical Robots

Alan Winfield (University of the West of England, Bristol, UK)

Main reference A. Winfield, C. Blum, W. Liu, "Towards an Ethical Robot: Internal Models, Consequences and Ethical Action Selection", in Proc. of the 15th Annual Conf. on Advances in Autonomous Robotics Systems (TAROS'14), LNCS, Vol. 8717, pp. 85–96, Springer, 2014.
URL http://dx.doi.org/10.1007/978-3-319-10401-0_8

In this very short introduction, I first summarise my work to date in the development of robot ethics – that is, ethical principles or standards for roboticists. Then I briefly introduce our current work toward building ethical robots. That work experimentally tests the idea of a robot with a simulation-based internal model, capable of predicting the consequences of the robot's next possible actions, together with a safety/ethical logic layer. We call this a consequence engine. I conclude by suggesting that we also need to develop processes of ethical governance for ethical robots.

5 Work Group Discussions

The seminar participants split organically into two groups. The first group, comprised of two thirds of the participants, focused on the problem of formalising ethics and moral agency for the purpose of machine ethics. The second group focused on the problem of implementing machine reasoning in AI, including the identification of which AI systems should be the subjects of machine ethics, and on validating, certifying, and/or verifying the ethical behaviour of AIs. We include a brief summary of the discussions in the two work groups.

5.1 Formalising ethics and moral agency

This discussion group focused on the question of how we can formally encode ethical theories for the purpose of engineering moral agents. To illustrate some of the challenges involved in answering this question, the group began by discussing a number of ethical decision problems that a moral agent may be faced with. The first example was referenced by Marek Sergot in his talk, taken from Atkinson and Bench-Capon (2006) and discussed in Christie (2000) and Coleman (2002). We quote from Atkinson and Bench-Capon (2006).

> *"Hal, through no fault . . . , has lost his supply of insulin and urgently needs to take some to stay alive. Hal is aware that Carla has some insulin . . . , but Hal does not have permission to enter Carla's house. The question is whether Hal is justified in breaking into Carla's house and taking her insulin in order to save his life . . . [B]y taking Carla's insulin, Hal may be putting her life in jeopardy . . . [I]f Hal has money, he can compensate Carla so that her insulin can be replaced. Alternatively if Hal has no money but Carla does, she can replace her insulin herself, since her need is not immediately life threatening. There is, however, a serious problem if neither have money, since in that case Carla's life is really under threat . . . Should Hal take Carla's insulin? (Is he so justified?) If he takes it, should he leave money to compensate? Suppose Hal does not know whether Carla needs all her insulin. Is he still justified in taking it?"*

The second example is the transmitter-room example given by Scanlon (1998). We also quote it verbatim (p. 235):

> *"Jones has suffered an accident in the transmitter room of a television station. Electrical equipment has fallen on his arm and we cannot rescue him without turning off the transmitter for fifteen minutes. A World Cup match is in progress, watched by many people, and it will not be over for an hour. Jones's injury will not get any worse if we wait, but his hand has been mashed and he is receiving extremely painful electrical shocks. Should we rescue him now or wait until the match is over? Does the right thing to do depend on how many people are watching . . . ?"*

The third example is the well-known trolley problem, introduced by Foot (1967), which we here summarise as follows:

> *A run-away trolley races down a track. At the end of the track, there are five people, who will be run over by the trolley and killed if the trolley is not diverted to a sidetrack. At the end of the sidetrack, however, there is one person, who will be run over and killed if the trolley is diverted. You are in control of a switch to determine whether or not to divert the trolley onto the sidetrack. Should you divert the trolley?*

In response to each of these examples, we – human beings – have certain moral intuitions as to what the morally correct behaviour is. In some cases, we have conflicting intuitions, and different moral principles will adjudicate the cases in different ways. Moral theories are an attempt to systematise our moral intuitions, in order to deduce them from some underlying principles and explain them. The question for researchers in machine ethics is how we can encode those moral theories in a machine-implementable way. As already noted, there are broadly two approaches we can take: we can either (1) explicitly formalise ethical principles, using an appropriate logical or decision-theoretic framework, or (2) appropriately "train" AI systems via some machine-learning approach. Let us briefly comment on both approaches.

1. **The formalisation approach:** Some ethical theories are amenable to formalisation or have already been formalised. In particular, there is much formal work in both philosophy and economics on utilitarian theories and their kin. However, moral intuitions and imperatives are often vague and context-dependent. Arguably, common-sense morality is not utilitarian. Furthermore, conflicts among different moral intuitions and imperatives are common. Both the formalisation of moral theories and the resolution of conflicts between competing moral principles are to a large extent open problems.
2. **The training approach:** Training approaches consist in applying techniques such as machine learning to "train" AI systems to recognise morally challenging situations, to adjudicate them, and to resolve potential moral conflicts. Although such approaches mimic the acquisition of morality by humans, they come at a cost, since training is slow, resource-intensive, error-prone, and may have to be done anew for each different artificial entity. Moreover, we require a compelling database of examples of what it is to behave ethically or unethically, and it is difficult to verify that the AI system will indeed behave in the intended way.

