



DAGSTUHL REPORTS

Volume 4, Issue 12, December 2014

Socio-Technical Security Metrics (Dagstuhl Seminar 14491) <i>Dieter Gollmann, Cormac Herley, Vincent Koenig, Wolter Pieters, and Martina Angela Sasse</i>	1
The Synergy Between Programming Languages and Cryptography (Dagstuhl Seminar 14492) <i>Gilles Barthe, Michael Hicks, Florian Kerschbaum, and Dominique Unruh</i>	22
Programming Languages for Big Data (PlanBig) (Dagstuhl Seminar 14511) <i>James Cheney, Torsten Grust, and Dimitrios Vytiniotis</i>	48
Collective Adaptive Systems: Qualitative and Quantitative Modelling and Analysis (Dagstuhl Seminar 14512) <i>Jane Hillston, Jeremy Pitt, Martin Wirsing, and Franco Zambonelli</i>	68

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

April, 2015

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

License

This work is licensed under a Creative Commons Attribution 3.0 Unported license: CC-BY.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

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 Board

- Bernd Becker
- Stephan Diehl
- Hans Hagen
- Hannes Hartenstein
- Oliver Kohlbacher
- Stephan Merz
- Bernhard Mitschang
- Bernhard Nebel
- Bernt Schiele
- Nicole Schweikardt
- Raimund Seidel (*Editor-in-Chief*)
- Arjen P. de Vries
- Michael Waidner
- Reinhard Wilhelm

Editorial Office

Marc Herbstritt (*Managing Editor*)
Jutka Gasiorowski (*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>

Digital Object Identifier: 10.4230/DagRep.4.12.i

Socio-Technical Security Metrics

Edited by

Dieter Gollmann¹, Cormac Herley², Vincent Koenig³,
Wolter Pieters⁴, and Martina Angela Sasse⁵

- 1 TU Hamburg-Harburg, DE, diego@tu-harburg.de
- 2 Microsoft Research, Redmond, US, cormac@microsoft.com
- 3 University of Luxembourg, LU, vincent.koenig@uni.lu
- 4 TU Delft & University of Twente, NL, w.pieters@tudelft.nl
- 5 University College London, GB, a.sasse@cs.ucl.ac.uk

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14491 “Socio-Technical Security Metrics”. In the domain of safety, metrics inform many decisions, from the height of new dikes to the design of nuclear plants. We can state, for example, that the dikes should be high enough to guarantee that a particular area will flood at most once every 1000 years. Even when considering the limitations of such numbers, they are useful in guiding policy. Metrics for the security of information systems have not reached the same maturity level. This is partly due to the nature of security risk, in which an adaptive attacker rather than nature causes the threat events. Moreover, whereas the human factor may complicate safety and security procedures alike, in security this “weakest link” may be actively exploited by an attacker, such as in phishing or social engineering. In order to measure security at the level of socio-technical systems, one therefore needs to compare online hacking against such social manipulations, since the attacker may simply take the easiest path. In this seminar, we searched for suitable metrics that allow us to estimate information security risk in a socio-technical context, as well as the costs and effectiveness of countermeasures. Working groups addressed different topics, including security as a science, testing and evaluation, social dynamics, models and economics. The working groups focused on three main questions: what are we interested in, how to measure it, and what to do with the metrics.

Seminar November 30 to December 5, 2014 – <http://www.dagstuhl.de/14491>

1998 ACM Subject Classification K.6.5 Security and Protection

Keywords and phrases Security risk management, security metrics, socio-technical security, social engineering, multi-step attacks, return on security investment

Digital Object Identifier 10.4230/DagRep.4.12.1



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Socio-Technical Security Metrics, *Dagstuhl Reports*, Vol. 4, Issue 12, pp. 1–28

Editors: Dieter Gollmann, Cormac Herley, Vincent Koenig, Wolter Pieters, and Martina Angela Sasse



DAGSTUHL
REPORTS

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

1 Executive Summary


Dieter Gollmann

Cormac Herley

Vincent Koenig

Wolter Pieters

Martina Angela Sasse

License  Creative Commons BY 3.0 Unported license

© Dieter Gollmann, Cormac Herley, Vincent Koenig, Wolter Pieters, and Martina Angela Sasse

Introduction

Socio-technical vulnerabilities

Information security, or cyber security, is not a digital problem only. Humans have been termed “the weakest link”, but also physical access plays a role. Recent cyber attacks cleverly exploit multiple vulnerabilities of very different nature in the socio-technical systems that they target. For example, the StuxNet attack relied both on Industrial Control System (ICS) vulnerabilities and on the physical distribution of infected USB sticks, allowed by the business processes in the target facilities [8]. With new developments such as cloud computing, the attack surface of the systems only increases, and so do the options for potential attackers. At any company in the service supply chain, there may be malicious insiders or benevolent employees who fall victim to social engineering, and they influence the security of the system as a whole significantly. In order to compare and prioritize attacks and countermeasures, for example in terms of risk, the different types of vulnerabilities and threats need to be expressed in the same language. The seminar on “Socio-technical security metrics” aims at developing cross-domain metrics for this purpose.

Defining metrics

The idea of defining information security in terms of risk already appeared quite a while ago [2, 10]. Since then, many metrics have been proposed that aim to define attacks and attack opportunities in information systems in quantitative terms (see e.g. [7, 12]). Often, likelihood and impact of loss are mentioned as the key variables, from which risk can then be calculated. Furthermore, notions of vulnerability, difficulty, effort, cost, risk for the attacker, and many more, show up in the literature.

Even in a purely technical setting it is not always clear how all these different concepts are related. Still, including the human element forms a particular challenge, which deserves a separate event and a better integrated community. Too often it is thought that models of humans in the social sciences and models of technology are fundamentally incompatible. This inhibits progress on some very relevant questions: How does sending a phishing message compare to an SQL injection, in terms of the above mentioned variables? Or do we need additional notions in the technical models to express the human elements, or in the social science models to express the technical ones?

We thus need unified – or at least comparable – metrics that apply to all types of vulnerabilities. In order to represent socio-technical attacks, the key concepts need to apply to very different types of actions in an attack, including technical exploits and social engineering alike. This requires knowledge on technical infrastructures, social science, and actual incidents. To enable meaningful socio-technical security metrics, key features to be addressed in the seminar are outlined below.

Multi-step attacks

Cyber attacks, like StuxNet, tend to consist of multiple steps, combining technical and social or organizational vulnerabilities. Attack trees [17] are often used to represent possible multi-step attacks on systems, and they can be annotated with quantitative metrics. It has also been proposed to develop formal analysis techniques and simulations (“attack navigators”) that generate such trees based on a model of the socio-technical system at hand [5, 16]. By defining methods to calculate metrics for attacks from metrics for steps, one can compare the attacks in terms of the metrics, e.g. difficulty. However, next to methods for prediction, one would also want to be able to estimate the relevant parameters for the model based on observed events. For example, if one observes a set of successful and unsuccessful attacks, what does that say about the difficulty of the steps involved, and how does that influence the prediction of possible future events? Statistical methods from social science may assist here [15].

Estimating metrics from data

Data is thus key to developing good metrics, but obtaining them requires care. Given the data that is typically available in organizations already, including enterprise architecture, network logs, and potentially even organizational culture, how to obtain the right metrics from that data? What could be the role of “Big Data” in improving security metrics? And how to acquire additional data in tailor-made experiments? From the modeling point of view, a distinction can be made here between bottom-up approaches, leveraging existing data, and top-down approaches, defining targeted data collection methods and experiments. A good example on the social side are the phishing studies by Jakobsson & Ratkiewicz [6]. On the technical side, intrusion detection systems may constitute an important source of data.

Attacker models

As security threats originate from attackers and not from nature, attacker models are key for security metrics [9]. Attackers will adapt their strategies to the security situation, and also to newly deployed countermeasures. We therefore need meaningful and measurable features of attackers that can be used as a basis for the metrics. For example, the motivation of an attacker may determine the goal of the attack, the resources available to an attacker may determine the number of attacks that he can attempt, and attacker skill may determine the likelihood of success. Costs of an attack as well as risk of detection influence attacker behavior [3]. Again, the theoretical and empirical basis of such models needs to be carefully studied, and (security) economics may provide important insights here.

Countermeasures

All these aspects come together in one final goal: supporting investments. In order to estimate the cost-effectiveness of security measures (also called ROSI, for return on security investment), one would need metrics for both the risk prevented by the countermeasures, and of their cost. The former could be calculated based on the properties discussed above. The latter, however, is far from trivial by itself, as costs not only involve investment, but also operational costs. Operational costs, in turn, may include maintenance and the like, but an important factor in the total cost of ownership is impact on productivity. Security features may increase the time required to execute certain tasks, and people have a limited capacity for complying with security policies. If security is too cumbersome or misdirected,

people will find workarounds, and this may reduce the effect of the measures on risk [1]. Thus, metrics for countermeasure cost form an important topic in itself, requiring input from the human factors and usable security domains.

Another application area for the metrics would be selection among alternative system designs. For example, if two vendors offer the same equipment or service, but one is much cheaper, how to take security risk into account when making this decision? Both vendors as well as customers may be interested in security metrics from this point of view. However, metrics would need to be designed carefully in order to avoid creating perverse incentives, tweaking systems to score high on the metrics without actually being “better”.

Communities

In order to develop meaningful metrics for socio-technical security, participants from the following communities were invited:

- Security metrics and data-driven security, for obvious reasons;
- Security risk management, to provide input on suitable risk variables to be included;
- Security economics, to build upon economic theories of behavior of both attackers and defenders;
- Security architectures, to get relevant data on information system architecture and incidents;
- Formal methods, to analyze attack opportunities in complex systems;
- Social / crime science, to understand attacker behavior and the influence of controls;
- Human factors, to understand the impact of security controls on users.

Main findings

Paraphrasing some ancient philosophical questions (what is there, what can we know, what should we do), we can structure the main outcomes of this seminar as follows:

1. What properties are we interested in?
2. What can we measure?
3. What should we do with the measurements?

What properties

One of the main outcomes of the seminar is a much better view on which types of security metrics there are and for which purposes they can be used.

This leads to a distinction between metrics that exclude the real-life threat environment (type I) and metrics that include the real-life threat environment (type II). Metrics describing difficulty or resistance are typically of type I. They give a security metric that is independent of the actual activity of adversaries, or of the targets that they might be after. For example, which percentage of the people fall for a simulated phishing mail. This is similar to what Böhme calls “security level” [4]. The threat environment is often specified explicitly in such metrics, and the metrics may thus enumerate threat types. However, they do not estimate their occurrence rates, and in fact the occurrence rate is often controlled. In the phishing case, the researchers control the properties and occurrence of the phishing e-mails, and describe the e-mail (controlled threat) in their results.

Metrics describing loss (risk) or incidents are typically of type II. They describe undesired events that happen based on interaction of the system with a threat environment (activity of

adversaries), and their consequences. For example, the number of infected computers of a particular Internet Service Provider [18].

An illustration of this difference is the following. Consider two systems, system A and system B [13]. In system A, a locked door protects € 1,000. In system B, an identical locked door protects € 1,000,000. Which system is more secure? Or, alternatively, which door is more secure? One might say that system A is more secure, as it is less likely to be attacked (assuming the attacker knows the system). On the other hand, one might say that the doors are equally secure, as it is equally difficult to break the lock. The former argument is based on including an evaluation of the threat environment, the latter on excluding it.

Obviously, when trying to derive type II metrics from type I metrics, one needs metrics on the threat environment as well. For example, when one wants to calculate risk related to phishing attempts, and one knows how likely one's employees are to fall for phishing mails based on their sophistication, then one also needs information on the expected frequency of phishing mails of certain levels of sophistication in order to calculate the risk. Such models of the threat environment may be probabilistic or strategic (game-theoretic), representing non-adaptive and adaptive attackers, respectively. Probabilistic models, in turn, may be either frequentist (based on known average frequencies) or Bayesian (based on subjective probabilities). The various points of view have not been fully reconciled up to this point, although integration attempts have been made [14].

Another consideration is the integration of security metrics from different domains: digital, physical and social. Often, there are different styles of type I metrics, which one would like to integrate in a single type II metric representing the level of risk in a socio-technical system (e.g. an organization). Digital metrics may represent difficulty as required skill (e.g. CVSS), physical metrics may use required time (e.g. burglar resistance), and social metrics may use likelihood of success (e.g. likelihood of success of phishing attempts). Integration of these metrics is still an open challenge.

What measurements

The seminar discussed methods applied in different scientific communities for measurement purposes. Some of those methods rely on quantitative indicators, some rely on qualitative indicators, and some combine both. A further distinction can be made between subjective and empirical metrics, e.g. expert judgements versus monitoring data. Hereafter, and for the purpose of illustration, we have drawn a non-comprehensive list of such methods. They can be applied individually or in a complementary way, covering one measure or combined measures. A specific usage we consider underrepresented so far is the combination of methods in an effort to augment the measurement quality, or to provide information about the validity of a new measure. This approach has often been referred to, during the seminar, as triangulation of measures.

These are social methods discussed in the seminar:

- semi-structured interviews; in-depth interviews; surveys;
- observations of behavior;
- critical incident analysis;
- laboratory experiments; field experiments;
- expert / heuristic analysis / cognitive walkthrough;
- root cause analysis.

These are technical methods discussed in the seminar:

- security spending;

- implemented controls;
- maturity models;
- incident counts;
- national security level reports;
- service level agreements.

It is important to assess which type of metric (type I or type II) is produced by each of the techniques. For example, penetration testing experiments produce type I metrics, whereas incident counts produce type II. Maturity models and national security level reports may be based on a combination of type I and type II metrics. In such cases, it is important to understand what the influence of the threat environment on the metrics is, in order to decide how the metrics can be used.

What usage

Security metrics can contribute to answering questions about a concrete system or questions about a design (hypothetical system), and questions about knowledge versus questions about preferences. Here, we focus on a simpler distinction, namely between knowledge and design questions. In the case of knowledge questions, metrics are used to gather information about the world. In the case of design questions, metrics are used to investigate a design problem or to evaluate the performance of a design, such as a security control. In terms of knowledge questions, a typical usage discussed is a better understanding of the human factor in security. In terms of design, possible questions are how much security feedback a system should give to users or operators, or how to provide decision support for security investment.

Security metrics may have several limitations. In particular, many metrics suffer from various forms of uncertainty. It may be unclear whether the metrics measure the right thing (validity). Even if this is the case, random variations may induce uncertainty in the values produced (reliability). It is therefore important to understand the implications of such uncertainties for decisions that are made based on the metrics. Triangulation may contribute to the reduction of uncertainty. In some cases, quantitative metrics may not be possible at all, and qualitative methods are more appropriate.

Another limitation is that stakeholders may behave strategically based on what they know about the metrics (gaming the metrics). If stakeholders are rewarded when their security metrics become higher, they may put effort into increasing the metrics, but not “actual security”. Even if the metrics are valid under normal circumstances, this needs not be the case under strategic behavior.

Conclusions

Security is difficult to measure, which should not be a surprise to those involved. However, to understand security in today’s complex socio-technical systems, and to provide decision support to those who can influence security, rigorous conceptualisation, well-defined data sources and clear instructions for use of the metrics are key assets. This seminar laid the foundations for understanding and applying socio-technical security metrics.

In particular, we strove for clarity on (a) the different types of security metrics and their (in)compatibility, (b) the different sources and methods for data extraction, and (c) the different purposes of using the metrics, and the link with types, methods and sources. Several papers are planned as follow-up activities, as described in the reports of the working groups (Section 4). On many topics there are different views, which may not always be compatible, as was clear from the panel discussion (Section 5). Future follow-up seminars would be very valuable to address the open problems (Section 6).

References

- 1 A. Beautement, M. A. Sasse, and M. Wonham. The compliance budget: Managing security behaviour in organisations. In *Proc. of the 2008 Workshop on New Security Paradigms*, NSPW'08, pp. 47–58, New York, NY, USA, 2008. ACM.
- 2 B. Blakley, E. McDermott, and D. Geer. Information security is information risk management. In *Proc. of the 2001 New Security Paradigms Workshop*, pp. 97–104, New York, NY, USA, 2001. ACM.
- 3 A. Buldas, P. Laud, J. Priisalu, M. Saarepera, and J. Willemsen. Rational choice of security measures via multi-parameter attack trees. In *Critical Information Infrastructures Security*, volume 4347 of *LNCS*, pp. 235–248. Springer, 2006.
- 4 R. Böhme. Security metrics and security investment models. In Isao Echizen, Noboru Kunihiro, and Ryoichi Sasaki, editors, *Advances in Information and Computer Security*, volume 6434 of *LNCS*, pp. 10–24. Springer, 2010.
- 5 T. Dimkov, W. Pieters, and P. H. Hartel. Portunes: representing attack scenarios spanning through the physical, digital and social domain. In *Proc. of the Joint Workshop on Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security (ARSPA/WITS'10)*, volume 6186 of *LNCS*, pp. 112–129. Springer, 2010.
- 6 P. Finn and M. Jakobsson. Designing ethical phishing experiments. *Technology and Society Magazine, IEEE*, 26(1):46–58, 2007.
- 7 M. E. Johnson, E. Goetz, and S. L. Pfleeger. Security through information risk management. *IEEE Security & Privacy*, 7(3):45–52, May 2009.
- 8 R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE*, 9(3):49–51, 2011.
- 9 E. LeMay, M. D. Ford, K. Keefe, W. H. Sanders, and C. Muehrcke. Model-based security metrics using adversary view security evaluation (ADVISE). In *Proc. of the 8th Int'l Conf. on Quantitative Evaluation of Systems (QEST'11)*, pp. 191–200, 2011.
- 10 B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid, and D. Gollmann. Towards operational measures of computer security. *Journal of Computer Security*, 2(2–3):211–229, 1993.
- 11 H. Molotch. *Against security: How we go wrong at airports, subways, and other sites of ambiguous danger*. Princeton University Press, 2014.
- 12 S. L. Pfleeger. Security measurement steps, missteps, and next steps. *IEEE Security & Privacy*, 10(4):5–9, 2012.
- 13 W. Pieters. Defining “the weakest link”: Comparative security in complex systems of systems. In *Proc. of the 5th IEEE Int'l Conf. on Cloud Computing Technology and Science (CloudCom'13)*, volume 2, pp. 39–44, Dec 2013.
- 14 W. Pieters and M. Davarynejad. Calculating adversarial risk from attack trees: Control strength and probabilistic attackers. In *Proc. of the 3rd Int'l Workshop on Quantitative Aspects in Security Assurance (QASA)*, LNCS, Springer, 2014.
- 15 W. Pieters, S. H. G. Van der Ven, and C. W. Probst. A move in the security measurement stalemate: Elo-style ratings to quantify vulnerability. In *Proc. of the 2012 New Security Paradigms Workshop*, NSPW'12, pages 1–14. ACM, 2012.
- 16 C. W. Probst and R. R. Hansen. An extensible analysable system model. *Information security technical report*, 13(4):235–246, 2008.
- 17 B. Schneier. Attack trees: Modeling security threats. *Dr. Dobb's journal*, 24(12):21–29, 1999.
- 18 M. J. G. Van Eeten, J. Bauer, H. Asghari, and S. Tabatabaie. The role of internet service providers in botnet mitigation: An empirical analysis based on spam data. OECD STI Working Paper 2010/5, Paris: OECD, 2010.

2 Table of Contents

Executive Summary

Dieter Gollmann, Cormac Herley, Vincent Koenig, Wolter Pieters, and Martina Angela Sasse 2

Overview of Talks

Metrics for Security Awareness?
Zinaida Benenson 10

The National Role of CS Metrics
Kas P. Clark 10

Normative Security
Simon N. Foley 11

Socio-Technical Security Metrics
Aleksandr Lenin 11

Attack Trees and Socio-Technical Trees
Sjouke Mauw 12

Security-Related Behavior and Economics
Frank Pallas 12

Comparison of Cloud Provider Security
Sebastian Pape 13

Metrics for Security Behaviour in Organisations
Simon Parkin 13

Metrics in social engineering experiments
Wolter Pieters 13

Metrics for Security of Cooperating Systems
Roland Rieke 14

Three challenges with respect to measurement from a risk perspective
Ketil Stolen 15

Ideas for Socio-Technical Security Metrics
Axel Tanner 15

Susceptibility to Social Engineering
Sven Übelacker 15

How should we measure implementation complexity?
Jan Willemsen 16

Playing poker for fun, profit and science
Jeff Yan 16

Working Groups


Models, Economics and Threats – Working Group Report
Tristan Caulfield, Kas Clark, Trajce Dimkov, Carrie Gates, Cormac Herley, Mass Soldal Lund, Sjouke Mauw, Roland Rieke, and Jeff Yan 16

Social Dynamics Metrics – Working Group Report <i>Zinaida Benenson, Sören Bleikertz, Simon N. Foley, Carlo Harpes, Stewart Kowalski, Gabriele Lenzini, Daniela Oliveira, Simon Parkin, Shari Lawrence Pfleger, Paul Smith, and Sven Übelacker</i>	17
Testing, Evaluation, Data, Learning (Technical Security Metrics) – Working Group Report <i>Rainer Böhme, Michel Van Eeten, Simon Foley, Dina Hadžiosmanović, Aleksandr Lenin, Sebastian Pape, and Wolter Pieters</i>	20
Security as a Science – Working Group Report <i>Roeland Kegel, Vincent Koenig, Frank Pallas, Kai Rannenber, Ketil Stølen, and Axel Tanner</i>	22
Panel Discussion	24
Open Problems	26
Relation with previous seminars	26
Programme overview / Organisation	27
Participants	28

3 Overview of Talks

3.1 Metrics for Security Awareness?

Zinaida Benenson (*Universität Erlangen – Nürnberg, DE*)

License  Creative Commons BY 3.0 Unported license
© Zinaida Benenson

The usefulness of measures for raising security awareness in organizations and for the general public is controversially discussed in the current IT security research and practice. The differences in opinions range from publishing detailed guidelines for planning and conducting security awareness campaigns to reasoning that most security awareness measures are pointless. Measuring the effectiveness of security awareness interventions is an important tool for resolving this debate. Unfortunately, approaches from the computer science and information systems literature are not sufficiently well developed to fulfill this task. Moreover, the state of the art does not clearly define security awareness, which makes measuring anything connected to this concept even more difficult, if not impossible.

An attempt to characterize the existing security awareness definitions according to three orthogonal dimensions “Knowledge about threats”, “Knowledge about protection mechanisms” and “Behavior” is presented in this talk. Its purpose is to understand what security awareness actually means and what is missing in the current research on this topic. A preliminary version of this systematization can be found in the joint work with Norman Hänsch [1].

References

- 1 Hänsch, Norman and Benenson, Zinaida. *Specifying IT Security Awareness*. 1st Workshop on Security in Highly Connected IT Systems (SHCIS), collocated with 12th International Conference on Trust, Privacy, and Security in Digital Business (TrustBus), IEEE, 2014

3.2 The National Role of CS Metrics

Kas P. Clark (*Ministry of Security and Justice – The Hague, NL*)

License  Creative Commons BY 3.0 Unported license
© Kas P. Clark

The Dutch government needs the help of the research community to develop better, quantitative cyber security metrics for use at the national level. What data do we need and how can we combine it together to form coherent, relevant cybersecurity indicators?

3.3 Normative Security

Simon N. Foley (University College Cork, IE)

License © Creative Commons BY 3.0 Unported license
© Simon N. Foley

Joint work of Foley, Simon; Pieczul, Olgierd; Rooney, Vivien

Main reference O. Pieczul, S. N. Foley, V. M. Rooney, “I’m OK, You’re OK, the System’s OK: Normative Security for Systems,” in Proc. of the 2014 Workshop on New Security Paradigms Workshop (NSPW’14), pp. 95–104, ACM, 2014.

URL <http://dx.doi.org/10.1145/2683467.2683476>

The increasing scale and complexity of modern computer systems means that the provision of effective security is challenging, as well as being prohibitively expensive. Consequently, security tends to regulate those activities perceived to be critical, with the assumption that other unregulated activities, whether known or unknown, are not of significance. An added complication with security regimes that are overly strict, is that such unregulated activities can become the means of getting things done in the system. However, the difficulty is that these side-activities often lead to the compromise of security in a system. While security controls may provide monitoring and enforcement of the critical activities related to the security policy, little may be known about the nature of the other activities.

Normative security seeks to view a system as a society in which security is achieved by a combination of legislative provisions and normative behaviors. Drawing solely on legislative provisions is insufficient to achieve a just and orderly society. Similarly, security regimes that focus solely on security policies and controls are insufficient. Our position is that systems have analogous normative behaviors – behavioral norms – whereby the security of a system is based not only on the regulation of what is perceived to be its security critical activities, but also on the orderliness of its unregulated activities.

Using this analogy we are exploring how current theories about social norms in society can provide insight into using normative behavior in systems to help achieve security. We are investigating how these behavioral norms, representing potentially unknown side-activities, can be revealed by mining detailed system logs. The assumption is that, absent other information, adherence to past normative behavior can be taken as some indication of continuing orderliness. However, we note that these behavioral norms can be used to gauge the order or disorder in a system and, therefore, adherence to past normative behavior may also indicate a continuation of disorderliness

3.4 Socio-Technical Security Metrics

Aleksandr Lenin (Technical University – Tallinn, EE)

License © Creative Commons BY 3.0 Unported license
© Aleksandr Lenin

Joint work of Lenin, Aleksandr; Willemson, Jan

The talk outlines the socio-technical metrics used by the so-called “Failure-Free” models for quantitative security analysis, describes the problems obtaining quantitative input data from expert estimations, as well as suggests approaches that may be used to deal with the complexities of socio-technical security metrics.

References

- 1 Buldas, A., Lenin, A.: *New efficient utility upper bounds for the fully adaptive model of attack trees*. In Decision and Game Theory for Security – 4th International Conference,

- GameSec 2013, Fort Worth, TX, USA, November 11–12, 2013. Proceedings, pages 192–205, 2013.
- 2 Jürgenson, A., Willemson, J.: *On fast and approximate attack tree computations*. In IS-PEC, pages 56–66, 2010.
 - 3 Lenin, A., Buldas, A.: *Limiting adversarial budget in quantitative security assessment*. In Decision and Game Theory for Security – 5th International Conference, GameSec 2014, Los Angeles, CA, USA, November 6–7, 2014. Proceedings, pages 153–172, 2014.
 - 4 Lenin, A., Willemson, J., Sari, D. P.: *Attacker Profiling in Quantitative Security Assessment Based on Attack Trees*. In Simone Fischer-Hübner and Karin Bernsmed, editors, 19th Nordic Conference on Secure IT Systems, NordSec 2014, Tromsø, Norway, October 15–17, 2014. Proceedings, volume 8788 of Lecture Notes in Computer Science, pages 199–212. Springer, 2014.
 - 5 Pieters, W., Hadziomanovic, D., Lenin, A., Montoya, L., Willemson, J.: *Poster Abstract: TRESPASS: Plug-and-Play Attacker Profiles for Security Risk Analysis*. In Proceedings of the 35th IEEE Symposium on Security and Privacy, 2014. Poster and Extended Abstract.

3.5 Attack Trees and Socio-Technical Trees

Sjouke Mauw (University of Luxembourg, LU)

License  Creative Commons BY 3.0 Unported license
© Sjouke Mauw

In this presentation I sketched two tree-based modelling formalisms: attack trees and socio-technical trees. I briefly highlighted their syntax, semantics and pragmatics.

3.6 Security-Related Behavior and Economics

Frank Pallas (KIT – Karlsruher Institut für Technologie, DE)

License  Creative Commons BY 3.0 Unported license
© Frank Pallas

Main reference F. Pallas, “An Agency Perspective to Cloud Computing,” in J. Altmann, K. Vanmechelen, O.F. Rana (eds.), “Economics of Grids, Clouds, Systems, and Services – Proc. of 11th Int’l Conf. GECON 2014,” LNCS, Vol. 8914, pp. 36–51, Springer, 2014.

URL http://dx.doi.org/10.1007/978-3-319-14609-6_3

The application of economic theories and concepts to the field of information security has led to important findings and insights throughout the past years. In particular, the role of the yearly Workshop on the Economics of Information Security ¹ deserves explicit mention here.

From an economic perspective, achieving better information security does in most cases require cooperation between different players who pursue different goals. This cooperation, in turn, is hallmarked by information asymmetries, externalities and therefore often counter-productive incentives that lead to unfavourable security outcomes for all involved parties. In particular, this is the case when ultimate security-related decisions and activities are delegated from one party to another one which is assumed to have better capabilities and/or better situational knowledge allowing for more appropriate outcomes. This does, for example, apply to all security instruments focused on individual users as well as to most scenarios of cloud computing.

¹ See <http://econinfosec.org>

As laid out in the talk, economic agency theory provides valuable insights on the fundamental characteristics shaping such settings, the respective conflicts of interests and the reasonableness of different countermeasures. Socio-technical security metrics, in turn could be employed to diminish agency-related inefficiencies. In particular, they could in the future play an important role in the context of signalling (e.g. audit certificates), screening (e.g. inspections), and monitoring.

3.7 Comparison of Cloud Provider Security

Sebastian Pape (TU Dortmund, DE)

License © Creative Commons BY 3.0 Unported license
© Sebastian Pape

Joint work of Pape, Sebastian; Paci, Federica; Jürjens, Jan; Massacci, Fabio

Suppose, you want to start a new service and have the task of selecting a cloud provider. How do you determine which one is most secure? How do you decide which data is helpful for the selection? There already exist some approaches, but each of them (more or less) has its own utility function. How do you decide which approach returns the best results for ranking / comparison? Obviously the solution depends on the requirements of the tenant. Is it possible to come up with a 'requirement independent' ranking?

3.8 Metrics for Security Behaviour in Organisations

Simon Parkin (University College London, GB)

License © Creative Commons BY 3.0 Unported license
© Simon Parkin

In this short presentation I discuss monitoring of security behaviour in organisations, looking at user compliance and non-compliance, and the appropriateness of the security implementation for users and the business. I then discuss directions for measurement, including articulating incentives and costs for organisations to measure security behaviour, and the approachability and packaging of socio-technical expertise for practitioners and business in education and tools.

3.9 Metrics in social engineering experiments

Wolter Pieters (TU Delft, NL)

License © Creative Commons BY 3.0 Unported license
© Wolter Pieters

Joint work of Bullée, Jan-Willem; Montoya, Lorena; Pieters, Wolter; Junger, Marianne; Hartel, Pieter

Main reference J.-W. Bullée, L. Montoya, W. Pieters, M. Junger, P. Hartel, "The persuasion and security awareness experiment: reducing the success of social engineering attacks," *Journal of Experimental Criminology*. January 2015.

URL <http://dx.doi.org/10.1007/s11292-014-9222-7>

In social science, experimental setups can provide information on the risk associated with attack steps that involve social engineering, i.e. the manipulation of people. Typically, the threat environment is controlled by executing carefully scripted actions, which may involve phishing e-mails but also real-life interaction. One example of such an experiment is the "persuasion and security awareness experiment", in which we measured the success rate of obtaining physical credentials from university employees. It was found that a combined

awareness intervention was effective in reducing this success rate. Other possible metrics may include the time taken until success, or the stage in which the attack succeeds (if the script supports multiple stages). In this way, social science experiments with controlled threat environments can provide information on the difficulty of social engineering attack steps, and the effect of interventions. Because the threat environment is controlled, the metrics obtained are independent of actual attacker activity, which is not the case in studies that measure actual victimisation.

3.10 Metrics for Security of Cooperating Systems

Roland Rieke (*Fraunhofer SIT – Darmstadt, DE*)

License © Creative Commons BY 3.0 Unported license
© Roland Rieke

Main reference R. Rieke, “Abstraction-based analysis of known and unknown vulnerabilities of critical information infrastructures”, *International Journal of System of Systems Engineering (IJSSE)*, 1(1/2):59–77, 2008.

URL <http://dx.doi.org/10.1504/IJSSE.2008.018131>

Systems of systems that collaborate for a common purpose are called cooperating systems. They are characterised by freedom of decision and loose coupling of their components. Typical examples of cooperating systems are electronic health systems, vehicular ad hoc networks, distributed air traffic management systems, telephone systems, and electronic money transfer systems.


In this talk, three problems with respect to security metrics for cooperating systems have been addressed, namely, (1) abstract representation of security information, (2) security information quality, and (3) elicitation, linkage, and management of security information.

References

- 1 Roland Rieke. Abstraction-based analysis of known and unknown vulnerabilities of critical information infrastructures. *International Journal of System of Systems Engineering (IJSSE)*, 1(1/2):59–77, 2008.
- 2 Roland Rieke, Luigi Coppolino, Andrew Hutchison, Elsa Prieto, and Chrystel Gaber. Security and reliability requirements for advanced security event management. In Igor Kottenko and Victor Skormin, editors, *Computer Network Security*, volume 7531 of *Lecture Notes in Computer Science*, pp. 171–180. Springer Berlin Heidelberg, 2012.
- 3 Roland Rieke and Zaharina Stoynova. Predictive security analysis for event-driven processes. In *Computer Network Security*, volume 6258 of *LNCS* (pp. 321–328). Springer.
- 4 Roland Rieke, Jürgen Repp, Maria Zhdanova, and Jörn Eichler. Monitoring security compliance of critical processes. In *Parallel, Distributed and Network-Based Processing (PDP), 2014 22th Euromicro International Conference on*, pp. 525–560. IEEE Computer Society, Feb 2014.
- 5 Roland Rieke, Julian Schütte, and Andrew Hutchison. Architecting a security strategy measurement and management system. In *Proceedings of the Workshop on Model-Driven Security, MDsec’12*, pp. 2:1–2:6, New York, NY, USA, 2012. ACM.

3.11 Three challenges with respect to measurement from a risk perspective

Ketil Stolen (SINTEF – Oslo, NO)

License  Creative Commons BY 3.0 Unported license
© Ketil Stolen

One challenge is the gap between the data or information required by methods and tools for risk analysis put forward in main stream academic publications and the data available in practice.

A second challenge is the communication of risk relevant information among humans. What scales are best suited for what purpose? In particular, how should we measure likelihood and uncertainty?

A third challenge is the validation of risk models. How to determine that a risk model is sufficiently reliable?

3.12 Ideas for Socio-Technical Security Metrics

Axel Tanner (IBM Research GmbH – Zürich, CH)

License  Creative Commons BY 3.0 Unported license
© Axel Tanner

To induce discussions, four ideas for potential socio-technical security metrics are presented:

- Graph based: as many models used or proposed for security analysis, e.g. in TREsPASS, are based on graph structures, with additional data on nodes and edges, could we use graph characterising parameters, like connectivity or centrality, possibly in combination with data like 'value' on nodes and 'resistance' on edges to build and define security relevant metrics?
- Coverage based: many processes and operations happen in every organisation – can we measure what part of these is covered by operations and automated security policies? Or what part is covered by information flowing into tamper-proof log files?
- Reality gap: out of the security policies covering processes and operations in an organisation – how many and to which degree are these actually fulfilled in reality?
- Time to detect: in case of a breach of a security policy – how long will it take to detect this non-compliance?

3.13 Susceptibility to Social Engineering


Sven Übelacker (TU Hamburg-Harburg, DE)

License  Creative Commons BY 3.0 Unported license
© Sven Übelacker

In my short presentation I discussed my research on factors influencing the susceptibility to social engineering attacks which I try to categorise via existing research. Beside factors like socio-demographics, knowledge, impulsiveness, or stressors, I focused on the question: How big is the impact of personality traits on this susceptibility? I talked about my ongoing work on a scenario-based social engineering questionnaire including many of the aforementioned factors.

3.14 How should we measure implementation complexity?

Jan Willemson (Cybernetica AS – Tartu, EE)

License  Creative Commons BY 3.0 Unported license
© Jan Willemson

One of the weakest points in computer security are the implementations. The amount of potential mistakes correlates with complexity of the application. Should we acknowledge application complexity as one source of insecurity? Should we design a measure for this?

Some examples of implementation complexity:

- Highway speed limits do not guarantee the globally optimal outcome (e.g. that the total time needed for everyone to get home is minimal), but they have a virtue of being easy to follow and easy to verify.
- The definition of safe elliptic curves by Dan Bernstein and Tanja Lange includes several criteria that are designed to minimize the risk of getting the implementation wrong.

3.15 Playing poker for fun, profit and science

Jeff Yan (Newcastle University, GB)

License  Creative Commons BY 3.0 Unported license
© Jeff Yan


I propose to use poker as a new instrument for studying the psychology of deception, which is fundamental to many security and cybercrime problems such as social engineering. Poker enables the studies of a wide range of deceptive behaviours, and in these settings, observable, measurable and computable metrics are often available. Moreover, poker offers better ecological validity than trust games that have been widely used in economics studies.

I also explore how to inform cyber security with poker research, and discuss experiments designed for this purpose.

4 Working Groups

4.1 Models, Economics and Threats – Working Group Report

Tristan Caulfield, Kas Clark, Trajce Dimkov, Carrie Gates, Cormac Herley, Mass Soldal Lund, Sjouke Mauw, Roland Rieke, and Jeff Yan

License  Creative Commons BY 3.0 Unported license
© Tristan Caulfield, Kas Clark, Trajce Dimkov, Carrie Gates, Cormac Herley, Mass Soldal Lund, Sjouke Mauw, Roland Rieke, and Jeff Yan


In the cyber security domain, policy makers in both the public and private sectors make decisions regarding which project to fund, which legislation to propose and how to increase the overall resilience of their respective society or company given their finite resources. These decisions are made based on the best information available that given moment. Generally speaking, these decisions are based on qualitative metrics, such as expert or public opinion.

Some policy makers, including the Dutch Parliament, have officially asked that the existing metrics are supplemented with quantitative metrics. The underlying assumption is that quantitative metrics are more reliable as they are impartial and less susceptible to

anecdotal evidence. This working group is interested in exploring the available metrics and creating a framework to organize and evaluate them. To this end, this working group will: (1) identify relevant socio-technical security metrics, (2) estimate the desired properties of these metrics and (3) define a taxonomy to organize and correlate these metrics.

4.2 Social Dynamics Metrics – Working Group Report

Zinaida Benenson, Sören Bleikertz, Simon N. Foley, Carlo Harpes, Stewart Kowalski, Gabriele Lenzini, Daniela Oliveira, Simon Parkin, Shari Lawrence Pfleeger, Paul Smith, and Sven Übelacker

License  Creative Commons BY 3.0 Unported license

© Zinaida Benenson, Sören Bleikertz, Simon N. Foley, Carlo Harpes, Stewart Kowalski, Gabriele Lenzini, Daniela Oliveira, Simon Parkin, Shari Lawrence Pfleeger, Paul Smith, and Sven Übelacker

Introduction

Individuals continually interact with security mechanisms when performing tasks in everyday life. These tasks may serve personal goals or work goals, be individual or shared. These interactions can be influenced by peers and superiors in the respective environments (workplace, home, public spaces), by personality traits of the users, as well as by contextual constraints such as available time, cognitive resources, and perceived available effort.

All these influencing factors, we believe, should be considered in the design, implementation and maintenance of good socio-technical security mechanisms. Therefore, we need to observe reliable socio-technical data, and then transform them into meaningful and helpful metrics for user interactions and influencing factors.

More precisely, there are three main questions that the group discussed:

1. What data do we need to observe and what of this data we actually can observe and measure?
2. How can we observe and measure?
3. What can we do with the results of the observations?

What do we need to (and can) observe?

General data and metrics for individuals and groups

The discussion converged towards the idea of observing elements of behavior, not knowledge or attitudes, as the latter are not considered reliable indicators of security-related behavior. These observations can focus on behavior at an individual or group level.

Additionally to observing behavioral elements, e.g. patterns, we need to understand which factors influence people's behavior and trigger actions, and how to measure them. Among possible factors are personality traits, including irrationality and heuristics in decision-making. For example, deception susceptibility (or resilience) is an important personality trait that has been studied in the psychology of persuasion. The group also discussed measuring moral dimensions and risk perception. Other possible metrics include habits and cognitive abilities.

Collecting socio-demographic data such as age and sex is also important in order to know how these characteristics relate to other observable data.

Data and metrics for organizations

In the context of organizations, the group discussed what data and metrics can be used to indicate, estimate and quantify the security culture of the organization, its behavioral norms, risk perception (from the organization point of view), national differences, and the level of commitment of the group and individuals to the organization's goals. Metrics relating to individuals' capacity to expend effort for organization security without a perception of personal benefit (their 'compliance budget' is also important, as the employees can experience a number of draws from the security mechanisms on their available effort).

Data and metrics for adversaries

In the adversary domain, the group discussed metrics on attackers' risk perception of the possibility of being caught. Other possible metrics are organizational defense strengths as perceived by the attackers and attack resource costs (time, money and cognitive effort).

Data and metrics for employees

Collecting data on employees' privilege levels and roles is important, as this information helps to identify potentially dangerous deviations in behavior. Regarding employee activity, especially in the context of insider attacks, important metrics discussed were artifact collection rate per employee (number of artifacts per hour, week, and day), number and size of files transferred (flash drive, other local machines, remote machines), and number of artifacts printed.

Unintentional mistakes (such as accidentally printing out a document that is not allowed to be printed) or intentional workarounds in cases where security measures are perceived as impediments to task execution (such as sharing of login credentials) can mislead inferences about the prevalence of malicious insider behavior, indicating misconfigured systems and unusable security mechanisms instead. Therefore, it is important to develop metrics that can reduce false positives and lead to adjustments of security mechanisms. These metrics are especially related to organizational culture in terms of learning from mistakes, how mistakes are treated and reported, and also to individual metrics such as level of commitment and risk perception.

Limitations and future work

The group did not have enough time to discuss metrics for the general public and society, and also for special groups such as software developers, system administrators or managers, leaving these metrics to future work.

How can we observe and measure?