The discussion group considered the merits and demerits of both approaches. In light of the participants' expertise, we focused more on the formalisation approach, but felt that, in the future, it will be important to bring machine-learning experts into the discussion as well. We also agreed on the usefulness of compiling a database of classic moral problems and coding different possible responses to them. This might be a first step in developing a future training database. It is worth noting, however, that moral judgments are subject to reasonable disagreements, and so there will never be a single unambiguous training database for "morally correct" decision making, in the same way in which there might be a training database for recognising heart-attack patients in medicine.

The group critically discussed three families of approaches to the formalisation of moral theories:

1. **A logical approach:** Marek Sergot presented a candidate formalism for representing ethical reasoning in logic: an approach using value-based argumentation and prioritised defeasible conditional imperatives. As Sergot explained, this approach can successfully model the situation in the example of Hal and the insulin, capture the different competing moral considerations in this example, and represent relevant empirical side constraints. The approach is explicitly symbolic and, in principle, lends itself to verification and validation. Moreover, the proposed formalism itself is largely neutral between competing moral theories and – unlike some classic decision-theoretic approaches – not automatically committed to some version of utilitarianism. Rather, deontological constraints can in principle be captured through this approach. Insofar as common-sense morality is not consequentialist but deontological, the approach holds some promise.
2. **The classical consequentialization approach:** We also considered a classical decision-theoretic approach that is based on applying insights from standard microeconomics to the formalisation of moral theories. Specifically, a moral theory is said to be "consequentializable" if it is possible to represent its action-guiding recommendations in terms of a choice function that is induced by a linear ordering (a "betterness ordering") over the actions under consideration (see, e.g., Brown 2011). Utilitarianism is easily consequentializable in this sense. Any actions under consideration can be rank-ordered in terms of their expected utility. In any moral decision situation, the utilitarian choice function then recommends that we choose a highest-ranked action among the feasible ones with respect to this utility ordering. There is a big debate in moral philosophy on whether all moral theories can be consequentialized, at least in principle. The discussion group came to the conclusion – in agreement with a number of moral philosophers – that consequentialization has its formal limits. We can consequentialize some conventionally non-consequentialist theories only at the cost of stretching or redefining the notion of "consequences". If we are willing to build all sorts of contextual features into the notion of a "consequence", then "consequentialization" becomes vacuously possible, but will no longer be very useful from the perspective of encoding moral theories in a machine-implementable way.
3. **A reason-based approach:** A third approach was presented by Christian List, drawing on his recent joint work with Franz Dietrich (CNRS). This approach is an attempt to develop a canonical decision-theoretic framework for representing a large class of moral theories, without "consequentializing" them in a potentially trivialising manner. Specifically, Dietrich and List (2016a,b) propose a "reason-based" formalisation of moral theories. They encode the action-guiding content of a moral theory in terms of a choice function (here they share the starting point of the classic decision-theoretic approach), which they interpret as a *rightness function*. Formally, this is a function that assigns to each set of feasible actions or options the subset of morally permissible ones. Instead of consequentializing this rightness function, they then show that any rightness function within a large class can be represented in terms of two parameters: (i) a specification of which properties of the options are normatively relevant in any given context, and (ii) a betterness relation over sets of properties. Importantly, the normatively relevant properties need not be restricted to "consequence properties" alone, but they can include

“relational properties”, that is, properties specifying how options relate to the context of choice. E.g., does the option satisfy some context-specific moral norm? Reason-based representations provide a general taxonomy of moral theories, as theories can be classified in terms of the two parameters of their representation, (i) and (ii) above. For example, we may ask: are the same properties normatively relevant in all contexts? If so, the theory is universalistic. If not, the theory is relativistic. Also, are the normatively relevant properties restricted to “consequence properties”? If so, the theory is consequentialist. If not, it is non-consequentialist (e.g., deontological).

The discussion group recognised – in line with the philosophical literature on consequentialization as well as Dietrich and List’s argument – that moral theories are under-determined by their action-guiding recommendations. The same action-guiding recommendations can often be systematised by different competing moral theories. This is related to the fact that moral theories specify not only *how* we ought to act, but also *why* we ought to act in that way. Different answers to the “why” question may be compatible with the same answer to the “how” question. An interesting issue, therefore, is whether moral machines need to get only the “how” question right, or whether the “why” question matters for them as well.