Most of the mentioned data can be collected using qualitative and quantitative methods from social sciences and psychology, although some data have technical nature (such as artifact collection rates). Quantitative methods include field and laboratory experiments, large-scale observations of behavior and carefully designed surveys, whereas qualitative methods include semi-structured interviews and in-depth observations (e.g., ethnography). Quantitative methods can be used to collect descriptive statistics as well as to test hypotheses.

Researchers and practitioners should pay attention to the constraints and limitations of the respective methods, such as external and internal validity and generalizability. Observations in organizations are challenging because it will usually take time before a relevant number of

events is collected. However, this approach represents probably the most natural means of measuring security behaviors.

What can we do with the results of the measurements?

Good social dynamics metrics support decision-making in an organization, improve its security processes and promote visibility of security policies. They also help with the communication of security needs at the high level of the organization, thus influencing the company's security budget.

Provide appropriate security

Security provisioning over time is important – stable metrics can allow baseline measurement of security before changes are made. This allows managers and providers to objectively measure elements of behavior over time to determine if end-users are being adequately supported.

Communicate human factors evidence

Social dynamics metrics can provide a common language that has the potential to support engagement with technology-oriented security researchers and practitioners. This common language would communicate the value of considering human factors in the design and implementation of future security solutions. Further, this common language would help to better frame the expectations and requirements for security training programs and security policies within organizations. We need both, social and technical metrics, as only a combination of them can provide enough supporting evidence for the design of better security processes.

Understand the appropriation of security


Social dynamic metrics also help discovering optimal levels of feedback about the security state of a system and of the control that the users can and should have over the security means. One possibility is the personalization of user engagement in security depending on their personality traits and experience (at the individual or per-task level, depending on the qualities of a task or group of tasks). Some people may wish to defer choices about security to the technology and receive minimal feedback (we call this *black box security*), whereas some other people may wish to have a lot of control and detailed feedback (*white box security*).

Next steps

The group discussed the importance of studying metrics for specific domains and producing a generalized framework for social security metrics, as some metrics will be valid for several domains or perspectives. A subset of the working group agreed to continue research within this area and to consolidate findings towards producing publications that support researchers and practitioners. This group includes Zinaida Benenson, Carlo Harpes, Stewart Kowalski, Gabriele Lenzini, Daniela Oliveira, Simon Parkin, and Sven Übelacker.

4.3 Testing, Evaluation, Data, Learning (Technical Security Metrics) – Working Group Report

Rainer Böhme, Michel Van Eeten, Simon Foley, Dina Hadžiosmanović, Aleksandr Lenin, Sebastian Pape, and Wolter Pieters

License  Creative Commons BY 3.0 Unported license
 © Rainer Böhme, Michel Van Eeten, Simon Foley, Dina Hadžiosmanović, Aleksandr Lenin, Sebastian Pape, and Wolter Pieters

Questions and objectives

The WG session started by brainstorming potential research questions around the topics of security evaluation and testing using security metric. Some of the questions were:

- What are different types of (technical) security metric?
- What kind of outcomes can we expect using different types of security metric?
- What kind of metric can be used for evaluation/testing purposes?
- What kind of data is required for using specific types of security metrics?

The WG then focused on two concrete objectives: *(i)* identify different dimensions to characterise a security metric and *(ii)* discuss the existing metrics with respect to the identified dimensions.

Properties of security metric

Böhme [2] presents a framework characterising security levels with different indicators mapped across *the level of abstraction* (as concrete or abstract) and *the amount of probabilistic nature in measuring* (as deterministic or probabilistic). The framework represented an excellent starting point for the WG discussion on different types of security metrics. For example, *security spending* represents an abstract but deterministic measure of security investment (i.e., as it represents the total spending). By contrast, specific *protection measures* represent concrete and deterministic measure (i.e., as they provide concrete technical checklists which can be directly related to security vulnerabilities). In this context, *incident counts* represent concrete yet probabilistic measure (i.e., as it reasons on the level of security based on the outcomes).

During the WG session, we introduced another aspect of the characterisation: *inclusion of threat environment*. More specifically, indicators like protection measures and penetration testing do not consider specific threat environment into the measure (as they mainly focus on the system itself). On the other hand, incident counting implicitly includes the specific threat environments (i.e., by referring to attackers and specific attack vectors).

Security metrics in practice

To understand how metrics used in practice map to the theoretical framework, the WG focused on discussing several approaches for measuring the security level:

Security service level agreement The agreements are typically used by security-service providers to indicate the scope and the character of the provided security service. Commonly, the agreements include guarantees on service availability (e.g., 99%), the maximum time for incidents response (e.g., 30 minutes) and repair time (e.g., 3 business days) [1].

Maturity models Organisations use maturity models to evaluate the overall level of security awareness and technological implementation of the organisation. Common approaches include maturity models like O-ISM3 [4], OpenSAMM [6] and BSIMM [3]. The models use

combinations of checklists to indicate the estimated level of security in the organisation or in software. For example, “have a lightweight approach to risk classification and prioritization” corresponds to the first (lowest) maturity model regarding architecture design, while “build proactive security architecture” corresponds to the fourth (highest) maturity level in architecture design.

Government-driven security level assessment Different countries annually publish a general, nation-wide, report on the cyber security level. One such example is the Cyber Security Assessment in the Netherlands, published by National Cyber Security Centre (NSCS) [5]. As the input information, the report uses incidents across different industry and public domains to draw the threat landscape, current trends and predictions.

Observations

With respect to the security metric framework, the WG discussions on the existing security metrics resulted in the following observations:

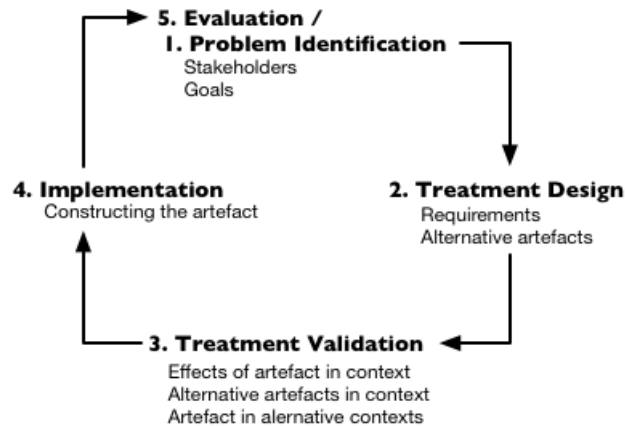
- Security service level agreements and maturity models represent metrics which weakly include threat environment into consideration (by focusing on protection measures) while security assessment reports largely include the threat environment (by using incident statistics).
- Metrics which do not consider the threat environment focus on security controls (e.g., protection measures).
- Metrics which consider the threat environment focus on evaluating the existing security controls (e.g., incidents indicate the accuracy of protection measures).
- Risk distribution in metrics is directly related to the inclusion of the threat environment. For example, security metrics in service level agreements focus on specifying controls (e.g., response time), and avoiding risk of guaranteeing the level of attack impact.
- A desirable security metric should include indicators across the whole spectrum of measurements (i.e., w/ and w/o threat environment).

Follow up

As the follow up activity, the participants of the WG agreed to revision the initial framework by introducing more practical examples and case studies, and exploring new possibilities for measuring the security level (e.g., measuring botnet/malware mitigation, measuring outcomes of penetration testing).

References

- 1 Avira. Service Level Agreement for Avira Managed Email Security of Avira. <http://www.avira.com/en/service-level-agreement>, 2011.
- 2 R. Böhme. Security metrics and security investment models. In I. Echizen, N. Kunihiro, and R. Sasaki, editors, *Advances in Information and Computer Security*, volume 6434 of *Lecture Notes in Computer Science*, pp. 10–24. Springer Berlin Heidelberg, 2010.
- 3 B. Chess and B. Arkin. Software security in practice. *IEEE Security & Privacy*, 9(2):89–92, 2011.
- 4 The Open Group. *Open Information Security Management Maturity Model (O-ISM3)*. Van Haren Publishing, 2011.
- 5 NCSC. Cyber Security Assessment Netherlands. Technical report, National Cyber Security Centre, 2014.
- 6 OWASP. Software Assurance Maturity Model.



■ **Figure 1** An illustration of the design cycle used in design science.

4.4 Security as a Science – Working Group Report

Roeland Kegel, Vincent Koenig, Frank Pallas, Kai Rannenber, Ketil Stølen, and Axel Tanner

License © Creative Commons BY 3.0 Unported license

© Roeland Kegel, Vincent Koenig, Frank Pallas, Kai Rannenber, Ketil Stølen, and Axel Tanner

This working group initially focused on identifying the type of research that is done in computer science in general, in an attempt to answer the question *what kind of science does computer security as a research field practice?* The discussions that followed from this question led to the conclusion that security research involves answering both knowledge questions and design questions:

- **Knowledge questions** are the archetypical form of science designed to answer specific questions by way of experimentation. The added value of answering these questions comes in the form of knowledge about the world as it is.
- **Design questions** are challenges: A call to change the world by introducing a new artefact in a certain context. This is something designed to improve the context into which it is introduced, such as a faster search algorithm, a test methodology or a new software system.

To define a scope for the discussion that fits the aim of the seminar, we then focused on design science and the role it plays in answering the following questions: *what can we measure, how do we measure it, and what can we do with these measurements?* We found that we could map these questions to elements of the design cycle illustrated by Wieringa [1] (see Figure 1). The design cycle corresponds closely with the engineering cycle of investigation, design, implementation and evaluation.

Metrics and the Design Cycle

The design cycle uses metrics in different stages of the cycle. Below is a summary of these relationships:

- **Problem investigation:** In this stage, problems and stakeholders are identified. Completion of this step results in a clearly defined problem. As such, the question *what do we measure?* can be answered using the results of this step.

- **Treatment design:** In this stage, the solution (new artefact) is designed to solve or mitigate the problem. The result of this step is a blueprint of the solution. By considering this proposed solution, we can find guidelines on *how to measure*, since the implementation guides what measurements are possible/practical.
- **Implementation:** In this stage, the proposed solution is inserted into the context. This corresponds to an *application* of the chosen security metrics.
- **Evaluation:** Finally, in this stage we consider the effects of applying this solution to the context. Using the measurements performed in the previous step, we can now consider the question *what to do with these measurements*.

Having identified the relationship of these questions and the design cycle, we can now reason about the issues with using the design cycle with security.

Problems and Pitfalls when using the Design Cycle

We identified three problems with the use of this cycle within the context of security research:

- **Invalidation by anecdote:** Often, a proposed treatment for a problem is invalidated by anecdotes, the availability of which being random. As a result, the random absence of anecdotes (i.e., the absence of proof for invalidation) might be confounded with the proof that no such anecdote exists (i.e., the proof of absence of arguments that could invalidate a treatment). Systematic evidence, supported by metrics, should however be sought for invalidation: a single counter-example to a security measure will lead to redesign of the treatment and only the proof of absence of such counter-examples will validate a design.
- **Skipping problem identification:** After the proposed treatment is deemed unsatisfactory, the problem is often not reconsidered. The treatment is immediately adapted to incorporate a defense to the anecdote. However, such counterexamples might be indicative of an incorrect problem investigation. Care has to be taken not to skip this step without due thought being given to the matter.
- **Problem considered static in following cycles:** When an iteration of the design cycle is complete and problems are identified with the current solution, often the problem definition is not reconsidered. Subsequent iterations of the process should consider whether the gathered evidence suggests that the problem identification needs updating (because of changing requirements, or identified shortcomings).

We feel that these problems are typical in security research. Good socio-technical security metrics can offer valuable support to mitigate these problems, especially for systemising the invalidation by anecdote rather than relying on random availability of anecdotes.

Conclusions

We feel that metrics should play a larger role in supporting treatment validation, lowering the reliance on randomly available anecdotes to validate (often expensive to implement) treatments. Additionally, we feel that metrics can play a vital role in reassessing whether the solution has proven successful. Finally, we are interested in the question of whether the design cycle is an effective methodology to use in the *development* of these metrics, rather than just the evaluation. To this end, as future work, we intend to use a case study in order to further investigate the interactions between design science and security metrics.

References

- 1 R. J. Wieringa. The design cycle. In *Design Science Methodology for Information Systems and Software Engineering*, pages 27–34. Springer Berlin Heidelberg, 2014.

5 Panel Discussion

At the end of the seminar, a panel discussion was organized with the following participants:

- Sasse, Martina Angela (moderator)
- Gates, Carrie
- Herley, Cormac
- Pfleeger, Shari Lawrence
- Stølen, Ketil

The panel helped identify fundamental differences in metrics, as well as open problems. It discussed a variety of key points as represented hereafter.

Definitions

The first discussion focused on defining what we need when intending to measure human behaviour in a security context. The panel suggests defining

- what behaviours we can expect to see;
- what triggers behaviours;
- what the range of behaviours is;
- what behaviours we want to encourage or discourage;
- what the differences between individual and group behaviours are;
- what triggers for sharing are;
- what attitudes lead to what behaviours.

The panelists identify an additional challenge which is understanding when people want to be in control, and when they want ‘to be taken care of’ in terms of security.

Data and research methods

The second point of discussion regarded the difficulty to rely on the ‘right’ data and the right methods for producing such data. There is a gap between the data that is required, and what is available – one reason being that data capture techniques originate from safety and process industry, where capturing data is much simpler than in cyber security.

The panel focused on the difficulty of getting reliable data; they formulated the following problems and recommendations:

- Use metrics that are as explicit as possible;
- People collecting data need hands-on experience of risk analysis – this is currently often confused with requirements analysis;
- Predict risk level after changes have been implemented;
- Combine risk analysis with other techniques to check risk model;
- Use two risk models – before and after;
- Combine with other measures, e.g. vulnerability scans, to check predictions – program and functional testing.

The panel agreed that there are many ways of measuring risk, e.g. attack trees; the ISO 27000 2-factor measure of risk consequence and risk likelihood; or by quantifying the ability of threat – e.g. OWASP risk rating methodology.

Transition to practice

It is felt that research methods can contribute to gathering reliable data. The transfer from research to practice however is difficult and might be very slow. To illustrate how distant research and practice might sometimes be, the panel provides a set of statements meant to describe the “industry view”, as opposed or distant to the research view:

- “Social metrics are hard and expensive, which is why we don’t do it”;
- “Security awareness – we assume that it works if people pass the test” (we want to believe it works);
- “Testing is hard and expensive to do” – technical responses are easy and cheap, and work ‘well enough’ – so people buy them – measuring staff ‘grumpiness’ is not easy and cheap;
- “We prefer capital expenditure to consultancy” – results need to be easy and cheap to measure;
- “It’s very hard to resist a good test” – people test and measure what’s easy to test and measure;
- “Standards drive adoption”.

In addition, the following observations were made:

- ‘Best practices’ are not quickly updated;
- Gartner and other influencers have a lot of power – everybody wants to be ‘best of breed & forward looking’ quadrant;
- As far as socio-technical security metrics are concerned, the phrase “garbage in, garbage out” applies;
- The industry approach to measurement is insufficiently mature – it’s a vacuum that research could fill;
- Honest and authoritative tests and criteria are needed.

The usage of metrics is another important point of discussion and the panelists feel that metrics and numbers are used to justify decisions already made. It is thus unsure *why* we want the measure. The answer ought to be: we should spend on things that have value for the system overall, not just to prevent something bad from happening, which is also the argument of Harvey Molotch [1]. We must combat exceptionalism – ‘security is special’ – as this seems to be an excuse for not working along scientific principles.

Also, we should not measure proxies or shortcuts. Examples:

- Studies on how many users give their passwords for a chocolate bar – these are numbers that don’t tell us very much;
- Mechanical Turk (mTurk) studies: the composition of participant groups is often limited, and motivation of those who participate for very little money may be to complete the study as quickly as possible.

The panelists feel there are too many of this type of debatable study – and bad data drives out good, myths about user behaviour are perpetuated. These hide the need to consider if technology and policies are actually working.

Finally, the panel agrees on a series of concrete recommendations and take-home messages:

- Be honest about what you don’t know;
- Throw out data that is not grounded, and start with what is left over;
- Triangulate your measurements or results with other metrics;
- Look at multiple metrics, especially context metrics, to understand what is causing changes in metrics;

- Relate your measurements to business metrics to understand cost and benefit of doing the measurements;
- Question how good the metric is. Are any of the insights actionable?

Conclusion

We need to work together to develop better studies, experimental paradigms, data collection and analysis techniques, and standards of proof and evidence.

References

- 1 H. Molotch *Against security: How we go wrong at airports, subways, and other sites of ambiguous danger*. Princeton University Press, 2014.

6 Open Problems

Despite interdisciplinary efforts, progress in socio-technical security is still slow. Research and practice are relying on security approaches that are felt to be unsatisfactory, but we are currently lacking demonstrably better alternatives. This seminar has made important contributions to advancing understanding, developing ontologies and identifying key issues, but much more research is needed in this domain. The following open problems have been identified:

- Reconciling metrics based on cost, time, and probability;
- Analysing security of complex systems based on attack step metrics;
- Relation with economic metrics;
- Relation with privacy metrics;
- Application to specific domains, such as critical infrastructures;
- Simulation of socio-technical systems;
- Defining “good” policies that are not only oriented towards liability but well grounded in what really happens in companies; that also rely on an understanding of human behavior rather than a prescription of behavior only;
- Triangulation of metrics;
- A clear definition of what socio-technical means, as opposed to the sum of two systems with different rules and concepts.

In particular, we recommend a follow-up seminar on analyzing the security of complex socio-technical systems based on metrics.

7 Relation with previous seminars

This seminar builds on the Insider Threat series (08302, 10341, 12501) and the seminar Secure Architectures in the Cloud (11492). However, this new seminar is focused on risk and security metrics, which is a specialized topic that can be of value to a broader community, and as such does not belong to the series.

Other related seminars include Verifying Reliability (12341), and From Security to Dependability (06371). Furthermore, a GI Dagstuhl Research Seminar on Dependability Metrics was held Oct. 30 – Nov. 1, 2005 (<http://link.springer.com/book/10.1007/978-3-540-68947-8/>)

	Sunday 30-11	Monday 1-12	Tuesday 2-12	Wednesday 3-12	Thursday 4-12	Friday 5-12
7:30 – 8:45		Breakfast	Breakfast	Breakfast	Breakfast	Breakfast**
9:00 – 10:30		Introduction	Pitches	Mid-seminar review	Pitches	Summary
10:30 – 10:45		Short break	Short break	Short break	Short break	Short break
10:45 – 12:15		Pitches	Working groups	Working groups	Working groups	Follow-up activities
12:15 – 13:30		Lunch	Lunch	Lunch	Lunch	Lunch
14:00 – 15:30		Working groups	Results from WGs	Social event	Results from WGs	Departure
15:30 – 16:00	Arrival*	Short break	Short break		Short break	
16:00 – 17:45		Tutorials	Demo session		Panel discussion	
18:00 – 19:00	Dinner	Dinner	Dinner		Dinner	
19:30 – 21:00	Arrival*	Social event	Meet / socialise	Meet / socialise		
21:00 -	Wine and cheese	Wine and cheese	Wine and cheese	Wine and cheese	Wine and cheese	

* Dagstuhl staff present 15:00 - 19:00

** Please check out before 9:00

Colour coding:

Non-seminar	Working groups
Food and drinks	Special sessions
Talks	High-level

■ **Figure 2** The programme of the seminar.

page/1). These seminars covered broader classes of metrics, not specifically focused on security or socio-technical integration. The present seminar brings together the socio-technical security angle from the Insider Threat series and the focus on metrics of the above mentioned seminars.

In the Lorentz Center in the Netherlands, a related seminar took place on Formal Methods for the Informal World (<http://www.lorentzcenter.nl/lc/web/2013/531//info.php3?wsid=531>). In this seminar, formal models of socio-technical systems were discussed, although not primarily focused on cyber security.

8 Programme overview / Organisation

In an effort to foster exchange among the participants and take full advantage of the Dagstuhl seminar concept, the organizers purposefully defined a program without long ex cathedra talks (Figure 2). The aim was twofold: (1) put emphasis on short presentations, involving a broad variety of people, each followed by sufficient discussion time; (2) avoid the style of presentations that are given in other contexts and that focus more on reporting rather than on sharing new ideas, visions, questions. As a result, the program included the following activities:

- 24 pitches (short talks, focusing at new ideas, visions, major questions);
- 3 tutorials (same objective than pitches, with increased talking and discussion time; suited for topics that are felt shared across the participants);
- 3 demo sessions (focus on concrete use-cases, video material, etc.);
- 1 panel discussion;
- 4 working groups (parallel break-out sessions; see Section Working Groups).

Furthermore, the parallel activities have been complemented by plenary sessions in order to present results to the entire group of participants and facilitate discussing those results.

Participants

- Zinaida Benenson
Univ. Erlangen-Nürnberg, DE
- Sören Bleikertz
IBM Research GmbH –
Zürich, CH
- Rainer Böhme
Universität Münster, DE
- Tristan Caulfield
University College London, GB
- Kas P. Clark
Ministry of Security and Justice –
The Hague, NL
- Trajce Dimkov
Deloitte – Eindhoven, NL
- Simon N. Foley
University College Cork, IE
- Carrie Gates
Dell Research, CA
- Dieter Gollmann
TU Hamburg-Harburg, DE
- Dina Hadziosmanovic
TU Delft, NL
- Carlo Harpes
itrust – Berbourg, LU
- Cormac Herley
Microsoft Corp. – Redmond, US
- Roeland Kegel
University of Twente, NL
- Vincent Koenig
University of Luxembourg, LU
- Stewart Kowalski
Gjøvik University College, NO
- Aleksandr Lenin
Technical University –
Tallinn, EE
- Gabriele Lenzini
University of Luxembourg, LU
- Mass Soldal Lund
Norwegian Defence Cyber
Academy – Lillehammer, NO
- Sjouke Mauw
University of Luxembourg, LU
- Daniela Oliveira
University of Florida –
Gainesville, US
- Frank Pallas
KIT – Karlsruher Institut für
Technologie, DE
- Sebastian Pape
TU Dortmund, DE
- Simon Parkin
University College London, GB
- Shari Lawrence Pfeleger
Dartmouth College Hanover, US
- Wolter Pieters
TU Delft and University of
Twente, NL
- Kai Rannenberg
Goethe-Universität Frankfurt am
Main, DE
- Roland Rieke
Fraunhofer SIT – Darmstadt, DE
- Martina Angela Sasse
University College London, GB
- Paul Smith
AIT – Wien, AT
- Ketil Stolen
SINTEF – Oslo, NO
- Axel Tanner
IBM Research GmbH –
Zürich, CH
- Sven Übelacker
TU Hamburg-Harburg, DE
- Michel van Eeten
TU Delft, NL
- Jan Willemson
Cybernetica AS – Tartu, EE
- Jeff Yan
Newcastle University, GB



The Synergy Between Programming Languages and Cryptography

Edited by

Gilles Barthe¹, Michael Hicks², Florian Kerschbaum³, and Dominique Unruh⁴

1 IMDEA Software Institute – Madrid, ES, gjbarthe@gmail.com

2 University of Maryland, US, mwh@cs.umd.edu

3 SAP Research – Karlsruhe, DE, florian.kerschbaum@sap.com

4 University of Tartu, EE, unruh@ut.ee

Abstract

Increasingly, modern cryptography (crypto) has moved beyond the problem of secure communication to a broader consideration of securing computation. The past thirty years have seen a steady progression of both theoretical and practical advances in designing cryptographic protocols for problems such as secure multiparty computation, searching and computing on encrypted data, verifiable storage and computation, statistical data privacy, and more.

More recently, the programming-languages (PL) community has begun to tackle the same set of problems, but from a different perspective, focusing on issues such as language design (e.g., new features or type systems), formal methods (e.g., model checking, deductive verification, static and dynamic analysis), compiler optimizations, and analyses of side-channel attacks and information leakage.

This seminar helped to cross-fertilize ideas between the PL and crypto communities, exploiting the synergies for advancing the development of secure computing, broadly speaking, and fostering new research directions in and across both communities.

Seminar November 30 to December 5, 2014 – <http://www.dagstuhl.de/14492>

1998 ACM Subject Classification D.4.6 Security and Protection, K.6.5 Security and Protection, F.3.1 Specifying and Verifying and Reasoning about Programs, D.2.4 Software/Program Verification

Keywords and phrases Security, Theory, Languages

Digital Object Identifier 10.4230/DagRep.4.12.29

Edited in cooperation with Matthew Hammer

1 Executive Summary

Michael Hicks

License  Creative Commons BY 3.0 Unported license
© Michael Hicks

The seminar schedule consisted of three components: short, two minute introduction talks (one for each participant), longer technical talks (Section 3) and open discussions on four different subjects. The first two days consisted of the introduction talks, followed by most of the technical talks. The seminar attendees had a mix of backgrounds, with one half (roughly) leaning heavily toward the PL (programming languages) side, and the other half leaning more towards the crypto side. The diversity of talks reflected this diversity of backgrounds,



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

The Synergy Between Programming Languages and Cryptography, *Dagstuhl Reports*, Vol. 4, Issue 12, pp. 29–47
Editors: Gilles Barthe, Michael Hicks, Florian Kerschbaum, and Dominique Unruh



DAGSTUHL
REPORTS

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

but there was much opportunity to meet in the middle and discuss open problems. The latter days mixed some remaining technical talks with open discussion sessions focusing on various problems and topics.¹ In particular, participants voted to select four breakout sessions: Secure Computation Compilers, Crypto verification, Obfuscation, and Verified implementations.

This section summarizes some interesting discussions from the seminar, in three parts. First, we consider the activities involved in developing programming languages the interface with cryptography, and surveying the research of the seminar participants. Second, we explore how reasoning in PL and Crypto compare and contrast, and how ideas from one area might be relevant to the other. Finally, we survey open problems identified during the discussions.

Programming languages for cryptography

One connection emerged repeatedly in the talks and discussions: the use of programming languages to do cryptography, e.g., to implement it, optimize it, and prove it correct.

Programming languages can be compiled to cryptographic mechanisms

Programming languages can make cryptographic mechanisms easier to use. For example, the systems Sharemind, ShareMonad, CBMC-GC, and WYSTERIA are all designed to make it easier for programmers to write secure multiparty computations (SMCs).

In an SMC, we have two (or more) parties X and Y whose goal is to compute a function F of their inputs x and y , whereby each party only learns the output $F(x, y)$, but does not “see” the inputs. Cryptographers have developed ways to compute such functions, such as garbled circuits² and computing on secret shares³, without need of a trusted third party. These systems shield the programmer from the workings of these mechanisms, compiling normal-looking programs to use the cryptography automatically. The languages can also provide additional benefits, such compiler-driven optimization.

This line of work is motivated by privacy- and/or integrity-preserving outsourcing of computation, e.g., as promised by The Cloud. Programming languages have been designed to compile to other kinds of crypto aside from SMC, like zero-knowledge proofs and authenticated data structures. Examples include Geppetto⁴, SNARKs for C⁵ and LambdaAuth⁶.

Combinations also exist, such as compiling to support Authenticated SNARKs.

Programming languages for implementing cryptography

The above languages aim to make computations secure through the use of cryptography, introduced by the language’s compiler. We are also interested in implementing the cryptographic algorithms themselves (e.g., for symmetric or public key encryption). The implementation

¹ As a break from the technical program, we went on a group outing to Trier on Wednesday afternoon, where we enjoyed a guided historical tour and enjoyed the city’s Christmas market.

² https://www.usenix.org/legacy/event/sec11/tech/full_papers/Huang.pdf

³ <http://www.math.ias.edu/~avi/PUBLICATIONS/MYPAPERS/GMW87/GMW87.pdf>

⁴ <https://eprint.iacr.org/2014/976.pdf>

⁵ <http://eprint.iacr.org/2013/507>

⁶ <http://amiller.github.io/lambda-auth/>

task could be made easier, more efficient, or more secure by employing a special-purpose language. Two representatives in this space are CAO⁷ and Cryptol⁸. Both are domain-specific, and both make it easier to connect implementations to tools for automated reasoning. The Seminar also featured work on synthesizing cryptography (block ciphers) from constraint-based specifications.⁹

Programming languages methods to prove security of cryptographic protocols and/or their implementations

When a cryptographer defines a cryptographic protocol, she must prove it is secure. Programming languages methods can be used mechanically confirm that a proof of security is correct. Systems like ProVerif¹⁰, CryptoVerif¹¹, EasyCrypt¹² and CertiCrypt¹³ support cryptographic protocol verification, with varying kinds of assurance. These systems build on ideas developed in general verification systems like Coq or Isabelle.

Likewise, when a programmer implements some cryptography (in a language like C), she would like to formally verify that the implementation is correct (no more Heartbleed!). For example, we'd like to know that an implementation does not have side channels, it uses randomness sufficiently, it has no buffer overflows, etc. Once again, verification can be achieved using tools that are underpinned by PL methods developed in formal verification research. Frama-C¹⁴ and Fstar¹⁵ have been used to verify implementations.

Formal reasoning for PL and Crypto

Beyond using PLs as a tool for easier/safer use of Crypto, there is an opportunity for certain kinds of thinking, or *reasoning*, to cross over fruitfully between the PL and Crypto communities. In particular, both communities are interested in formalizing systems and proving properties about them but they often use different methods, either due to cultural differences, or because the properties and systems of interest are simply different. During the seminar we identified both analogous, similar styles of reasoning in two communities and connection points between the different styles of reasoning.

Analogies between PL and Crypto reasoning

The Ideal/Real paradigm was first proposed by Goldreich, Micali, and Widgerson in their work on Secure Multiparty Computation (SMC) [3, 4], and further developed by Canetti in his universal composability (UC) framework¹⁶. The basic idea is to treat a cryptographic computation among parties as if it were being carried out by a trusted third party (the “ideal”), and then prove that the actual implementation (the “real”) emulates this ideal, in

⁷ <http://haslab.uminho.pt/mbb/software/cao-domain-specific-language-cryptography>

⁸ <https://galois.com/project/cryptol/>

⁹ <https://eprint.iacr.org/2014/774>

¹⁰ <http://prosecco.gforge.inria.fr/personal/bblanche/proverif/>

¹¹ <http://prosecco.gforge.inria.fr/personal/bblanche/cryptoverif/>

¹² <https://www.easycrypt.info/trac/>

¹³ <http://certicrypt.gforge.inria.fr/>

¹⁴ <http://frama-c.com/>

¹⁵ <http://research.microsoft.com/en-us/projects/fstar/>

¹⁶ <https://eprint.iacr.org/2000/067.pdf>

that the parties can learn nothing more than they would in a protocol involving a trusted party. (The paradigm also handles correctness, robustness, and other emergent properties.)

This is a classic kind of abstraction also present in formal verification: If a program P uses a module M that implements specification S , then relinking P to use M' , which also implements S , should preserve the correct execution of P . One talk, by Alley Stoughton, made the interesting observation that the Real/Ideal notion might be a suitable organizing principle around which to verify software is secure, essentially by using the Ideal as a richer kind of security property than is typical in PL (which often looks at properties like information flow control), and using abstraction in key ways to show it is enforced.

In the Crypto setting, the Real-to-Ideal connection is established probabilistically, considering a diminishing likelihood that a computationally bounded adversary would be able to tell the difference between the Real and Ideal. In the PL setting, the specification-implementation connection is established using methods of formal reasoning and logic, and usually without considering an adversary.

However, a notion of adversary does arise in PL-style reasoning. In particular, an adversary can be expressed as a context $C[\cdot]$ into which we place a computation e of interest that is subject to that adversary; the composition of the two is written $C[e]$. One PL property in this setup with a Crypto connection is contextual equivalence, which states that e and e' are equivalent iff for all contexts C the outcome of running $C[e]$ is the same as running $C[e']$ – e.g., both diverge or evaluate to the same result. In a PL setting this property is often of interest when proving that two different implementations of the same abstract data type have the same semantics (in all contexts). In a security setting we can view the contexts as adversaries, and e and e' as the Real and Ideal.

Another useful property is *full abstraction*.¹⁷ This property was originally introduced to connect an operational semantics to a denotational semantics – the former defines a kind of abstract machine that explains how programs compute, while the latter denotes the meaning of a program directly, in terms of another mathematical formalism (like complete partial orders). Both styles of semantics have different strengths, and full abstraction connects them: it requires that e and e' are observationally equivalent (according to the operational semantics) if and only if they have the same denotation (according to the denotational semantics).

In a Crypto setting, we might view the operational semantics as the Ideal and the denotational semantics as the Real, and full abstraction then states that despite the added observational power of the Real setting, an adversary cannot distinguish any more programs (i.e., learn any additional information) than he could in the Ideal setting. As a recent example of its use, Abadi and Plotkin used full abstraction to reason about the effectiveness of address space randomization. Another recent result is a fully abstract compiler from a type-safe high-level language to Javascript¹⁸; the compiler effectively defines the denotational semantics, and the fact that it is fully abstract means that the added adversarial power that Javascript provides cannot violate the source language's semantics.

Connections between PL and Crypto

The seminar also brought out ways that PL-style reasoning can be connected to Crypto-style reasoning for stronger end-to-end assurance of security. One connection point was at the Real/Ideal boundary. In particular, for privacy-preserving computation (or computation

¹⁷ <http://users.soe.ucsc.edu/~abadi/Papers/paper-csf-long.pdf>

¹⁸ <http://research.microsoft.com/en-us/um/people/nswamy/supp/full-abstraction.html>

preserving some other security property), Crypto-style reasoning can first be used to establish that the Real emulates the Ideal, and then PL-style reasoning can consider the security of the Ideal itself.

For example, consider the setting of SMC. Here, we have two (or more) parties X and Y that wish to compute a function F of their inputs x and y , whereby each party only learns the output $F(x, y)$, but does not “see” the inputs. That is, the security goal is to establish that the Real computation of $F(x, y)$ is indistinguishable from the Ideal model of executing F at a trusted third party. While Crypto can establish that a technique like garbled circuits effectively emulates a trusted third party, it does not establish that the output of F , even when computed by the Ideal, does not reveal too much information. For example, if $F(x, y) = y$ then X learns Y ’s value y directly. More subtly, if $F(x, y) = x > y$, then if $x = 1$, an output of TRUE tells X that Y ’s value $y = 0$. PL-style reasoning can be applied to functions F to establish whether they are sufficiently private, e.g., by using ideas like knowledge-based reasoning¹⁹ or type systems for differential privacy.²⁰ PL-style reasoning about knowledge can also be used to optimize SMCs by identifying places where a transformation would not affect security (e.g., no more is learned by an adversary observing the transformed program), but could improve performance.²¹

Another way to connect PL to Crypto is to factor security-sensitive computations into general-purpose and cryptographic parts. Then PL-style methods can be used to specify the overall computation with the Crypto parts carefully abstracted out. The proof of security then follows a PL approach, assuming guarantees provided by the Crypto parts, which are separately proved using Crypto techniques. In a sense we can think of the PL techniques as employing syntactic/symbolic reasoning, and the Crypto techniques employing computational/probabilistic reasoning.

This is the approach taken in LambdaAuth, a language extension for programming *authenticated data structures* (in the style of Merkle trees), in which the key idea involving the use of cryptographic hashes was abstracted into a language feature, and the proof of security combined a standard PL soundness proof along with a proof of the assumption that hash collisions are computationally difficult to produce. Recent work by Chong and Tromer on proof-carrying data similarly considers a language-level problem and proves useful guarantees by appealing to abstracted cryptographic mechanisms.²² Likewise, work on Memory Trace Obliviousness reasons about Oblivious RAM abstractly/symbolically in a PL setting to prove that the address trace of a particular program leaks no information.²³

Open problems

Beyond work that is being done, one goal of the seminar was to identify possible collaborations on future work. PL researchers and cryptographers work on common problems from different points of view, so one obvious next step is to collaborate on these problems.

One relevant problem is *side channels*. Cryptographers are concerned with side channels in their implementations, e.g., to make sure the time, space, or power consumption during

¹⁹ <http://www.cs.umd.edu/~mwh/papers/mardziel12smc.html>

²⁰ <http://www.cis.upenn.edu/~ahae/papers/dfuzz-popl2013.pdf>

²¹ <http://www.cs.umd.edu/~mwh/papers/rastogi13knowledge.html>

²² <https://eprint.iacr.org/2013/513>

²³ <http://www.cs.umd.edu/~mwh/papers/liu13oblivious.html>

an encryption/decryption operation does not reveal anything about the key. Likewise, PL folk care about side channels expressed at the language level, e.g. work by Andrew Myers' group on timing channels²⁴. Both groups bring a useful perspective.

Another common problem is *code obfuscation*. It was cryptographers that proved that virtual black box (VBB) obfuscation is impossible²⁵, and proposed an alternative indistinguishability-based definition. PL researchers, on the other hand, have looked at language-oriented views of obfuscation effectiveness, e.g., based on abstract interpretation²⁶. Just as the halting problem is undecidable, but practical tools exist that prove termination.²⁷ I believe that there is an opportunity here to find something useful, if not perfect.

Finally, the question of *composability* comes up in both Crypto and PL: Can we take two modules that provide certain guarantees and compose them to create a larger system while still ensuring properties proved about each module individually? Each community has notions for composability that are slightly different, though analogous, as discussed above. Can we make precise connections so as to bring over results from one community to the other? Crypto currencies, exemplified by BitCoin, are an area of exploding interest. An interesting feature about these currencies is that they provide a foundation for fair, secure multiparty computation, as demonstrated by Andrychowicz, Dziembowski, Malinowski, and Mazurek in their best paper at IEEE Security and Privacy 2014 [1, 2]. Could PL-style reasoning be applied to strengthen the guarantees provided by such computations? Cryptographic properties are often proved by making probabilistic statements about a system subject to a computationally bounded adversary. Could program analyses be designed to give probabilistic guarantees, drawing on the connection between adversary and context mentioned above, to thus speak more quantitatively about the chances that a property is true, or not, given the judgment of an analysis? How might random testing, which has proved highly useful in security settings, be reasoned about in a similar way?

Acknowledgements. We would like to thank Jonathan Katz for his initial involvement in organizing the seminar and Matthew Hammer for his help in preparing this report.

References

- 1 Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek. Secure Multiparty Computations on Bitcoin. 2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18–21, 2014, pp. 443–458, IEEE CS. <http://dx.doi.org/10.1109/SP.2014.35>
- 2 Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Łukasz Mazurek. Secure Multiparty Computations on Bitcoin. Cryptology ePrint Archive, Report 2013/784, 2013. <https://eprint.iacr.org/2013/784>
- 3 Oded Goldreich and Ronen Vainish. How to solve any protocol problem – an efficiency improvement (extended abstract). In Carl Pomerance, editor, *Advances in Cryptology – CRYPTO'87*, volume 293 of *Lecture Notes in Computer Science*, pages 73–86. Springer Berlin Heidelberg, 1988.
- 4 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, July 1991.

²⁴ <http://www.cs.cornell.edu/andru/papers/pltiming.html>

²⁵ <https://www.iacr.org/archive/crypto2001/21390001.pdf>

²⁶ http://dx.doi.org/10.1007/978-3-642-33125-1_11

²⁷ <http://research.microsoft.com/en-us/projects/t2/>

2 Table of Contents

Executive Summary	
<i>Michael Hicks</i>	29
Talk Abstracts	
SNARKs on Authenticated Data	
<i>Manuel Barbosa</i>	37
Introduction to computer-aided cryptographic proofs	
<i>Gilles Barthe</i>	37
From CryptoVerif Specifications to Computationally Secure Implementations of Protocols	
<i>Bruno Blanchet</i>	37
A two-level approach for programming secure multi-party computing	
<i>Dan Bogdanov</i>	38
CBMC-GC: Secure Two-Party Computations in ANSI C	
<i>Niklas Buescher</i>	39
Enforcing Language Semantics Using Proof-Carrying Data	
<i>Stephen Chong</i>	39
Secure composition of protocols	
<i>Veronique Cortier</i>	40
WYSTERIA: A Programming Language for Generic, Mixed-Mode Multiparty Computations	
<i>Matthew A. Hammer</i>	40
Compiling SQL for encrypted data	
<i>Florian Kerschbaum</i>	41
Rational Protection Against Timing Attacks	
<i>Boris Köpf</i>	41
Proving the TLS Handshake Secure (as it is) – and will be	
<i>Markulf Kohlweiss</i>	42
Automated Analysis and Synthesis of Modes of Operation and Authenticated Encryption Schemes	
<i>Alex Malozemoff</i>	42
Crash Course on Cryptographic Program Obfuscation	
<i>Alex Malozemoff</i>	43
A Practical Testing Framework for Isolating Hardware Timing Channels	
<i>Sarah Meiklejohn</i>	43
Dejà Q: Using Dual Systems to Revisit q-Type Assumptions	
<i>Sarah Meiklejohn</i>	44
A Computational Model including Timing Attacks	
<i>Esfandiar Mohammadi</i>	44

Proving the Security of the Mini-APP Private Information Retrieval Protocol in EasyCrypt <i>Alley Stoughton</i>	45
Using the Real/Ideal Paradigm to Define Program Security <i>Alley Stoughton</i>	45
Enforcing Language Semantics Using Proof-Carrying Data <i>Eran Tromer</i>	45
Information leakage via side channels: a brief survey <i>Eran Tromer</i>	46
Verification of Quantum Crypto <i>Dominique Unruh</i>	46
Participants	47

3 Talk Abstracts

3.1 SNARKs on Authenticated Data

Manuel Barbosa (University of Minho – Braga, PT)

License © Creative Commons BY 3.0 Unported license
© Manuel Barbosa

Joint work of Barbosa, Manuel;Backes, Michael;Fiore, Dario; Reischuk, Raphael;

Main reference M. Backes, M. Barbosa, D. Fiore, R. Reischuk, “ADSNARK: Nearly Practical and Privacy-Preserving Proofs on Authenticated Data,” Cryptology ePrint Archive, Report 2014/617, 2014.