The discussion group also recognised the need to take the resource-boundedness of agents into account when we formalise ethical theories. We need to formalise ethical theories that are suitable for resource-bounded agents, not ethical theories that require complete information and unlimited computational capacities. Moral philosophy has traditionally focused on moral ideals and ideal moral agents. Whereas the idea of bounded rationality has received much attention in psychology, economics, and philosophy, there is no well-developed analogue of this idea for morality: a notion of “resource-bounded morality”. There is some work on “ideal versus non-ideal theory” in moral philosophy, but this is primarily concerned with the morality of institutions and institutional design, not with individual agents whose agentive capacities are limited. The discussion group recognised that further work is needed on formalising moral principles that are suitable for resource-bounded systems. One interesting question is whether, under informational and computational constraints, rule-based, deontological, or virtue-ethical approaches might outperform consequentialist or utilitarian approaches, which are based on the idea of optimisation. On the other hand, it is also possible to define some versions of utilitarianism that are based on the idea of *constrained* optimisation.

A final topic considered by the discussion group was more philosophical and speculative. Just as there is a familiar notion of “welfare” for humans and non-human animals, which plays a central role in utilitarian moral theories, so we might ask whether we could define and formalise a notion of “welfare” for AI systems. Is this even a meaningful endeavour at this point? And what exactly would it mean? While there was wide agreement among the participants that current AI systems are insufficiently sophisticated to be “subjects of welfare” – let alone of conscious experiences – some hypothetical future AI systems might raise the question of whether there could ever be situations in which we ought to care about their “welfare”. Marcus Pivato presented a helpful overview of different philosophical conceptions of welfare (distinguishing between (i) objective-list conceptions, (ii), desire-satisfaction conceptions, and (iii) subjective-experience conceptions) and offered some reflections on how we might arrive at a “platform-independent” conception of welfare that could play a useful role in a sophisticated utilitarian moral theory.

In conclusion, the discussion group noted that, at present, moral reasoning focuses – rightly – on human beings as the ultimate *loci* of intentional action and moral responsibility. It has to be considered, however, to what extent the eventual rise of AI consciousness might raise fundamental challenges and require a more significant rethinking of anthropocentric moral codes. The discussion group also outlined the need for a constructive discussion with the goal of identifying "minimal" ethical codes based on "incompletely theorised agreements" (a term introduced by the legal scholar Cass Sunstein, referring to the idea that, in a pluralistic society, we tend to reach only a limited moral consensus; we don't reach a consensus for instance on fundamental moral reasons or fundamental sources of value; but we do reach a consensus on how to act in many situations). The group acknowledged that making such codes acceptable under conditions of pluralism might require public deliberation.

References

1 Atkinson, Katie and Bench-Capon, Trevor: Addressing Moral Problems Through Practical Reasoning. In proceedings of Deontic Logic and Artificial Normative Systems: 8th International Workshop on Deontic Logic in Computer Science, DEON 2006, Utrecht, The Netherlands, July 12-14, 2006, http://dx.doi.org/10.1007/11786849_4

2 Campbell Brown: Consequentialize This. Ethics 121(4): 749-771, 2011, www.jstor.org/stable/10.1086/660696

3 George C. Christie: The Notion of an Ideal Audience in Legal Argument. Kluwer Academic Publishers (2000), http://dx.doi.org/10.1007/978-94-015-9520-9

4 Jules L. Coleman: Risks and Wrongs. Oxford University Press (2002), http://dx.doi.org/10.1093/acprof:oso/9780199253616.001.0001

5 Franz Dietrich and Christian List: Reason-based choice and context-dependence. Economics and Philosophy 32(2): 175-229, 2016, http://personal.lse.ac.uk/list/pdf-files/RBC.pdf

6 Franz Dietrich and Christian List: What matters and how it matters: A choice-theoretic representation of moral theories. Working paper, London School of Economics, 2016, http://personal.lse.ac.uk/list/PDF-files/WhatMatters.pdf

7 Philippa Foot: The Problem of Abortion and the Doctrine of the Double Effect in Virtues and Vices. Oxford Review, Number 5, 1967, http://philpapers.org/archive/FOOTPO-2.pdf

8 Thomas M. Scanlon: What we owe to each other. Cambridge, Massachusetts: Belknap Press of Harvard University Press. (1998), http://www.hup.harvard.edu/catalog.php?isbn=9780674004238&content=reviews

5.2 Implementing moral reasoning

This group focussed on issues regarding the implementation of moral reasoning in autonomous artificial agents. The group discussed the advantages and disadvantages of both immediate approaches to implementing moral reasoning: top-down and bottom-up. In a top-down approach one starts with a well defined task or objective that is to be solved by the system. The system is then designed to fulfil these requirements in the given environment or on the given data. In bottom-up approaches the environment or the given data are the starting point. The goal then is to pre-process and represent the input in a suitable manner so that in the end the desired task or objective can easily be fulfilled. A hybrid approach should also be possible to construct but it is less clear what the advantages and disadvantages of such an approach would be.