URL <http://eprint.iacr.org/2014/617>

Presentation of joint work with M. Backes, D. Fiore and R. Reischuk, available on ePrint. We discuss the problem of privacy-preserving proofs on authenticated data, where a party receives data from a trusted source and is requested to prove computations over the data to third parties in a correct and private way, i.e., the third party learns no information on the data but is still assured that the claimed proof is valid. We formalize the above three-party model, discuss concrete application scenarios, and then we design, build, and evaluate ADSNARK, a nearly practical system for proving arbitrary computations over authenticated data in a privacy-preserving manner. ADSNARK improves significantly over state-of-the-art solutions for this model. For instance, compared to corresponding solutions based on Pinocchio (Oakland’13), ADSNARK achieves up to 25x improvement in proof-computation time and a 20x reduction in prover storage space.

3.2 Introduction to computer-aided cryptographic proofs

Gilles Barthe (IMDEA Software – Madrid, ES)

License © Creative Commons BY 3.0 Unported license
© Gilles Barthe

In this tutorial I will present some recent developments in computer-aided cryptography.

3.3 From CryptoVerif Specifications to Computationally Secure Implementations of Protocols

Bruno Blanchet (INRIA – Paris, FR)

License © Creative Commons BY 3.0 Unported license
© Bruno Blanchet

Joint work of Cadé, David; Blanchet, Bruno

Main reference D. Cadé, B. Blanchet, “Proved Generation of Implementations from Computationally-Secure Protocol Specifications,” in Proc. of the 2nd Int’l Conf. on Principles of Security and Trust (POST’13), LNCS, Vol. 7796, pp. 63–82, Springer, 2013.

URL http://dx.doi.org/10.1007/978-3-642-36830-1_4

This talk presents a novel technique for obtaining implementations of security protocols, proved secure in the computational model. We formally specify the protocol to prove, we prove this specification secure using the computationally-sound protocol verifier CryptoVerif, and we automatically translate it into an implementation in OCaml using a new compiler that we have implemented. We proved that our compiler preserves security. We applied this approach to the SSH Transport Layer protocol: we proved the authentication of the

server and the secrecy of the session keys in this protocol and verified that the generated implementation successfully interacts with OpenSSH. The secrecy of messages sent over the SSH tunnel cannot be proved due to known weaknesses in SSH with CBC-mode encryption.

References

- 1 David Cadé and Bruno Blanchet. From computationally-proved protocol specifications to implementations and application to SSH. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 4(1):4–31, Mar. 2013.
- 2 David Cadé and Bruno Blanchet. Proved generation of implementations from computationally secure protocol specifications. *Journal of Computer Security*. To appear.

3.4 A two-level approach for programming secure multi-party computing

Dan Bogdanov (*Cybernetica AS – Tartu, EE*)

License © Creative Commons BY 3.0 Unported license
© Dan Bogdanov

Joint work of Bogdanov, Dan; Kerik, Liisi; Laud, Peeter; Pankova, Alisa; Pettai, Martin; Randmets, Jaak; Ristioja, Jaak; Siim, Sander; Tarbe, Karl

Main reference D. Bogdanov, P. Laud, J. Randmets, “Domain-Polymorphic Programming of Privacy-Preserving Applications,” in Proc. of the 1st ACM Workshop on Language Support for Privacy-enhancing Technologies (PETShop’13), pp. 23–26, ACM, 2013.

URL <http://dx.doi.org/10.1145/2517872.2517875>

The implementation of secure multi-party computation applications needs specialized programming tools to hide the complexity of cryptography from the developer. Furthermore, secure multi-party computation seems to fit naturally into shared data analysis. We need tools that keep development simple, while preserving optimization opportunities and allowing formal security analyses.

Our solution is to separate the development into two layers. First, a high-level imperative language is used by the IT system developer to implement the algorithms and business logic. This language is independent of the underlying cryptographic protocols and the number of parties used in the execution. It emits bytecode that is interpreted by a specific virtual machine.

Second, a lower level language is used to implement the atomic secure operations in this virtual machine. This language is used by experts in secure computation to implement the protocols. Thus, it knows about parties, network channels and other necessary primitives. The language can be functional in order to simplify optimization and security analysis.

We have implemented this model in the Sharemind secure multi-party computation system with good results. The high-level language SecreC is used by non-cryptographers to implement real-world applications and it has a standard library of over 25 000 lines of code. For the lower layer, we have two options. Our own protocol DSL is a functional language for implementing protocols based on secret sharing. But we also support protocols generated by the CBMC-GC compiler as Boolean circuits.

References

- 1 Dan Bogdanov, Peeter Laud, Jaak Randmets. *Domain-Polymorphic Programming of Privacy-Preserving Applications*. In Proceedings of the First ACM Workshop on Language Support for Privacy-enhancing Technologies, PETShop 2013, ACM Digital Library. 2013.

- 2 Dan Bogdanov, Peeter Laud, Jaak Randmets. *Specifying Sharemind’s Arithmetic Black Box*. In Proceedings of the First ACM Workshop on Language Support for Privacy-enhancing Technologies, PETShop 2013, ACM Digital Library. 2013.

3.5 CBMC-GC: Secure Two-Party Computations in ANSI C

Niklas Buescher (TU Darmstadt, DE)

License © Creative Commons BY 3.0 Unported license

© Niklas Buescher

Joint work of Franz, Martin; Holzer, Andreas; Katzenbeisser, Stefan; Schallhart, Christian; Veith, Helmut

Main reference M. Franz, A. Holzer, S. Katzenbeisser, C. Schallhart, H. Veith, “CBMC-GC: An ANSI C Compiler for Secure Two-Party Computations,” in Proc. of the 23rd International Conference on Compiler Construction (CC’14), LNCS, Vol. 8409, pp. 244–249, Springer, 2014.

URL http://dx.doi.org/10.1007/978-3-642-54807-9_15

Secure two-party computation (STC) is a computer security paradigm where two parties can jointly evaluate a program with sensitive input data, provided in parts from both parties. By the security guarantees of STC, neither party can learn any information on the other party’s input while performing the STC task. For a long time thought to be impractical, until recently, STC has only been implemented with domain-specific languages or hand-crafted Boolean circuits for specific computations. Our open-source compiler CBMC-GC is the first ANSI-C compiler for STC. It turns C programs into Boolean circuits that fit the requirements of garbled circuits, a generic STC approach based on circuits. Here, the size of the resulting circuits plays a crucial role since each STC step involves encryption and network transfer and is therefore extremely slow when compared to computations performed on modern hardware architectures. We report on newly implemented circuit optimization techniques that substantially reduce the circuit sizes compared to the original release of CBMC-GC.

References

- 1 A. Holzer, M. Franz, S. Katzenbeisser, and H. Veith. Secure Two-Party Computations in ANSI C. In *CCS’12*, 2012.
- 2 M. Franz, A. Holzer, S. Katzenbeisser, C. Schallhart, and H. Veith. CBMC-GC: An ANSI C Compiler for Secure Two-Party Computations, In *CC’14*, 2014

3.6 Enforcing Language Semantics Using Proof-Carrying Data

Stephen Chong (Harvard University – Cambridge, US)

License © Creative Commons BY 3.0 Unported license

© Stephen Chong

Joint work of Chong, Stephen; Tromer, Eran; Vaughan, Jeffrey A.

Main reference S. Chong, E. Tromer, J. A. Vaughan, “Enforcing Language Semantics Using Proof-Carrying Data,” Cryptology ePrint Archive, Report 2013/513, 2013.

URL <http://eprint.iacr.org/2013/513>

Sound reasoning about the behavior of programs relies on program execution adhering to the language semantics. However, in a distributed computation, when a value is sent from one party to another, the receiver faces the question of whether the value is well-traced: could it have been produced by a computation that respects the language semantics? If not, then accepting the non-well-traced value may invalidate the receiver’s reasoning, leading to bugs or vulnerabilities.

Proof-Carrying Data (PCD) is a recently-introduced cryptographic mechanism that allows messages in a distributed computation to be accompanied by proof that the message, and the history leading to it, complies with a specified predicate. Using PCD, a verifier can be convinced that the predicate held throughout the distributed computation, even in the presence of malicious parties, and at a verification cost that is independent of the size of the computation producing the value. Unfortunately, previous approaches to using PCD required tailoring a specialized predicate for each application, using an inconvenient formalism and with little methodological support.

We connect these two threads by introducing a novel, PCD-based approach to enforcing language semantics in distributed computations. We show how to construct an object-oriented language runtime that ensures that objects received from potentially untrusted parties are well-traced with respect to a set of class definitions. Programmers can then soundly reason about program behavior, despite values received from untrusted parties, without needing to be aware of the underlying cryptographic techniques.

3.7 Secure composition of protocols

Veronique Cortier (LORIA – Nancy, FR)

License © Creative Commons BY 3.0 Unported license
© Veronique Cortier

Joint work of Ciobaca, Stefan; Cortier, Veronique; Delaune, Stéphanie; Le Morvan, Éric

Consider your favorite key-exchange protocol. Assume it is secure. Is it possible to use it to implement a secure channel?

In all generality, the answer is no. In this talk, we review techniques for securely composing protocols, for various notions of composition.

3.8 Wysteria: A Programming Language for Generic, Mixed-Mode Multiparty Computations

Matthew A. Hammer (University of Maryland, US)

License © Creative Commons BY 3.0 Unported license
© Matthew A. Hammer

Joint work of Rastogi, Aseem; Hammer, Matthew A.; Hicks, Michael

Main reference A. Rastogi, M. A. Hammer, M. Hicks, “Wysteria: A Programming Language for Generic, Mixed-Mode Multiparty Computations,” in Proc. of the 2014 IEEE Symposium on Security and Privacy (SP’14), pp. 655–670, IEEE, 2014.

URL <http://dx.doi.org/10.1109/SP.2014.48>

URL <https://bitbucket.org/aseemr/wysteria/>

In a Secure Multiparty Computation (SMC), mutually distrusting parties use cryptographic techniques to cooperatively compute over their private data; in the process each party learns only explicitly revealed outputs. In this paper, we present WYSTERIA, a high-level programming language for writing SMCs. As with past languages, like Fairplay, WYSTERIA compiles secure computations to circuits that are executed by an underlying engine. Unlike past work, WYSTERIA provides support for mixed-mode programs, which combine local, private computations with synchronous SMCs. WYSTERIA complements a standard feature set with built-in support for secret shares and with wire bundles, a new abstraction that supports generic n-party computations. We have formalized WYSTERIA, its refinement

type system, and its operational semantics. We show that WYSTERIA programs have an easy-to-understand single-threaded interpretation and prove that this view corresponds to the actual multi-threaded semantics. We also prove type soundness, a property we show has security ramifications, namely that information about one party's data can only be revealed to another via (agreed upon) secure computations. We have implemented WYSTERIA, and used it to program a variety of interesting SMC protocols from the literature, as well as several new ones. We find that WYSTERIA's performance is competitive with prior approaches while making programming far easier, and more trustworthy.

3.9 Compiling SQL for encrypted data

Florian Kerschbaum (SAP AG – Karlsruhe, DE)

License © Creative Commons BY 3.0 Unported license
© Florian Kerschbaum

Joint work of Kerschbaum, Florian; Härterich, Martin; Kohler, Mathias; Hang, Isabelle; Schaad, Andreas; Schröpfer, Axel; Tighzert, Walter

Main reference F. Kerschbaum, M. Härterich, M. Kohler, I. Hang, A. Schaad, A. Schröpfer, W. Tighzert, “An Encrypted In-Memory Column-Store: The Onion Selection Problem,” in Proc. of the 9th Int'l Conf. on Information Systems Security (ICISS'13), LNCS, Vol. 8303, pp. 14–26, Springer, 2013.

URL http://dx.doi.org/10.1007/978-3-642-45204-8_2

We present a problem in processing SQL over encrypted data. Encrypted database enable outsourcing to the cloud, but require adapting the encryption scheme to the SQL operation. A problem arises when one operator requires a different encryption scheme than its predecessor. Most notably sorting of (or range queries on) homomorphically encrypted data is not possible. A query like “SELECT TOP 3 zipcode GROUP BY zipcode ORDER BY SUM(revenue)” cannot be performed on encrypted data. The solution to this problem is deeper query analysis and compilation. We build the operator tree (relational algebra) of the SQL query, split it at the bottom most conflict and execute one part on the database and one part on the client. This implies a performance penalty for transferring more data to the client for some queries and always for query analysis, but enables full SQL functionality and policy configuration of the encryption (and hence potentially increasing security).

References

- 1 Florian Kerschbaum, Martin Härterich, Mathias Kohler, Isabelle Hang, Andreas Schaad, Axel Schröpfer, and Walter Tighzert. An Encrypted In-Memory Column-Store: The Onion Selection Problem. In *Proceedings of the 9th International Conference on Information Systems Security (ICISS)*, 2013.

3.10 Rational Protection Against Timing Attacks

Boris Köpf (IMDEA Software – Madrid, ES)

License © Creative Commons BY 3.0 Unported license
© Boris Köpf

We present a novel approach for reasoning about the trade-off between security and performance in timing attacks, based on techniques from game theory and quantitative information-flow analysis. Our motivating example is the combination of input blinding and discretization

of execution times, for which the trade-off between security and performance can be cast formally.

We put our techniques to work in a case study in which we identify optimal countermeasure configurations for the OpenSSL RSA implementation. We determine situations in which the optimal choice is to use a defensive, constant-time implementation and a small key, and situations in which the optimal choice is a more aggressively tuned (but leaky) implementation with a longer key.

References

- 1 Goran Doychev and Boris Köpf. Rational protection against timing attacks, 2015.
- 2 Boris Köpf and Markus Dürmuth. A provably secure and efficient countermeasure against timing attacks. In *CSF*, pages 324–335. IEEE, 2009.

3.11 Proving the TLS Handshake Secure (as it is) – and will be

Markulf Kohlweiss (Microsoft Research UK – Cambridge, GB)

License © Creative Commons BY 3.0 Unported license
© Markulf Kohlweiss

Joint work of M. Kohlweiss, K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Pironti, P-Y. Strub, S. Zanella-Béguélin

Main reference K. Bhargavan, C. Fournet, M. Kohlweiss, A. Pironti, P-Y. Strub, S. Zanella Béguélin, “Proving the TLS Handshake Secure (As It Is),” in Proc. of the 4th Annual Cryptology Conference – Advances in Cryptology – Part II (CRYPTO’14), LNCS, Vol. 8617, pp. 235–255, Springer, 2014.

URL http://dx.doi.org/10.1007/978-3-662-44381-1_14
URL <http://www.mitls.org>

The TLS Internet Standard features a mixed bag of cryptographic algorithms and constructions, letting clients and servers negotiate their use for each run of the handshake.

I present an analysis of the provable security of the TLS handshake, as it is implemented and deployed. To capture the details of the standard and its main extensions, it relies on miTLS, a verified reference implementation of the protocol. This motivates the use of new agile security definitions and assumptions for the signatures, key encapsulation mechanisms (KEM), and key derivation algorithms used by the TLS handshake. To validate the model of key encapsulation, the analysis shows that the KEM definition is satisfied by RSA ciphersuites under the plausible assumption that PKCS#1v1.5 ciphertexts are hard to re-randomize.

I also touch on the need to adapt our KEM model to support the recent session hash and extended master secret draft TLS extension that binds TLS master secrets to the context in which they were generated.

3.12 Automated Analysis and Synthesis of Modes of Operation and Authenticated Encryption Schemes

Alex Malozemoff (University of Maryland, US)

License © Creative Commons BY 3.0 Unported license
© Alex Malozemoff

Joint work of Malozemoff, Alex; Green, Matthew; Katz, Jonathan

Main reference A. J. Malozemoff, J. Katz, M.D. Green, “Automated Analysis and Synthesis of Block-Cipher Modes of Operation,” Cryptology ePrint Archive: Report 2014/774, 2014.

URL <https://eprint.iacr.org/2014/774>

In this talk, we present two approaches to synthesizing encryption schemes. We first discuss a work published at CSF 2014, where we synthesize block-cipher modes of operations, which

are mechanisms for probabilistic encryption of arbitrary length messages using any underlying block cipher. We propose an automated approach for the security analysis of block-cipher modes of operation based on a "local" analysis of the steps carried out by the mode when handling a single message block. We model these steps as a directed, acyclic graph, with nodes corresponding to instructions and edges corresponding to intermediate values. We then introduce a set of labels and constraints on the edges, and prove a meta-theorem showing that any mode for which there exists a labeling of the edges satisfying these constraints is secure (against chosen-plaintext attacks). This allows us to reduce security of a given mode to a constraint-satisfaction problem, which in turn can be handled using an SMT solver. We couple our security-analysis tool with a routine that automatically generates viable modes; together, these allow us to synthesize hundreds of secure modes.

In the second part of the talk, we discuss recent work extending this approach to authenticated encryption schemes, which both encrypts and authenticates arbitrary-length messages using a block-cipher as a building block.

3.13 Crash Course on Cryptographic Program Obfuscation

Alex Malozemoff (University of Maryland, US)

License © Creative Commons BY 3.0 Unported license
© Alex Malozemoff

Joint work of Apon, Daniel; Huang, Yan; Katz, Jonathan; Malozemoff, Alex

Main reference D. Apon, Y. Huang, J. Katz, A. J. Malozemoff, "Implementing Cryptographic Program Obfuscation," Cryptology ePrint Archive, Report 2014/779, 2014.

URL <https://eprint.iacr.org/2014/779>

In this talk, we give a brief overview of cryptographic program obfuscation, discussing the definitions, a high-level description of the main construction, and some performance results.

3.14 A Practical Testing Framework for Isolating Hardware Timing Channels

Sarah Meiklejohn (University College London, GB)

License © Creative Commons BY 3.0 Unported license
© Sarah Meiklejohn


This work identifies a new formal basis for hardware information flow security by providing a method to separate timing flows from other flows of information. By developing a framework for identifying these different classes of information flow at the gate level, one can either confirm or rule out the existence of such flows in a provable manner.

References

- 1 Jason Oberg, Sarah Meiklejohn, Timothy Sherwood, and Ryan Kastner. A practical testing framework for isolating hardware timing channels. In *Proceedings of the Conference on Design, Automation and Test in Europe, DATE'13*, pages 1281–1284, 2013.

3.15 Dejà Q: Using Dual Systems to Revisit q-Type Assumptions

Sarah Meiklejohn (University College London, GB)

License  Creative Commons BY 3.0 Unported license
© Sarah Meiklejohn


After more than a decade of usage, bilinear groups have established their place in the cryptographic canon by enabling the construction of many advanced cryptographic primitives. Unfortunately, this explosion in functionality has been accompanied by an analogous growth in the complexity of the assumptions used to prove security. Many of these assumptions have been gathered under the umbrella of the “uber-assumption,” yet certain classes of these assumptions—namely, q-type assumptions—are stronger and require larger parameter sizes than their static counterparts. In this paper, we show that in certain groups, many classes of q-type assumptions are in fact implied by subgroup hiding (a well-established, static assumption). Our main tool in this endeavor is the dual-system technique, as introduced by Waters in 2009. We show that q-type assumptions are implied—when instantiated in appropriate groups—solely by subgroup hiding.

References

- 1 Melissa Chase and Sarah Meiklejohn. Dejà Q: Using dual systems to revisit q-type assumptions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 622–639, 2014.

3.16 A Computational Model including Timing Attacks

Esfandiar Mohammadi (Universität des Saarlandes, DE)

License  Creative Commons BY 3.0 Unported license
© Esfandiar Mohammadi

Joint work of Backes, Michael; Manoharan, Praveen; Mohammadi, Esfandiar
Main reference M. Backes, P. Manoharan, E. Mohammadi, “TUC: Time-sensitive and Modular Analysis of Anonymous Communication,” in Proc. of the of the 27th IEEE Computer Security Foundations Symp. (CSF’14), pp. 383–397, IEEE, 2014.
URL <http://dx.doi.org/10.1109/CSF.2014.34>

Cryptographic proofs about security protocols typically abstract from timing attacks. For some security protocols, however, timing attacks constitute the most effective class of attacks, such as the anonymous communication protocol Tor. We present TUC (for Time-sensitive Universal Composability): the first universal composability framework that includes a comprehensive notion of time. TUC, in particular, includes network-based timing attacks against multi-party protocols (e.g., Tor). We, furthermore, discuss how system-level can be modelled in TUC.

3.17 Proving the Security of the Mini-APP Private Information Retrieval Protocol in EasyCrypt

Alley Stoughton (MIT Lincoln Laboratory – Lexington, US)

License © Creative Commons BY 3.0 Unported license
© Alley Stoughton

Joint work of Stoughton, Alley; Herzog, Jonathan; Varia, Mayank

Mini-APP is a simple private information retrieval (PIR) protocol involving a very simple kind of database. It's my simplification of a PIR protocol developed by cryptographers at the University of California, Irvine, as part of IARPA's APP (Advanced Privacy Protection) program. I will describe the Mini-APP protocol, define its security using the real/ideal paradigm, and give a high level explanation of how I proved its security using the EasyCrypt proof assistant.