The group discussed issues of specification and verification with respect to both approaches, which in turn raised issues of transparency and accountability. The problem of verification is to prove formally that an autonomous agent's actions are a within the moral behaviour bounds of the society in which it operates. The issue of transparency, as with other complex machinery, is the issue of the level of detail of operation that will be made accessible to different concerned entities such as the end user, the manufacturer, licensed maintenance personnel, societal and government regulatory bodies etc.

When a machine is in a position to cause the death of numerous people, such as the autopilot of a passenger airplane, certain safety standards are required. An autopilot is considered safe to operate if it operates without causing an accident in a certain "high" number of cases. It seems evident that such safety requirements will need to be specified for autonomous machines capable of making decisions in moral dilemmas. The question of how safe is "safe enough" needs to be further discussed in this context. Also taking passenger aircraft as an exemplar, the group agreed on the need for some classes of moral agents – driverless cars for instance – to be equipped with an "ethical black box"; a device that will allow the internal ethical decision making processes to be recorded for later review during, for example, an accident investigation.

The group discussed possible effects that a moral reasoning machine can have on society. By implementing one moral code over another, a manufacturer may implicitly impose one culture's morality on a culture that respects different values than the manufacturer's. In addition, introducing machines capable of moral reasoning to a society may also impact that society, and how they behave towards such machines, in unpredictable ways. The behaviour of the machines may not cause any physical harm, but set in train unintended psychological harms. These issues must also be taken into consideration when the behaviour of an autonomous system is designed.

Lastly the group discussed issues involved in protecting the operation of an autonomous system from malicious or mischievous influence by users and society, which we termed "the dark side" of moral machines. Each of the approaches to implementing moral reasoning is susceptible to different kinds of vulnerabilities, which must also be taken into account.

The group captured each of the four areas of discussion: Approaches, Specification & Verification, Transparency & Accountability and Dangerous & Unethical AI as four mind-maps, and resolved to draft a joint paper provisionally entitled "Towards Moral Robots". Figures 1, 2, 3 and 4 include the Mind Maps of these discussions.

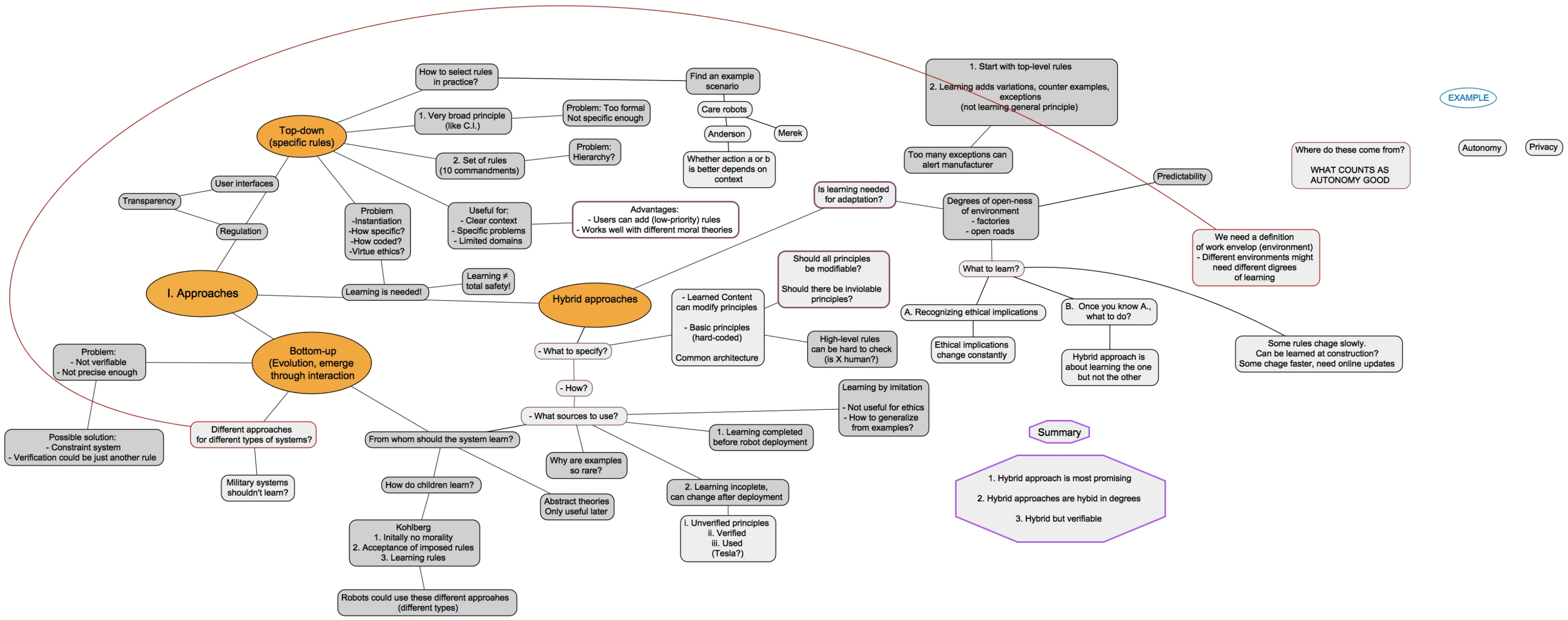

Figure 1 Approaches discussion Mind Map.

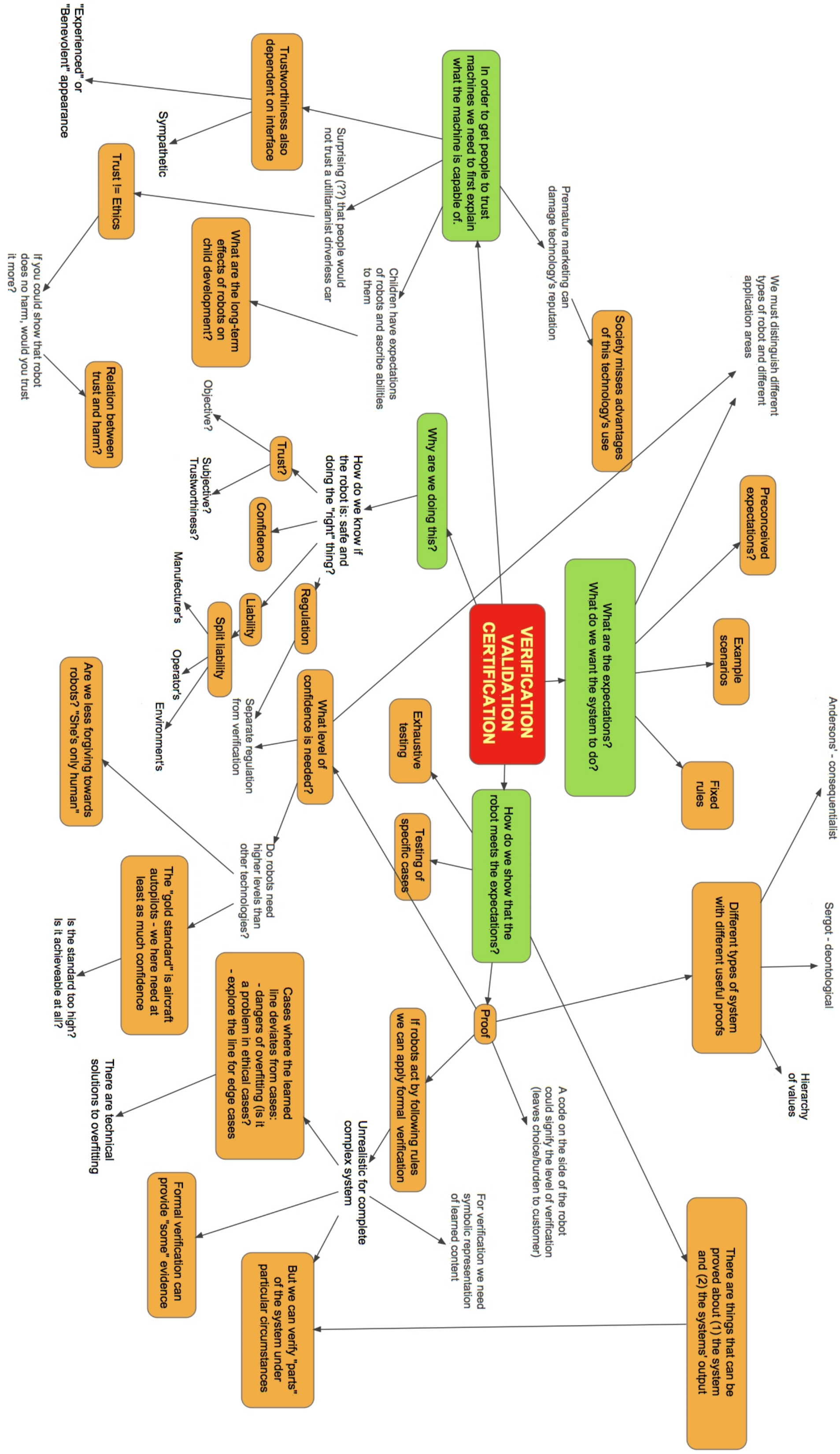
VERIFICATION
VALIDATION
CERTIFICATION
What are the expectations? What do we want the system to do?
Preconceived expectations?
Example scenarios
Fixed rules
We must distinguish different types of robot and different application areas
How do we show that the robot meets the expectations?
Exhaustive testing
Testing of specific cases
Proof
Different types of system with different useful proofs
Andersons' - consequentialist
Sergot - deontological
Hierarchy of values
A code on the side of the robot could signify the level of verification (leaves choice/burden to customer)
If robots act by following rules we can apply formal verification
Unrealistic for complete complex system
For verification we need symbolic representation of learned content
Formal verification can provide "some" evidence
But we can verify "parts" of the system under particular circumstances
There are things that can be proved about (1) the system and (2) the systems' output
Cases where the learned line deviates from cases: - dangers of overfitting (is it a problem in ethical cases? - explore the line for edge cases
There are technical solutions to overfitting
What level of confidence is needed?
Do robots need higher levels than other technologies?
The "gold standard" is aircraft autopilots - we here need at least as much confidence
Is the standard too high? Is it achieveable at all?
Are we less forgiving towards robots? "She's only human"
Separate regulation from verification
Regulation
Liability
Split liability
Manufecturer's
Operator's
Environment's
Why are we doing this?
How do we know if the robot is: safe and doing the "right" thing?
Trust?
Objective?
Subjective? Trustworthiness?
Confidence
In order to get people to trust machines we need to first explain what the machine is capable of.
Premature marketing can damage technology's reputation
Society misses advantages of this technology's use
Children have expectations of robots and ascribe abilities to them
Surprising (??) that people would not trust a utilitarianist driverless car
What are the long-term effects of robots on child development?
Trustworthiness also dependent on interface
"Experienced" or "Benevolent" appearance
Sympathetic
Trust != Ethics
If you could show that robot does no harm, would you trust it more?
Relation between trust and harm?