3.18 Using the Real/Ideal Paradigm to Define Program Security

Alley Stoughton (MIT Lincoln Laboratory – Lexington, US)

License © Creative Commons BY 3.0 Unported license
© Alley Stoughton

Joint work of Stoughton, Alley; Johnson, Andrew; Beller, Samuel; Chadha, Karishma; Chen, Dennis; Foner, Kenneth; Zhivich, Michael

Main reference A. Stoughton, A. Johnson, S. Beller, K. Chadha, D. Chen, K. Foner, M. Zhivich, "You Sank My Battleship!: A Case Study in Secure Programming," in Proc. of the 9th Workshop on Programming Languages and Analysis for Security (PLAS'14), pp. 2:2–2:14, ACM, 2014.

URL <http://dx.doi.org/10.1145/2637113.2637115>

I present an example of how the real/ideal paradigm of theoretical cryptography can be used as a framework for defining the security of programs that don't necessarily involve any cryptographic operations, and in which security is enforced using programming language abstractions. Our example is the two player board game Battleship, and we'll consider an implementation of Battleship in the concurrent functional programming language Concurrent ML, giving an informal argument as to why its players are secure against possibly malicious opponents.

3.19 Enforcing Language Semantics Using Proof-Carrying Data

Eran Tromer (Tel Aviv University, IL)

License © Creative Commons BY 3.0 Unported license
© Eran Tromer

Joint work of Chong; Stephen; Tromer, Eran; Vaughan, Jeffrey A.

Main reference S. Chong, E. Tromer, J. A. Vaughan, "Enforcing Language Semantics Using Proof-Carrying Data," Cryptology ePrint Archive, Report 2013/513, 2013.

URL <https://eprint.iacr.org/2013/513>


Sound reasoning about the behavior of programs relies on program execution adhering to the language semantics. However, in a distributed computation, when a value is sent from one party to another, the receiver faces the question of whether the value is well-traced: could it have been produced by a computation that respects the language semantics? If not, then accepting the non-well-traced value may invalidate the receiver's reasoning, leading to bugs or vulnerabilities.

Proof-Carrying Data (PCD) is a recently-introduced cryptographic mechanism that allows messages in a distributed computation to be accompanied by proof that the message, and the history leading to it, complies with a specified predicate. Using PCD, a verifier can be convinced that the predicate held throughout the distributed computation, even in the presence of malicious parties, and at a verification cost that is independent of the size of the computation producing the value. Unfortunately, previous approaches to using PCD required tailoring a specialized predicate for each application, using an inconvenient formalism and with little methodological support.

We connect these two threads by introducing a novel, PCD-based approach to enforcing language semantics in distributed computations. We show how to construct an object-oriented language runtime that ensures that objects received from potentially untrusted parties are well-traced with respect to a set of class definitions. Programmers can then soundly reason about program behavior, despite values received from untrusted parties, without needing to be aware of the underlying cryptographic techniques.

3.20 Information leakage via side channels: a brief survey


Eran Tromer (Tel Aviv University, IL)

License  Creative Commons BY 3.0 Unported license
© Eran Tromer

Security of modern computer systems relies on the ability to enforce separation between mutually-untrusting processes or virtual machines. The communication between these processes/VMs is supposedly controlled by the platform (OS, VMM and hardware) according to a policy. Alas, information flow is a fickle thing: subtle and unexpected interaction between processes through the underlying hardware can convey information, and thereby violate enforcement of separation. Such “side channels” have long been the bane of secure system partitioning. In recent years, they have been recognized as especially dangerous in the age of multitenancy in cloud computing. Analogous challenges arise for corruption of computation and data by induced faults. This talk briefly surveys the challenge, and approaches to mitigating such attacks at the levels of engineering, algorithms, software and program analysis.

3.21 Verification of Quantum Crypto

Dominique Unruh (University of Tartu, EE)

License  Creative Commons BY 3.0 Unported license
© Dominique Unruh

We discussed the challenge of verifying post-quantum secure cryptography, and argue that support for such verification might be achievable at little extra cost in tools like EasyCrypt, both for the implementer of the tool, as well as for the user who writes the proof.

Follow-up discussions have already led to active and fruitful collaboration with the developers of EasyCrypt (specifically Gilles Barthe, François Dupressoir, Pierre-Yves Strub) on this topic.

Participants

- Joseph Ayo Akinyele
Johns Hopkins University –
Baltimore, US
- David Archer
Galois – Portland, US
- Manuel Barbosa
University of Minho – Braga, PT
- Gilles Barthe
IMDEA Software – Madrid, ES
- Karthikeyan Bhargavan
INRIA – Paris, FR
- Bruno Blanchet
INRIA – Paris, FR
- Dan Bogdanov
Cybernetica AS – Tartu, EE
- Niklas Büscher
TU Darmstadt, DE
- Stephen Chong
Harvard University, US
- Véronique Cortier
LORIA – Nancy, FR
- Francois Dupressoir
IMDEA Software – Madrid, ES
- Cédric Fournet
Microsoft Research UK –
Cambridge, GB
- Matthew Hammer
University of Maryland, US
- Michael Hicks
University of Maryland, US
- Catalin Hritcu
INRIA – Paris, FR
- Stefan Katzenbeisser
TU Darmstadt, DE
- Florian Kerschbaum
SAP AG – Karlsruhe, DE
- Boris Köpf
IMDEA Software – Madrid, ES
- Markulf Kohlweiss
Microsoft Research UK –
Cambridge, GB
- Sven Laur
University of Tartu, EE
- Alex Malozemoff
University of Maryland, US
- Sarah Meiklejohn
University College London, GB
- Esfandiar Mohammadi
Universität des Saarlandes, DE
- Axel Schröpfer
SAP AG – Walldorf, DE
- Alley Stoughton
MIT Lincoln Laboratory –
Lexington, US
- Eran Tromer
Tel Aviv University, IL
- Dominique Unruh
University of Tartu, EE
- Santiago Zanella-Béguelin
INRIA – Paris, FR



Programming Languages for Big Data (PlanBig)

Edited by

James Cheney¹, Torsten Grust², and Dimitrios Vytiniotis³

1 University of Edinburgh, GB, jcheney@inf.ed.ac.uk

2 Universität Tübingen, DE, torsten.grust@uni-tuebingen.de

3 Microsoft Research UK – Cambridge, GB, dimitris@microsoft.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14511 “Programming Languages for Big Data (PlanBig)”. The seminar was motivated by recent developments in programming languages, databases, machine learning, and cloud computing, and particularly by the opportunities offered by research drawing on more than one of these areas. Participants included researchers working in each of these areas and several who have previously been involved in research in the intersection of databases and programming languages. The seminar included talks, demos and free time for discussion or collaboration. This report collects the abstracts of talks and other activities, a summary of the group discussions at the seminar, and a list of outcomes.

Seminar December 15–19, 2014 – <http://www.dagstuhl.de/14511>

1998 ACM Subject Classification D.3.2 [Programming Languages]: Language Classifications – Applicative (functional) languages, H.2.3 [Database Management]: Languages – Query Languages, H.2.4 Systems - Distributed Databases, Query Processing; H.2.8 Database Applications – Data mining, Scientific databases

Keywords and phrases Programming languages, databases, data-centric computation, machine learning, cloud computing

Digital Object Identifier 10.4230/DagRep.4.12.48

Edited in cooperation with Alexander Ulrich

1 Executive Summary

James Cheney

Torsten Grust

Dimitrios Vytiniotis

License © Creative Commons BY 3.0 Unported license
© James Cheney, Torsten Grust, and Dimitrios Vytiniotis

Large-scale data-intensive computing, commonly referred to as “Big Data”, has been influenced by and can further benefit from programming languages ideas. The MapReduce programming model is an example of ideas from functional programming that has directly influenced the way distributed big data applications are written. As the volume of data has grown to require distributed processing potentially on heterogeneous hardware, there is need for effective programming models, compilation techniques or static analyses, and specialized language runtimes. The motivation for this seminar has been to bring together researchers working on foundational and applied research in programming languages but also data-intensive computing and databases, in order to identify research problems and opportunities for improving data-intensive computing.



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Programming Languages for Big Data (PlanBig), *Dagstuhl Reports*, Vol. 4, Issue 12, pp. 48–67
Editors: James Cheney, Torsten Grust, and Dimitrios Vytiniotis



Dagstuhl Reports

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

To this extent, on the database side, the seminar included participants who work on databases, query languages and relational calculi, query compilation, execution engines, distributed processing systems and networks, and foundations of databases. On the programming languages side, the seminar included participants who work on language design, integrated query languages and meta-programming, compilation, as well as semantics. There was a mix of applied and foundational talks, and the participants included people from universities as well as industrial labs and incubation projects.

The work that has been presented can be grouped in the following broad categories:

- Programming models and domain-specific programming abstractions (Cheney, Alexandrov, Vitek, Ulrich). How can data processing and query languages be integrated in general purpose languages, in type-safe ways and in ways that enable traditional optimizations and compilation techniques from database research? How can functional programming ideas such as monads and comprehensions improve the programmability of big data systems? What are some language design issues for data-intensive computations for statistics?
- Incremental data-intensive computation (Acar, Koch, Green). Programming language support and query compilation techniques for efficient incremental computation for data set or query updates. Efficient view maintainance.
- Interactive and live programming (Green, Vaz Salles, Stevenson, Binnig, Suciu). What are some challenges and techniques for interactive applications. How to improve the live programming experience of data scientists? Ways to offer data management and analytics as cloud services.
- Query compilation (Neumann, Henglein, Rompf, Ulrich). Compilation of data processing languages to finite state automata and efficient execution. Programming languages techniques, such as staging, for enabling implementors to concisely write novel compilation schemes.
- Data programming languages and semantics (Wisnesky, Vansummeren). Functorial semantics for data programming languages, but also foundations for languages for information extraction.
- Foundations of (parallel) query processing (Suciu, Neven, Hidders). Communication complexity results, program equivalence problems in relational calculi.
- Big data in/for science (Teubner, Stoyanovich, Ré). Challenges that arise in particle physics due to the volume of generated data. How we can use data to speed up new material discovery and engineering? How to use big data systems for scientific extraction and integration from many different data sources?
- Other topics: architecture and runtimes (Ahmad), coordination (Foster), language runtimes (Vytiniotis), weak consistency (Gotsman).

The seminar schedule involved three days of scheduled talks, followed by two days of free-form discussions, demos, and working groups. This report collects the abstracts of talks and demos, summaries of the group discussion sessions, and a list of outcomes resulting from the seminar.

2 Table of Contents

Executive Summary

<i>James Cheney, Torsten Grust, and Dimitrios Vytiniotis</i>	48
--	----

Overview of Talks


Self-Adjusting Computation for Dynamic and Large Data Sets <i>Umut A. Acar</i>	52
Deconstructing Big Data Stacks <i>Yanif Ahmad</i>	52
Data Analytics with Flink <i>Alexander Alexandrov</i>	53
Interactive & Visual Data Exploration <i>Carsten Binnig</i>	53
From LINQ to QDSLs <i>James Cheney</i>	53
Demo: Normalization and Query Composition in LINQ <i>James Cheney</i>	54
The Homeostatis Protocol: Avoiding Transaction Coordination Through Program Analysis <i>Nate Foster</i>	54
Weak Consistency in Cloud Storage <i>Alexey Gotsman</i>	55
Live Programming for Big Data <i>Todd J. Green</i>	55
Towards Regular Expression Processing at 1 Gbps/core <i>Fritz Henglein</i>	56
MapReduce Optimisation in the Nested Relational Calculus <i>Jan Hidders</i>	56
Incremental Computation: The Database Approach <i>Christoph Koch</i>	56
Compiling SQL Queries into Executable Code <i>Thomas Neumann</i>	57
Parallel-Correctness and Transferability for Conjunctive Queries <i>Frank Neven</i>	57
DeepDive: A Data System for Macroscopic Science <i>Christopher Ré</i>	58
An Efficient SQL to C Compiler in 500 lines of Scala <i>Tiark Rumpf</i>	58
F#3.0 – Strongly-Typed Language Support for Internet-Scale Information Sources <i>Andrew Stevenson</i>	59
(Big) Data Challenges in Materials Science and Engineering <i>Julia Stoyanovich</i>	59

Big Data Management with the Myria Cloud Service <i>Dan Suciu</i>	60
Communication Cost in Parallel Query Processing <i>Dan Suciu</i>	60
Big Data Problems in Particle Physics <i>Jens Teubner</i>	60
Query Compilation Based on the Flattening Transformation <i>Alexander Ulrich</i>	61
Spanners: A Formal Framework for Information Extraction <i>Stijn Vansummeren</i>	61
Challenges in Interactive Applications <i>Marcos Vaz Salles</i>	62
The R Project and Language <i>Jan Vitek</i>	62
Broom: Sweeping Out Garbage Collection from Big Data systems <i>Dimitrios Vytiniotis</i>	63
The Functorial Data Model <i>Ryan Wisnesky</i>	63
Working Groups	63
Outcomes	66
Participants	67

3 Overview of Talks

3.1 Self-Adjusting Computation for Dynamic and Large Data Sets

Umut A. Acar (Carnegie Mellon University – Pittsburgh, US)

License  Creative Commons BY 3.0 Unported license
© Umut A. Acar

Developing efficient and reliable software is a difficult task. Rapidly growing and dynamically changing data sets further increase complexity by making it more challenging to achieve efficiency and performance. We present practical and powerful abstractions for taming software complexity in this domain. Together with the algorithmic models and programming languages that embody them, these abstractions enable designing and developing efficient and reliable software by using high-level reasoning principles and programming techniques. As evidence for their effectiveness, we consider a broad range of benchmarks including sophisticated algorithms in geometry, machine-learning, and large data sets. On the theoretical side, we show asymptotically significant improvements in efficiency and present solutions to several open problems using the proposed techniques. On the practical side, we present programming languages, compilers, and related software systems that deliver significant improvements in performance, usually with little effort from the programmer. This talk is based on research done jointly with collaborators including A. Ahmed, G. Blelloch, M. Blume, Y. Chen, J. Dunfield, M. Fluet, M. Hammer, R. Harper, B. Hudson, R. Ley-Wild, O. Sumer, K. Tangwongsan, D. Turkoglu.

3.2 Deconstructing Big Data Stacks

Yanif Ahmad (Johns Hopkins University, US)

License  Creative Commons BY 3.0 Unported license
© Yanif Ahmad

Modern big data applications deployed in datacenter environments are complex layered software stacks that provide functionality ranging from the networking and storage hardware, to the high-level analytics logic required by the application. Today's data systems play a central role in facilitating distributed data movement, query scheduling and fault tolerance for large-scale data processing. In this talk, we survey and deconstruct the design decisions made in the modern data systems architectures commonly found in a Big Data stack. This includes the storage services provided for input data as well as large intermediate results, support for both mid-query and inter-query fault tolerance, and the architectural impact of providing low-latency results, ideally without a long tail. The systems considered include HDFS, Hadoop, Spark, Impala, Storm and briefly NoSQL and NewSQL DBMS.

3.3 Data Analytics with Flink

Alexander Alexandrov (TU Berlin, DE)

License © Creative Commons BY 3.0 Unported license
© Alexander Alexandrov

Joint work of Katsifodimos, Asterios; Alexandrov, Alexander
URL <http://flink.apache.org/>

In this demo session we give an overview of Apache Flink – an open-source system for scalable data analysis. We present Flink’s functional programming model and discuss some unique system features: (1) the approach of managing a JVM-based heap through aggressive object serialization on byte buffers, (2) the cost-based dataflow optimizer, and (3) the support for native incremental iterations and their resemblance with semi-naive Datalog evaluation.

3.4 Interactive & Visual Data Exploration

Carsten Binnig (DHBW – Mannheim, DE)

License © Creative Commons BY 3.0 Unported license
© Carsten Binnig

Joint work of Sam Zhao; Binnig, Carsten; Tim Kraska; Ugur Cetintemel; Stan Zdonik

Data-centric applications in which data scientists of varying skill levels explore large data sets are becoming more and more relevant to make sense of the data, identify interesting patterns, and bring aspects of interest into focus for further analysis. Enabling these applications with ease of use and at “human speeds” is key to democratizing data science and maximizing human productivity. As a first step towards visual interactive data exploration, we implemented a visual index for computing histograms based on a B^+ -tree. The major differences to the traditional B^+ -tree are: (1) We annotate the index nodes with count values as discussed before. (2) We offer optimized index traversal strategies for all requested bins of a histogram. (3) We use typical bin definitions of a histogram as separators for the upper levels instead of using the normal balancing rules.

3.5 From LINQ to QDSLs

James Cheney (University of Edinburgh, GB)

License © Creative Commons BY 3.0 Unported license
© James Cheney

Joint work of Cheney, James; Lindley, Sam; Wadler, Philip

Main reference J. Cheney, S. Lindley, P. Wadler, “A practical theory of language-integrated query,” in Proc. of the 18th ACM SIGPLAN Int’l Conf. on Functional programming (ICFP’13), pp. 403–416, ACM, 2013.

URL <http://dx.doi.org/10.1145/2544174.2500586>

Language-integrated query techniques ease database programming by placing queries and ordinary program code on the same level, so that the language implementation can check and coordinate communication between the host language and database. Such techniques are based on foundations developed in the 90s including comprehension syntax, normalization results for nested relational calculus, and more recent work on generalizing normalization to a higher-order setting and embedding query languages in host languages using quotation (a technique we identify as Quotation-based Domain Specific Languages, or QDSLs). In this talk I give an overview of this prior work exemplifying interaction between database and programming language research, and illustrate its impact on LINQ for F#.

3.6 Demo: Normalization and Query Composition in LINQ

James Cheney (University of Edinburgh, GB)

License © Creative Commons BY 3.0 Unported license
© James Cheney

Joint work of Cheney, James; Lindley, Sam; Wadler, Philip

Main reference J. Cheney, S. Lindley, P. Wadler, “A practical theory of language-integrated query,” in Proc. of the 18th ACM SIGPLAN Int’l Conf. on Functional programming (ICFP’13), pp. 403–416, ACM, 2013.

URL <http://dx.doi.org/10.1145/2500365.2500586>

In this demo I explained the underlying ideas of LINQ in F#, and application of recent work with Lindley and Wadler on normalization of query expressions. LINQ already performs some transformations to query expressions at run time using quotation and reflection capabilities of F#, but it has some gaps in support for queries that involve higher-order functions. Our work overcomes this limitation by providing a guarantee that all query expressions of a certain class normalize to a form that can be turned into SQL – even if the query expression makes use of lambda-abstraction and application. This has subtle implications, and allows writing efficient queries using lambda-abstraction that are not executed efficiently by the built-in F# LINQ library, and constructing queries at run time by recursion over in-memory data (illustrated by showing how XPath queries and their mapping to SQL can be defined in F# LINQ)

3.7 The Homeostatis Protocol: Avoiding Transaction Coordination Through Program Analysis

Nate Foster (Cornell University – Ithaca, US)

License © Creative Commons BY 3.0 Unported license
© Nate Foster

Joint work of Roy, Sudip; Bender, Gabriel; Kot, Lucja; Ding, Bailu; Foster, Nate; Gehrke, Johannes; Koch, Christoph

Many datastores rely on distribution and replication to achieve improved performance and fault-tolerance. But correctness of many applications depends on strong consistency properties – something that can impose substantial overheads, since it requires coordinating the behavior of multiple nodes. This work developed a new approach to achieving strong consistency in distributed systems while minimizing communication between nodes. The key insight was to allow the state of the system to be inconsistent during execution, as long as this inconsistency is bounded and does not affect transaction correctness. In contrast to previous work, our approach used program analysis to extract semantic information about permissible levels of inconsistency and is fully automated. We also developed a novel “homeostatis protocol” to allow sites to operate independently, without communicating, as long as any inconsistency is governed by appropriate treaties between the nodes. We designed mechanisms for optimizing treaties based on workload characteristics to minimize communication, built a prototype implementation, and conducted experiments to demonstrate the benefits of our approach on transactional benchmarks.

To appear in SIGMOD 2015.

3.8 Weak Consistency in Cloud Storage

Alexey Gotsman (*IMDEA Software Institute, ES*)

License © Creative Commons BY 3.0 Unported license
© Alexey Gotsman

Joint work of Bernardi, Giovanni; Cerone, Andrea; Burckhardt, Sebastian; Yang, Hongseok; Zawirski, Marek

Modern geo-replicated databases underlying large-scale Internet services guarantee immediate availability and tolerate network partitions at the expense of providing only weak forms of consistency, commonly dubbed *eventual consistency*. At the moment there is a lot of confusion about the semantics of eventual consistency, as different systems implement it with different sets of features and in subtly different forms, stated either informally or using disparate and low-level formalisms.

We address this problem by proposing a framework for formal and declarative specification of the semantics of eventually consistent systems using axioms. Our framework is fully customisable: by varying the set of axioms, we can rigorously define the semantics of systems that combine any subset of typical guarantees or features, including conflict resolution policies, session guarantees, causality guarantees, multiple consistency levels and transactions. We prove that our specifications are validated by an example abstract implementation, based on algorithms used in real-world systems. These results demonstrate that our framework provides system architects with a tool for exploring the design space, and lays the foundation for formal reasoning about eventually consistent systems.