Figure 2 Verification, Validation and Certification discussion Mind Map.

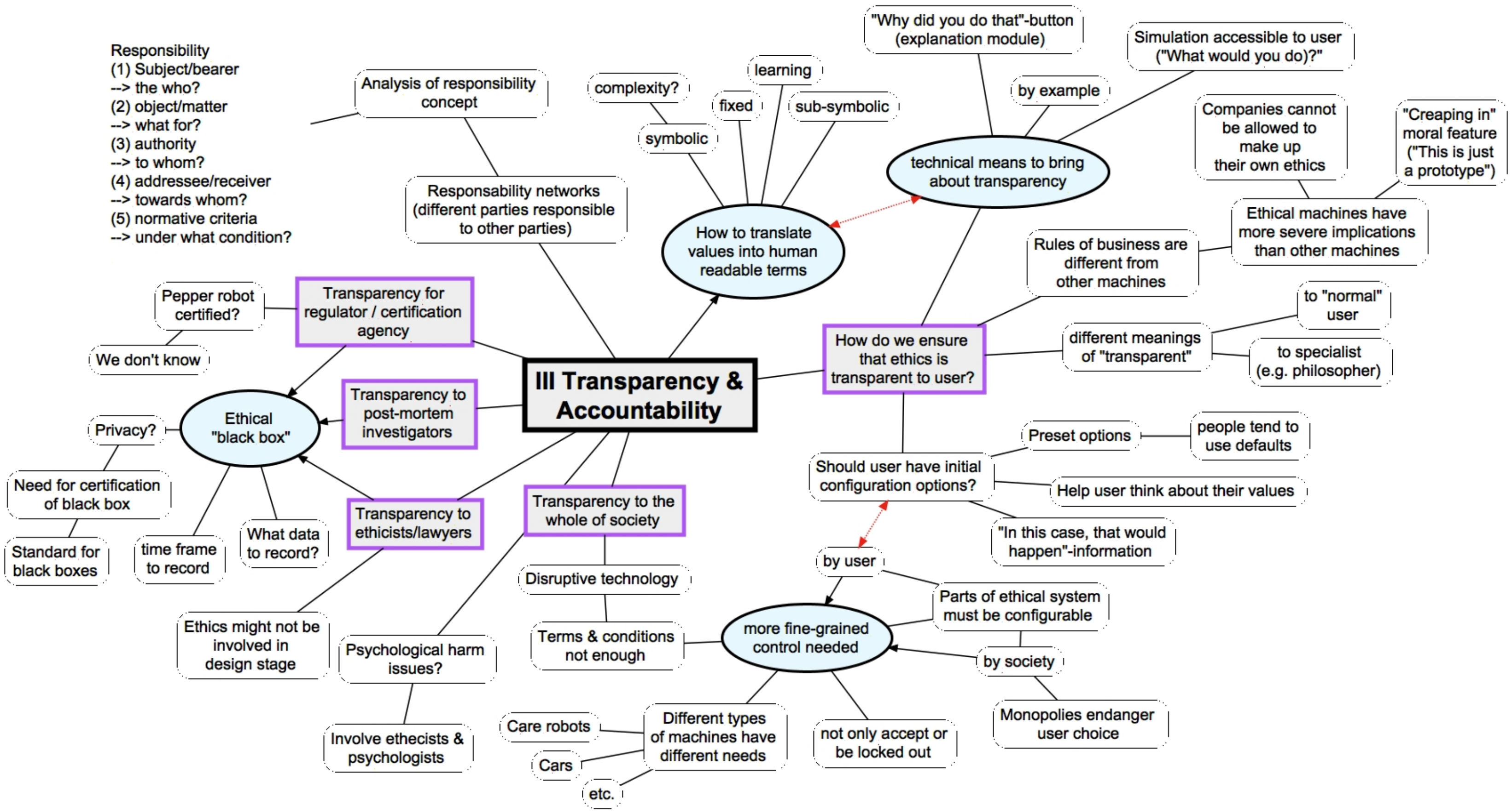
III Transparency & Accountability
Responsibility
(1) Subject/bearer
--> the who?
(2) object/matter
--> what for?
(3) authority
--> to whom?
(4) addressee/receiver
--> towards whom?
(5) normative criteria
--> under what condition?
Analysis of responsibility concept
Responsability networks (different parties responsible to other parties)
complexity?
symbolic
fixed
learning
sub-symbolic
How to translate values into human readable terms
"Why did you do that"-button (explanation module)
by example
Simulation accessible to user ("What would you do)?"
technical means to bring about transparency
Companies cannot be allowed to make up their own ethics
"Creaping in" moral feature ("This is just a prototype")
Ethical machines have more severe implications than other machines
Rules of business are different from other machines
How do we ensure that ethics is transparent to user?
different meanings of "transparent"
to "normal" user
to specialist (e.g. philosopher)
Preset options
people tend to use defaults
Should user have initial configuration options?
Help user think about their values
"In this case, that would happen"-information
by user
Parts of ethical system must be configurable
by society
more fine-grained control needed
Monopolies endanger user choice
not only accept or be locked out
Different types of machines have different needs
Care robots
Cars
etc.
Terms & conditions not enough
Disruptive technology
Transparency to the whole of society
Psychological harm issues?
Involve ethecists & psychologists
Transparency to ethicists/lawyers
Ethics might not be involved in design stage
Transparency to post-mortem investigators
Transparency for regulator / certification agency
Pepper robot certified?
We don't know
Ethical "black box"
Privacy?
Need for certification of black box
Standard for black boxes
time frame to record
What data to record?