3.9 Live Programming for Big Data

Todd J. Green (*LogicBlox – Atlanta, US*)

License © Creative Commons BY 3.0 Unported license
© Todd J. Green

Joint work of Green, Todd J.; Olteanu, Dan; Washburn, Geoffrey

We observe that the emerging category of self-service enterprise applications motivates support for “live programming” in the database, where the user’s iterative exploration of the input data triggers changes to installed application code and its output in real time. This talk discusses the technical challenges in supporting live programming in the database and presents the solution implemented in version 4.1 of the LogicBlox commercial database system. The workhorse architectural component is a novel “meta-engine” that incrementally maintains metadata representing application code, guides compilation of input application code into its internal representation in the database kernel, and orchestrates maintenance of materialized views based on those changes. Our approach mirrors LogicBlox’s declarative programming model and describes the maintenance of application code using declarative meta-rules; the meta-engine is essentially a “bootstrap” version of the database engine proper. Beyond live programming, the meta-engine turns out effective for a range of static analysis and optimization tasks, which we discuss. Outside of the database systems context, we speculate that our design may even provide a novel means of building incremental compilers for general-purpose programming languages.

3.10 Towards Regular Expression Processing at 1 Gbps/core

Fritz Henglein (University of Copenhagen, DK)

License © Creative Commons BY 3.0 Unported license
© Fritz Henglein

Joint work of Bjørn Grathwohl; Henglein, Fritz; Ulrik Rasmussen

Main reference N. B. B. Gratewohl, F. Henglein, U. T. Rasmussen, “Optimally Streaming Greedy Regular Expression Parsing,” in Proc. of the 11th Int’l Colloquium on Theoretical Aspects of Computing (ICTAC’14), LNCS, Vol. 8687, pp. 224–240, Springer, 2014.

URL http://dx.doi.org/10.1007/978-3-319-10882-7_14

URL <http://www.diku.dk/kmc>

We describe how type theory, prefix codes, nondeterministic automata, streaming and determinization to register automata yield a worst-case linear-time regular expression parser for fixed regular expressions. Early tests indicate that it operates at a sustained 100+ Mbps rate on complex regular expressions and large data sets; this seems to be significantly faster than existing tools, which operate at 2 to 20 Mbps (commodity PC). We sketch how we believe an expressive regular expression processor executing at 1 Gbps per 64-bit core can be designed and implemented, without employing machine-specific or hardware oriented tricks.

3.11 MapReduce Optimisation in the Nested Relational Calculus

Jan Hidders (TU Delft, NL)

License © Creative Commons BY 3.0 Unported license
© Jan Hidders

Joint work of Grabowski, Marek; Hidders, Jan; Sroka, Jacek; Vansummeren, Stijn

Main reference M. Grabowski, J.n Hidders, J. Sroka, “Representing mapreduce optimisations in the nested relational calculus,” in Proc. of the 29th British Nat’l Conf. on Big Data (BNCOD’13), LNCS, Vol. 7968, pp. 175–188, Springer, 2013.

URL http://dx.doi.org/10.1007/978-3-642-39467-6_17

We introduced sNRC, a variant of the Nested Relational Calculus over bags which allows heterogeneous bags and has two special operations: basic value equality and a duplicate elimination operator that selects only basic values. In this language we can readily represent a MapReduce operator, and so reasoning about equivalence of expressions in the language becomes equivalent to reasoning over MapReduce workflows over nested data. It is discussed how it might be possible to axiomatise equivalence of expressions with relatively simple equations. We also show some conjectures about the decidability of this problem for the presented fragment, and how this relates to existing results and open problems.

3.12 Incremental Computation: The Database Approach

Christoph Koch (EPFL – Lausanne, CH)

License © Creative Commons BY 3.0 Unported license
© Christoph Koch

Main reference C. Koch, Y. Ahmad, O. Kennedy, M. Nikolic, An. Nötzli, D. Lupei, A. Shaikhha, “DBToaster: higher-order delta processing for dynamic, frequently fresh views,” The VLDB Journal, 23(2):253–278, 2014.

URL <http://dx.doi.org/10.1007/s00778-013-0348-4>

In this talk, I presented the database approach to incremental computation – incremental view maintenance by compile-time query transformation. I first presented the classical approach

to incremental view maintenance using delta queries and then presented the DBToaster approach – recursive or higher-order incremental view maintenance. I also gave a demo of the DBToaster system, available at www.dbtoaster.org. Finally, I presented our recent work on higher-order incremental view maintenance for nested relational queries and the simply-typed lambda calculus, available as a preprint as [1].

References

- 1 Daniel Lupei, Christoph Koch, and Val Tannen. Incremental View Maintenance for Nested Relational Algebra. <http://arxiv.org/abs/1412.4320>, 2014.

3.13 Compiling SQL Queries into Executable Code

Thomas Neumann (TU München, DE)

License © Creative Commons BY 3.0 Unported license
© Thomas Neumann

Joint work of Neumann, Thomas; Leis, Viktor

Main reference T. Neumann, V. Leis, “Compiling Database Queries into Machine Code,” IEEE Data Engineering Bulletin, 37(1):3–11, 2014.

URL <http://sites.computer.org/debull/A14mar/p3.pdf>

On modern servers the working set of database management systems becomes more and more main memory resident. Slow disk accesses are largely avoided, and thus the in-memory processing speed of databases becomes an important factor. One very attractive approach for fast query processing is just-in-time compilation of incoming queries. By producing machine code at runtime we avoid the overhead of traditional interpretation systems, and by carefully organizing the code around register usage we minimize memory traffic and get excellent performance. In this talk we show how queries can be brought into a form suitable for efficient translation, and how the underlying code generation can be orchestrated. By carefully abstracting away the necessary plumbing infrastructure we can build a query compiler that is both maintainable and efficient. The effectiveness of the approach is demonstrated by the HyPer system that uses query compilation as its execution strategy and achieves excellent performance.

3.14 Parallel-Correctness and Transferability for Conjunctive Queries

Frank Neven (Hasselt University – Diepenbeek, BE)

License © Creative Commons BY 3.0 Unported license
© Frank Neven

Joint work of Ameloot, Tom; Geck, Gaetano; Ketsman, Bas; Neven, Frank; Schwentick, Thomas

Main reference T. J. Ameloot, G. Geck, B. Ketsman, F. Neven, T. Schwentick, “Parallel-Correctness and Transferability for Conjunctive Queries,” arXiv:1412.4030v2 [cs.DB], 2015.

URL <http://arxiv.org/abs/1412.4030v2>

A dominant cost for query evaluation in modern massively distributed systems is the number of communication rounds. For this reason, there is a growing interest in single-round multiway join algorithms where data is first reshuffled over many servers and then evaluated in a parallel but communication-free way. The reshuffling itself is specified as a distribution policy. We introduce a correctness condition, called parallel-correctness, for the evaluation of queries with respect to a distribution policy. We study the complexity of parallel-correctness for conjunctive queries as well as transferability of parallel-correctness between queries. We also investigate the complexity of transferability for certain families of distribution policies, including, for instance, the Hypercube distribution.

3.15 DeepDive: A Data System for Macroscopic Science

Christopher Ré (Stanford University, US)

License © Creative Commons BY 3.0 Unported license
© Christopher Ré

Main reference C.r Ré, A. Abbas Sadeghian, Z. Shan, J. Shin, F. Wang, S. Wu, C. Zhang, “Feature Engineering for Knowledge Base Construction,” IEEE Data Engineering Bulletin, 37(3):26–40, 2014; pre-print available as arXiv:1407.6439v3 [cs.DB].

URL <http://sites.computer.org/debull/A14sept/p26.pdf>

URL <http://arxiv.org/abs/1407.6439v3>

URL <http://DeepDive.stanford.edu>

Many pressing questions in science are macroscopic in that these questions require that a scientist integrate information from many data sources. Often, these data sources are documents that contain natural language text, tables, and figures. Such documents contain valuable information, but they are difficult for machines to understand unambiguously. This talk describes *DeepDive*, a statistical extraction and integration system to extract information from such documents. For tasks in paleobiology, *DeepDive*-based systems are surpassing human volunteers in data quantity, recall, and precision. This talk describes recent applications of *DeepDive* and *DeepDive*'s technical core. One of those core technical issues is efficient statistical inference. In particular, we describe our recent *Hogwild!* and *DimmWitted* engines that explore a fundamental tension between statistical efficiency (steps until convergence) and hardware efficiency (efficiency of each of those steps). In addition, we offer thoughts about how domain specific languages can help.

DeepDive is open source and available from <http://DeepDive.stanford.edu>.

3.16 An Efficient SQL to C Compiler in 500 lines of Scala

Tiark Rompf (Purdue University, US)

License © Creative Commons BY 3.0 Unported license
© Tiark Rompf

For hard-core systems level programming, low-level C code is still the industry standard. The drawbacks are well known: buggy systems, security vulnerabilities, poor programmer productivity, etc. Generative programming is an alternative: writing expressive high-level programs that generate fast low-level code at runtime. While many languages come with basic code generation facilities, generative programming has remained somewhat of a black art. Recent developments, however, promise to make generative programming much more accessible. This talk will provide a step-by-step introduction to the open-source LMS (Lightweight Modular Staging) framework, which brings runtime code generation and compilation to Scala programs. We will build a SQL query engine that outperforms commercial and open source database systems and consists of just about 500 lines of high-level Scala code. Along the way, we will discuss concepts such as mixed-stage data structures that contain both static and dynamic parts (e.g. static schema and dynamic values for data records) and staged interpreters which can be mechanically turned into compilers (e.g. for SQL queries or regular expressions).

3.17 F#3.0 – Strongly-Typed Language Support for Internet-Scale Information Sources

Andrew Stevenson (Queen's University – Kingston, CA)

- License** © Creative Commons BY 3.0 Unported license
© Andrew Stevenson
- Joint work of** Syme, Don; Battocchi, Keith; Takeda, Kenji; Malayeri, Donna; Fisher, Jomo; Hu, Jack; Liu, Tao; McNamara, Brian; Quirk, Daniel; Taveggia, Matteo; Chae, Wonseok; Matsveyeu, Uladzimir; Petricek, Tomas
- Main reference** D. Syme, K. Battocchi, K. Takeda, D. Malayeri, J. Fisher, J. Hu, T. Liu, B. McNamara, D. Quirk, M. Taveggia, W. Chae, U. Matsveyeu, T. Petricek, “F#3.0 – Strongly-Typed Language Support for Internet-Scale Information Sources,” Technical Report, MSR-TR-2012-101, 2012.
- URL** <http://research.microsoft.com/apps/pubs/?id=173076>

A growing trend in both the theory and practice of programming is the interaction between programming and rich information spaces. From databases to web services to the semantic web to cloud-based data, the need to integrate programming with heterogeneous, connected, richly structured, streaming and evolving information sources is ever-increasing. Most modern applications incorporate one or more external information sources as integral components. Providing strongly typed access to these sources is a key consideration for strongly-typed programming languages, to insure low impedance mismatch in information access. At this scale, information integration strategies based on library design and code generation are manual, clumsy, and do not handle the internet-scale information sources now encountered in enterprise, web and cloud environments. In this report we describe the design and implementation of the type provider mechanism in F# 3.0 and its applications to typed programming with web ontologies, web-services, systems management information, database mappings, data markets, content management systems, economic data and hosted scripting. Type soundness becomes relative to the soundness of the type providers and the schema change in information sources, but the role of types in information-rich programming tasks is massively expanded, especially through tooling that benefits from rich types in explorative programming.

3.18 (Big) Data Challenges in Materials Science and Engineering

Julia Stoyanovich (Drexel University – Philadelphia, US)

- License** © Creative Commons BY 3.0 Unported license
© Julia Stoyanovich

Materials Science and Engineering (MSE) is focused on the process of engineering matter into new and useful forms. It is a vast field that seeks to understand the properties of materials, to create materials appropriate for particular tasks, and to predict material behavior. Like many other disciplines, MSE is looking for ways to leverage data-driven approaches to make the process of scientific discovery and engineering more efficient. In this talk I present two interesting MSE use cases, outline ongoing efforts towards making MSE a data-intensive domain, and discuss ingredients of an MSE cyberinfrastructure.

3.19 Big Data Management with the Myria Cloud Service

Dan Suciu (University of Washington – Seattle, US)

License © Creative Commons BY 3.0 Unported license
© Dan Suciu

Joint work of Halperin, Daniel; de Almeida, Victor Teixeira; Choo, Lee Lee; Chu, Shumo; Koutris, Paraschos; Moritz, Dominik; Ortiz, Jennifer; Ruamviboonsuk, Vaspol; Wang, Jingjing; Whitaker, Andrew; Xu, Shengliang; Balazinska, Magdalena; Howe, Bill; Suciu, Dan

Main reference D. Halperin, V. Teixeira de Almeida, L. L. Choo, S. Chu, P. Koutris, D. Moritz, J. Ortiz, V. Ruamviboonsuk, J. Wang, A. Whitaker, S. Xu, M. Balazinska, B. Howe, D. Suciu, “Demonstration of the Myria big data management service,” in Proc. of the 2014 ACM SIGMOD Int’l Conf. on Management of Data (SIGMOD’14), pp. 881–884, ACM, 2014.

URL <http://dx.doi.org/10.1145/2588555.2594530>

URL <http://myria.cs.washington.edu/>

Myria is a novel cloud service for big data management and analytics designed to improve productivity. Myria’s goal is for users to simply upload their data and for the system to help them be self-sufficient data science experts on their data – self-serve analytics. From a web browser, Myria users can upload data, author efficient queries to process and explore the data, and debug correctness and performance issues. Myria queries are executed on a scalable, parallel cluster that uses both state-of-the-art and novel methods for distributed query processing.

3.20 Communication Cost in Parallel Query Processing

Dan Suciu (University of Washington – Seattle, US)

License © Creative Commons BY 3.0 Unported license
© Dan Suciu

Joint work of Beame, Paul; Koutris, Paraschos; Suciu, Dan

Main reference P. Beame, P. Koutris, D. Suciu, “Skew in parallel query processing,” in Proc. of the 33rd ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems (PODS’14), pp. 212–223, ACM, 2014.

URL <http://dx.doi.org/10.1145/2594538.2594558>

We study the problem of computing a conjunctive query q in parallel using p servers on a large database. We consider algorithms with one round of communication, and study the complexity of the communication. We prove matching upper and lower bounds based on the fractional edge packing of the query.

3.21 Big Data Problems in Particle Physics

Jens Teubner (TU Dortmund, DE)

License © Creative Commons BY 3.0 Unported license
© Jens Teubner

Joint work of Teubner, Jens; Spaan, Bernhard

The Large Hadron Collider at CERN is often cited as a source of extremely large data volumes, or “Big Data”. The talk gives a brief intuition of the type of experiments that are being ran at CERN (specifically the LHCb sub-project) and I will show what types of data are being produced and how they are being accessed by physical analyses. I will sketch my vision on how database-oriented techniques could be used to allow for more efficient data analysis and – as a consequence – to improve the insights that can be gained from the experimental data.

3.22 Query Compilation Based on the Flattening Transformation

Alexander Ulrich (*Universität Tübingen, DE*)

License © Creative Commons BY 3.0 Unported license
© Alexander Ulrich

Main reference A. Ulrich, T. Grust, “The Flatter, the Better – Query Compilation Based on the Flattening Transformation,” to appear in Proc. of the 34th ACM SIGMOD Int’l Conf. on the Management of Data (SIGMOD’15).

We tackle the problem of supporting an expressive, fully compositional list-based query language that allows nested results efficiently on off-the-shelf relational query engines. Query formulation is centered around comprehensions and a rich set of order-aware combinators including grouping, aggregation and sorting. This query language provides a basis for the construction of language-integrated query systems that seamlessly embed querying capabilities into functional programming languages. In this talk, we sketch the internals of a query compiler centered around the *flattening transformation*, a program transformation originally conceived to support nested data parallelism on vector processors. Adapted to query compilation, the flattening-based approach shreds nested queries into a small, statically determined number of efficient relational queries. In contrast to previous work, flattening-based query compilation (a) consists of a composition of simple steps that build on previous work and are easy to reason about (b) supports ordered collections and operations like aggregation, grouping and sorting and (c) produces efficient code.

In addition, we demonstrate Database-Supported Haskell (DSH), an implementation of flattening-based query shredding. DSH is an embedded query DSL that allows to formulate complex queries in idiomatic Haskell style. DSH queries are constructed from (higher-order) combinators and comprehensions, support abstraction over sub-queries and are subject to the same static typing discipline as other parts of a Haskell program. DSH compiles such queries with nested results into a bundle of efficient flat queries for off-the-shelf relational query engines.

3.23 Spanners: A Formal Framework for Information Extraction

Stijn Vansummeren (*University of Brussels, BE*)

License © Creative Commons BY 3.0 Unported license
© Stijn Vansummeren

Joint work of Fagin, Ron; Kimelfeld, Benny; Reiss, Frederick; Vansummeren, Stijn

An intrinsic part of information extraction is the creation and manipulation of relations extracted from text. In this talk, we present a foundational framework where the central construct is what we call a spanner. A spanner maps an input string into relations over the spans (intervals specified by bounding indices) of the string. The focus of this presentation is on the representation of spanners. Conceptually, there are two kinds of such representations. Spanners defined in a primitive representation extract relations directly from the input string; those defined in an algebra apply algebraic operations to the primitively represented spanners. This framework is driven by SystemT, an IBM commercial product for text analysis, where the primitive representation is that of regular expressions with capture variables. We define additional types of primitive spanner representations by means of two kinds of automata that assign spans to variables. We prove that the first kind has the same expressive power as regular expressions with capture variables; the second kind expresses precisely the algebra of the regular spanners – the closure of the first kind under standard relational operators.

The core spanners extend the regular ones by string-equality selection (an extension used in SystemT). We give some fundamental results on the expressiveness of regular and core spanners.

3.24 Challenges in Interactive Applications

Marcos Vaz Salles (University of Copenhagen, DK)

License © Creative Commons BY 3.0 Unported license
© Marcos Vaz Salles

Joint work of Vaz Salles, Marcos; Kefaloukos, Pimin Konstantin; Zachariasen, Martin

Main reference P. K. Kefaloukos, M. A. Vaz Salles, M. Zachariasen, “Declarative cartography: In-database map generalization of geospatial datasets,” in Proc. of the 2014 IEEE 30th Int’l Conf. on Data Engineering (ICDE’14), pp. 1024–1035, IEEE, 2014.

URL <http://dx.doi.org/10.1109/ICDE.2014.6816720>

Interactive applications, such as data visualizations and maps, computer games and simulations, or in-memory transactional and analytics systems, are becoming ever more pervasive and important to our society. In this talk, we describe lessons learned and challenges emerging from our research with these applications. First, we explore the challenge of declarative pre-computation of complex data transformations in these applications, discussing an example of selecting data for zoomable maps [1]. Second, we discuss the challenge of performance visibility in programming models for online computations, suggesting a way to revisit the transaction model for this goal [2].

References

- 1 Pimin Konstantin Kefaloukos, Marcos Vaz Salles, and Martin Zachariasen. *Declarative Cartography: In-Database Map Generalization of Geospatial Datasets*. Proc. ICDE 2014, Chicago, Illinois, USA, 2014.
- 2 Vivek Shah. *Transactional Partitioning: A New Abstraction for Main-Memory Databases*. VLDB PhD Workshop, Hangzhou, China, 2014. Best paper runner-up.

3.25 The R Project and Language

Jan Vitek (Northeastern University – Boston, US)

License © Creative Commons BY 3.0 Unported license
© Jan Vitek

URL <http://www.r-project.org>

Jan introduced the seminar attendees to the R project for statistical computing and the associated R scripting language. Through a series of live examples, from simple and obvious to quirky and outright surprising, Jan demonstrated relevant bits of the R language semantics. The discussion with the audience had a particular focus on R’s family of collection data types (vectors, matrices, arrays, lists, factors, and data frames). Issues of R’s interpreted execution model and the possibility of compiling R code were brought up later in the seminar.

Jan maintains his collection *AllR* of R-related implementation projects on GitHub: <https://github.com/allr/>.

3.26 Broom: Sweeping Out Garbage Collection from Big Data systems

Dimitrios Vytiniotis (Microsoft Research UK – Cambridge, GB)

License © Creative Commons BY 3.0 Unported license
© Dimitrios Vytiniotis

Many popular systems for processing “big data” are implemented in high-level programming languages with automatic memory management via garbage collection (GC). However, high object churn and large heap sizes put severe strain on the garbage collector. As a result, applications underperform significantly: GC increases the runtime of typical data processing tasks by up to 40%. We propose to use region-based memory management instead of GC in distributed data processing systems. In these systems, many objects have clearly defined lifetimes. It is natural to allocate these objects in fate-sharing regions, obviating the need to scan a large heap. Regions can be memory-safe and could be inferred automatically. Our initial results show that region-based memory management reduces emulated Naiad vertex runtime by 34% for typical data analytics jobs.

3.27 The Functorial Data Model

Ryan Wisnesky (MIT – Cambridge, US)

License © Creative Commons BY 3.0 Unported license
© Ryan Wisnesky

Joint work of Wisnesky, Ryan; Spivak, David

Main reference D. I. Spivak, R. Wisnesky, “Relational Foundations for Functorial Data Migration,” arXiv:1212.5303v5 [cs.DB], 2014.