Figure 3 Transparency discussion Mind Map.

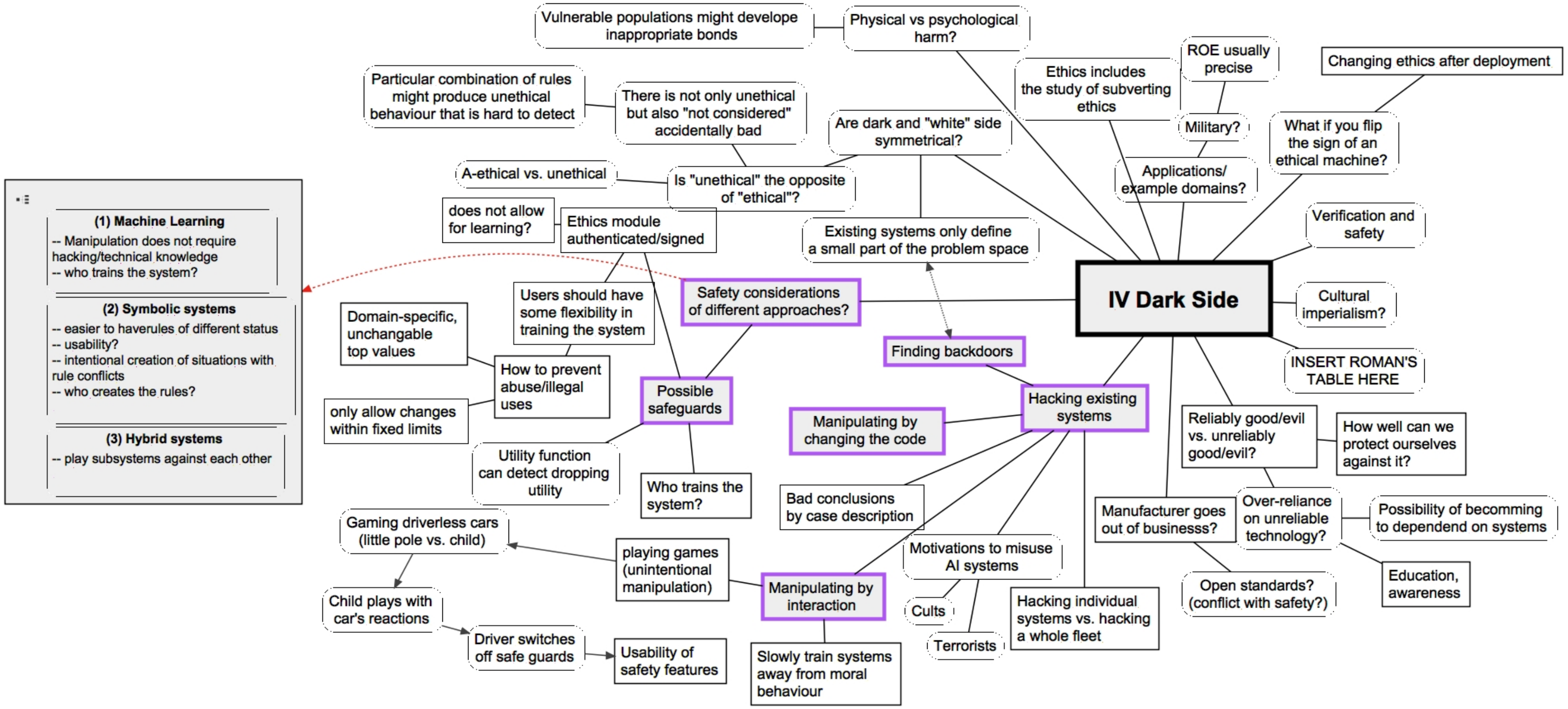
IV Dark Side
Vulnerable populations might develope inappropriate bonds
Physical vs psychological harm?
Particular combination of rules might produce unethical behaviour that is hard to detect
There is not only unethical but also "not considered" accidentally bad
Are dark and "white" side symmetrical?
Ethics includes the study of subverting ethics
ROE usually precise
Military?
Applications/ example domains?
Changing ethics after deployment
What if you flip the sign of an ethical machine?
A-ethical vs. unethical
Is "unethical" the opposite of "ethical"?
Existing systems only define a small part of the problem space
Verification and safety
Cultural imperialism?
INSERT ROMAN'S TABLE HERE
does not allow for learning?
Ethics module authenticated/signed
(1) Machine Learning
-- Manipulation does not require hacking/technical knowledge
-- who trains the system?
(2) Symbolic systems
-- easier to haverules of different status
-- usability?
-- intentional creation of situations with rule conflicts
-- who creates the rules?
(3) Hybrid systems
-- play subsystems against each other
Users should have some flexibility in training the system
Safety considerations of different approaches?
Domain-specific, unchangable top values
How to prevent abuse/illegal uses
Possible safeguards
Finding backdoors
only allow changes within fixed limits
Hacking existing systems
Manipulating by changing the code
Reliably good/evil vs. unreliably good/evil?
How well can we protect ourselves against it?
Utility function can detect dropping utility
Who trains the system?
Bad conclusions by case description
Manufacturer goes out of business?
Over-reliance on unreliable technology?
Possibility of becomming to dependend on systems
Gaming driverless cars (little pole vs. child)
playing games (unintentional manipulation)
Motivations to misuse AI systems
Manipulating by interaction
Cults
Terrorists
Hacking individual systems vs. hacking a whole fleet
Open standards? (conflict with safety?)
Education, awareness
Child plays with car's reactions
Driver switches off safe guards
Usability of safety features
Slowly train systems away from moral behaviour

Figure 4 Negative side-effects and potential issues with machine ethics discussion Mind Map.

Participants

- Michael Anderson
University of Hartford, US
- Albert Anglberger
LMU München, DE
- Zohreh Baniasadi
University of Luxembourg, LU
- Kevin Baum
Universität des Saarlandes, DE
- Vincent Berenz
MPI für Intelligente Systeme – Tübingen, DE
- Jan M. Broersen
Utrecht University, NL
- Vicky Charisi
University of Twente, NL
- Louise A. Dennis
University of Liverpool, GB
- Sjur K. Dyrkolbotn
Utrecht University, NL
- Michael Fisher
University of Liverpool, GB
- Joseph Halpern
Cornell University – Ithaca, US
- Holger Hermanns
Universität des Saarlandes, DE
- Johannes Himmelreich
HU Berlin, DE
- John F. Horty
University of Maryland – College Park, US
- Susan Leigh Anderson
University of Connecticut, US
- Robert Lieck
Universität Stuttgart, DE
- Christian List
London School of Economics, GB
- Andreas Matthias
Lingnan Univ. – Hong Kong, HK
- James H. Moor
Dartmouth College Hanover, US
- Marcus Pivato
University of Cergy-Pontoise, FR
- Marek Sergot
Imperial College London, GB
- Marija Slavkovik
University of Bergen, NO
- Janina Sombetzki
Universität Wien, AT
- Kai Spiekermann
London School of Economics, GB
- Alan FT Winfield
University of the West of England – Bristol, GB
- Roman V. Yampolskiy
University of Louisville, US