URL <http://arxiv.org/abs/1212.5303v5>

We study the data transformation capabilities associated with schemas that are presented by directed multi-graphs and path equations. Unlike most approaches which treat graph-based schemas as abbreviations for relational schemas, we treat graph-based schemas as categories. A schema S is a finitely-presented category, and the collection of all S -instances forms a category, $S\text{-inst}$. A functor F between schemas S and T , which can be generated from a visual mapping between graphs, induces three adjoint data migration functors, $\Sigma_F : S\text{-inst} \rightarrow T\text{-inst}$, $\Pi_F : S\text{-inst} \rightarrow T\text{-inst}$, and $\Delta_F : T\text{-inst} \rightarrow S\text{-inst}$. We present an algebraic query language FQL based on these functors, prove that FQL is closed under composition, prove that FQL can be implemented with the select-project-product-union relational algebra (SPCU) extended with a key-generation operation, and prove that SPCU can be implemented with FQL.

4 Working Groups

The participants expressed a clear preference to avoid splitting into smaller groups to have discussions; instead, on Thursday and Friday there were plenary discussions in the main seminar room.

“Standard” intermediate language for data-centric programming

There are now lots of “programming languages for big data”, exhibiting signs of convergent evolution, with similar primitives (usually starting with some variation on map and reduce operations). Nevertheless, most such languages seem not to be well-informed by principles of programming language design, or at least, these appear to be afterthoughts. One discussion session considered the question whether these efforts are now stable enough that there is a case for a community-led “standard” – drawing inspiration from the lazy functional programming community, which consolidated its effort behind a single language design (Haskell) after a number of exploratory language designs gained momentum in the 1980s and early 1990s.

There was an extensive discussion of what this would mean, with different participants taking different views of what a “standard” would mean and what its benefits would be. One question raised was the community that such a standard would serve – would it serve PL researchers (as a “standard calculus” for language-based work on big data / data-centric computation)? Would it serve system developers (as an API) or users (as a standard surface language)? Another concern raised was that industry tends to view academic work as irrelevant due to limited scale – would this limit the value of a standard language model?

One participant mentioned recent experience with eventual consistency: after an initial burst of enthusiasm, industry seems to be reverting to stronger consistency models and tested higher-level abstractions such as transactions. Thus, it may be premature to try to consolidate effort on language designs/calculi for dealing with big data, as work in this area may still be at an experimental stage and may be at risk of abandonment if its value is not realized soon.

At a more concrete level, participants discussed what kind of standard would be of value for their research. The lambda-calculus was cited as a (widely successful) example of a “standard” formalism that programming languages researchers use as a starting point for understanding and formalizing language design ideas, abstracting away somewhat from the full complexity of implemented languages. By analogy, a calculus that plays a similar role for cloud computing, MapReduce systems, or multicore CPU or GPU code could be valuable (it should be noted that there are already some such proposals). It might be a good idea to take experience from the OpenFlow standard in software-defined networking into account; OpenFlow was established by an industry consortium but has enabled programming languages and systems researchers to work to a common interface. Likewise, formalisms such as the relational calculus/algebra (and formal standards such as SQL) have played a similar role in the database community for decades.

An interesting issue for a proposed “standard model” is that of cost modeling: a calculus or language that attempts to abstract away from the implementation details risks abstracting away the computational costs as well, so there is a tension between abstraction/portability and performance transparency/scalability. A standard model that is *operationally transparent* would be valuable for parallel or distributed computing (but there was no clear consensus on what this would mean). It would be desirable for such a model to give an explicit account of physical properties or distances between components in the system to avoid cost-opacity. Cellular automata models were mentioned as an example of how to do this but it was argued that they are too low-level. The Delite system was also mentioned as an example providing a set of high-level operators that can be mapped to different execution architectures; it is higher-level than real hardware or systems and needs to be mapped to abstract machines that model the underlying hardware well. A standard formalism might need to handle multiple layers of abstraction (by analogy with relational query optimization with its logical, physical and run-time layers). Something that is “good enough” for typical uses and portable might

be the best tradeoff (analogously to C which is not perfect but represents a workable tradeoff between abstraction and performance).

In addition, there was a short side-discussion about the desirability of benchmarking and diversity clusters for the evaluation of “big data” systems (and language techniques for them). This would aid performance tuning and portability. The Stabilizer system from the University of Massachusetts was mentioned as an example of this. The general topic of reproducibility for computer science/systems research was also mentioned (and it was pointed out that this is currently receiving attention from several quarters).

Community-building

Another topic that was discussed was the need for, and options for, building a community to improve communication among and interaction between communities relevant to the topics of the seminar. There seemed to be consensus that it would be beneficial to encourage community-building in this area. Some participants expressed concern that existing workshops seem to be diminishing in popularity and value, while it is at least possible (sometimes with greater effort) to publish work with (for example) a significant DB component in PL venues or vice-versa. Others expressed the opinion that workshops are no longer as worthwhile and a lighter-weight approach such as Dagstuhl-like events every 2 years or so is preferable. This approach, however, has the disadvantage that it limits participation to those whom the organizers can identify well in advance of the event, so may limit diversity and community growth.

One concrete option that was discussed was the possibility of organizing a new conference (rather than workshop) on “data-centric computing” to encourage work and cross-fertilization between PL and systems/databases/machine learning. The pros and cons of this strategy were discussed. On the one hand, it was recognized that this would require buy-in from “big names” / thought leaders (beyond the participants in the Dagstuhl seminar). Another potential challenge was the need to encourage significant industry participation, which could impose constraints on logistics or venues. On the other hand, participants cited recent experience with new workshops on hot topics such as USENIX HotCloud and HotSDN workshops, the ACM Symposium on Cloud Computing, which has grown rapidly to an independent event since its inception in 2010.

Overall, it was recognized that a new venue might be feasible but a strong scientific case (going beyond identifying the shortcomings of existing venues) needs to be made, in terms of increased benefit to participants and better science. One participant (Umut Acar) volunteered to coordinate subsequent discussion of the idea of a new “data-centric computation” conference. Establishing such a new conference may be difficult and so experience with DBPL 2015 may help build the case for this.

DBPL

The final morning of the seminar saw a discussion of the future of DBPL, the International Symposium on Database Programming Languages, which has been running biennially since 1987. Recent occurrences of DBPL in 2011 and 2013 had seen a decrease in submissions and participation compared to previous years. Members of both events PC chair teams were present and as of the week of the seminar its status in 2015 was unclear. There was some

feeling that DBPL may have run its course, but also that it would be a shame for the series to end when events such as this Dagstuhl seminar showcase such a range of relevant activity. It was felt that this question was largely orthogonal to the question of developing a new conference venue (though a strong showing for DBPL in 2015 might contribute to a case for the “data-centric computation” conference idea).

DBPL had been co-located with VLDB (a major database conference, which seminar participants from the DB community would typically attend) until 2013, and since 2009 took place as a one-day workshop. In 2015, VLDB takes place the same week as ICFP, a major PL conference (and one which a number of seminar participants would normally attend). This clash highlighted a problem with DBPL’s recent role as a “VLDB workshop”: even in years when there is no clash with other events, participants from outside the DB community may find it difficult to justify the time/expense of attending another conference (or of just attending one day of an event they would otherwise not attend).

A number of alternatives were discussed, including the possibility of co-locating DBPL with ICFP in 2015, holding it as a stand-alone event (close in time/space to VLDB or ICFP but not formally affiliated with either), or seeking another co-location option. The possibility of co-locating with SPLASH 2015 (an umbrella PL conference including OOPSLA and several other events) was also raised, but did not seem to generate much enthusiasm at the seminar. An alternative proposal was considered, which attracted considerable support: to try to hold DBPL at *both* venues, with a video link connecting speakers and audience members at VLDB (in Hawaii) and ICFP (in Vancouver). Although this arrangement was recognized to have disadvantages (e.g. the inability to talk to speakers or other participants informally outside the conference room), participants felt that it offered the most promising route if it could be done. Of approximately 20 participants present in the discussion, a clear majority indicated willingness to either help organize or participate in/submit to DBPL if it were held in 2015.

5 Outcomes

- Umut Acar agreed to coordinate a discussion of the possibility of starting a “data-centric computation” conference.
- James Cheney started a “data-centric programming languages” mailing list, invited Dagstuhl participants to join and subsequently advertised it on relevant mailing lists such as TYPES and DBworld. The list currently has over 120 members.
- Fritz Henglein and Torsten Grust agreed to investigate the possibility of DBPL taking place “virtually” at two locations, with VLDB in Hawaii and ICFP in Vancouver connected by a video link. This turned out to be infeasible due to the high up-front cost of the link.
- Based on a straw poll conducted with Dagstuhl participants it was decided to approach the SPLASH 2015 organizers to see if DBPL could be co-located there. The SPLASH organizers were willing to approve this without going through the formal workshop application process. The two co-chairs are James Cheney and Thomas Neumann and 6 of the 10 PC members were participants in the Dagstuhl seminar.

Participants

- Umut A. Acar
Carnegie Mellon University – Pittsburgh, US
- Yanif Ahmad
Johns Hopkins University – Baltimore, US
- Alexander Alexandrov
TU Berlin, DE
- Carsten Binnig
DHBW – Mannheim, DE
- Giuseppe Castagna
University Paris-Diderot, FR
- James Cheney
University of Edinburgh, GB
- Laurent Daynès
Oracle Corporation, FR
- Nate Foster
Cornell University – Ithaca, US
- Pierre Geneves
INRIA – Grenoble, FR
- Alexey Gotsman
IMDEA Software – Madrid, ES
- Todd J. Green
LogicBlox – Atlanta, US
- Torsten Grust
Universität Tübingen, DE
- Fritz Henglein
University of Copenhagen, DK
- Jan Hidders
TU Delft, NL
- Christoph Koch
EPFL – Lausanne, CH
- Tim Kraska
Brown University, US
- Sam Lindley
University of Edinburgh, GB
- Todd Mytkowicz
Microsoft Corp. – Redmond, US
- Thomas Neumann
TU München, DE
- Frank Neven
Hasselt Univ. – Diepenbeek, BE
- Ryan R. Newton
Indiana University – Bloomington, US
- Kim Nguyen
University Paris-Sud – Gif sur Yvette, FR
- Klaus Ostermann
Universität Tübingen, DE
- Christopher Ré
Stanford University, US
- Tiark Rompf
Purdue University, US
- Andrew Stevenson
Queen’s Univ. – Kingston, CA
- Julia Stoyanovich
Drexel Univ. – Philadelphia, US
- Dan Suciu
University of Washington – Seattle, US
- Jens Teubner
TU Dortmund, DE
- Alexander Ulrich
Universität Tübingen, DE
- Jan Van den Bussche
Hasselt Univ. – Diepenbeek, BE
- Stijn Vansummeren
Université Libre de Bruxelles, BE
- Marcos Vaz Salles
University of Copenhagen, DK
- Jan Vitek
Northeastern University – Boston, US
- Dimitrios Vytiniotis
Microsoft Research UK – Cambridge, GB
- Ryan Wisnesky
MIT – Cambridge, US



Report from Dagstuhl Seminar 14512

Collective Adaptive Systems: Qualitative and Quantitative Modelling and Analysis

Edited by

Jane Hillston¹, Jeremy Pitt², Martin Wirsing³, and Franco Zambonelli⁴

1 University of Edinburgh, GB, Jane.Hillston@ed.ac.uk

2 Imperial College London, GB, j.pitt@imperial.ac.uk

3 LMU München, DE, wirsing@lmu.de

4 University of Modena, IT, franco.zambonelli@unimore.it

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14512 “Collective Adaptive Systems: Qualitative and Quantitative Modelling and Analysis”. Besides presentations on current work in the area, the seminar focused on the following topics: (i) Modelling techniques and languages for collective adaptive systems based on the above formalisms. (ii) Verification of collective adaptive systems. (iii) Humans-in-the-loop in collective adaptive systems.

Seminar December 14–19, 2014 – <http://www.dagstuhl.de/14512>

1998 ACM Subject Classification C.2.4 Distributed Systems, D.2 Software Engineering, D.2.4 Software/Program Verification, H.1.2 User/Machine Systems

Keywords and phrases Collective Adaptive Systems, Qualitative and Quantitative Modelling and Analysis, Verification, Humans-In-The-Loop

Digital Object Identifier 10.4230/DagRep.4.12.68

Edited in cooperation with Lenz Belzner

1 Executive Summary

Jane Hillston

Jeremy Pitt

Martin Wirsing

Franco Zambonelli

License © Creative Commons BY 3.0 Unported license
© Jane Hillston, Jeremy Pitt, Martin Wirsing, and Franco Zambonelli

Modern systems are often structured as complex, multi-layered networks of interconnected parts, where different layers interact and influence each other in intricate and sometimes unforeseen ways. It is infeasible for human operators to constantly monitor these interactions and to adjust the system to cope with unexpected circumstances; instead systems have to adapt autonomously to dynamically changing situations while still respecting their design constraints and requirements. Because of the distributed and decentralized nature of modern systems, this usually has to be achieved by collective adaptation of the nodes comprising the system. In open systems exhibiting collective adaptation, unforeseen events and properties can arise, e.g. as side effects of the interaction of the components or the environment. Modelling and engineering collective adaptive systems (CAS) has to take into account such “emergent” properties in addition to satisfying functional and quantitative requirements.



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Collective Adaptive Systems: Qualitative and Quantitative Modelling and Analysis, *Dagstuhl Reports*, Vol. 4, Issue 12, pp. 68–113

Editors: Jane Hillston, Jeremy Pitt, Martin Wirsing, and Franco Zambonelli



DAGSTUHL
REPORTS Dagstuhl Reports

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

Finding ways to understand and design CAS, and to predict their behaviour, is a difficult but important endeavour. One goal of this seminar was to investigate techniques for modelling and analysing systems that adapt collectively to dynamically changing environment conditions and requirements. In many cases, these models and analysis techniques should not only capture qualitative properties of the system, such as absence of deadlocks, they should also be able to express quantitative properties such as quality of service.

Research on CAS builds on and integrates previous research efforts from several areas:

- Formal foundations and modelling techniques for concurrent systems deal with problems such as enabling and limiting concurrency, access to shared resources, avoidance of anomalies, communication between processes, and estimation of performance.
- Analysis of concurrent systems typically exploits such notions as bisimilarity of different processes or reasons on stochastic properties of systems consisting of many equivalent processes.
- The area of adaptive systems also investigates systems consisting of interacting entities, but is more concerned with the reaction of whole systems or individual actors in a system to a changing environment.

An important aim of this seminar was to combine research from concurrent systems with results from the adaptive systems community in order to develop formalisms for specifying CAS, to increase the scalability of qualitative and quantitative modelling and analysis techniques to large systems, and to apply them to systems that dynamically change their structure or adapt to novel situations.

The seminar was organised with a mixture of talks and working group sessions which facilitated more in-depth discussions and exploration of topics. In this report we include the abstracts of a selection of the presented talks, and three longer contributions compiled after the meeting which seek to reflect the activities of the working groups. The first group, considering modelling, specification and programming for CAS, start their presentation with brief descriptions of four diverse applications developed on the basis of CAS, ranging from national level power management to personal wearable devices. To complement this identification of application domains, the group also catalogued common and contrasting features that can be found in CAS. This consideration highlights the role of physical space in all the considered domains and the urgent need to develop modelling and analysis techniques which reflect this central role played by space. This was key amongst a number of challenges identified by the group in their conclusions. Spatio-temporal aspects were also identified as a key challenge by the second working group who considered verification of CAS. The report from this group outlines the role of verification within the design and management of CAS ranging from seeking to guarantee global emergent behaviour from local specifications to using online verification to drive adaptation. Two specific challenges were explored in more detail, namely handling the inherent uncertainty in CAS, and specification and verification of spatial properties of systems composed of self-organising patterns. The third working group focused on the issues that arise from the recognition that some of the entities within a CAS may be humans and outside technological control, i.e. the design of socio-technical systems. A number of different scenarios are provided to illustrate the difference between socio-technical CAS and 'technical' CAS, and the human factors which must be taken into account. To remediate some of the problems identified, the group propose the idea of a general intervention framework, based around the 3I life-cycle – inspection-innovation-intervention. It was foreseen that intervention would be achieved by shaping mechanisms, and the report goes on to describe some possible shaping mechanisms which were considered. To conclude a number of research challenges are discussed.

2 Table of Contents

Executive Summary

Jane Hillston, Jeremy Pitt, Martin Wirsing, and Franco Zambonelli 68

Overview of Talks

Creating Predictable Collective Behaviors with Aggregate Programming

Jacob Beal 72

Algebraic Reinforcement Learning

Lenz Belzner 72

Dynamic change of collaboration patterns: motivations and perspectives

Giacomo Cabri 73

A formal approach to autonomic systems programming: The SCEL Language

Rocco De Nicola 73

Discrete Time Markovian Agents

Marco Gribaudo 74

On bootstrapping sensori-motor patterns for a constructivist learning system in continuous environments

Salima Hassas 74

Challenges for Quantitative Analysis of Collective Adaptive Systems

Jane Hillston 75

Role-based Adaptation

Annabelle Klarl 75

Diversity, Heterogeneity and Dynamics in Collective Systems

Peter Lewis 76

Modelling Collective Adaptive Systems in CARMA

Michele Loreti 76

Stochastic Coordination in CAS: Expressiveness and Predictability

Stefano Mariani 76

On-the-fly Fast Mean Field Model Checking for Collective Adaptive Systems

Mieke Massink 77

Declarative vs. Procedural Approach for SCSP with an Application to an E-mobility Optimization Problem

Ugo Montanari 78

Procedural Justice and ‘Fitness for Purpose’ of Self-Organising Electronic Institutions

Jeremy Pitt 78

LollyScript, a concurrent programming language to ensure that promises are kept

Christophe Scholliers 79

Testing as a useful complement to verification of SOAS?

Hella Seebach 79

How Collective and Adaptive our CAS are?


Nikola Serbedzija 80

Three Behavioural Equivalences for Chemical Reaction Networks <i>Mirco Tribastone</i>	80
Engineering Autonomous Ensembles <i>Martin Wirsing</i>	81
Smart Cities as Heterogeneous Superorganisms: Scenarios and Challenges <i>Franco Zambonelli</i>	81
Working Group Reports	
Modelling, Specification, and Programming for Collective Adaptive Systems <i>Hella Seebach, Lenz Belzner, Marco Gribaudo, Anabelle Klarl, Michele Loreti, Ugo Montanari, Laura Nenzi, Rocco De Nicola, Christophe Scholliers, Petr Tuma, and Martin Wirsing</i>	82
Verification of CAS <i>Luca Bortolussi, Giacomo Cabri, Giovanna Di Marzo Serugendo, Vashti Galpin, Jane Hillston, Roberta Lanciani, Mieke Massink, Mirco Tribastone, and Danny Weyns</i>	91
Humans-in-the-Loop in Collective Adaptive Systems <i>Jake Beal, Peter Lewis, Stefano Mariani, Jeremy Pitt, Nikola Serbedzija, and Franco Zambonelli</i>	102
Participants	113

3 Overview of Talks

3.1 Creating Predictable Collective Behaviors with Aggregate Programming


Jacob Beal (BBN Technologies – Cambridge, US)

License  Creative Commons BY 3.0 Unported license
© Jacob Beal

Practical collective adaptive systems typically comprise many different interacting elements, with a great degree of heterogeneity in the activities required of any given element and the capabilities of different elements. This tends to make engineering such systems extremely difficult, and proving properties about the engineered systems effectively impossible. Recently developed methods in aggregate programming, however, offer the possibility of creating “operator algebras” in any collective adaptive system created using a fairly general API is proved by construction to have desirable adaptive properties such as self-stabilization, scalability, and toleration of network perturbation. These methods thus offer a path to engineering systems that exhibit implicit, safe, and ubiquitous adaptivity.

3.2 Algebraic Reinforcement Learning

Lenz Belzner (LMU München, DE)

License  Creative Commons BY 3.0 Unported license
© Lenz Belzner

This talk expands on support for decision making in autonomous adaptive systems by identifying proper qualitative state abstractions through quantitative (i.e. statistical) analysis of environment data sampled at system runtime. The TG relational reinforcement learning algorithm [1] learns relational decision trees by statistical evaluation of runtime data. Here, qualitative state abstractions to be analyzed statistically are specified manually and a-priori.

The talk introduces a quantifiable metric for adaptation in the context of learning systems to allow for quantitative evaluation of adaptation. By identifying operators for model modification and evaluation, the relation of relational reinforcement learning to evolutionary programming is shown. An approach for automatic extraction of relevant qualitative abstractions via algebraic term generalization with the ACUOS system [2] is presented.

References

- 1 Driessens, K., Ramon, J., Blockeel, H.: *Speeding up relational reinforcement learning through the use of an incremental first order decision tree learner*. In: Machine Learning: ECML 2001. Springer (2001) 97–108
- 2 Alpuente, M., Escobar, S., Espert, J., Meseguer, J.: *Acuos: A system for modular acu generalization with subtyping and inheritance*. In Ferm, E., Leite, J., eds.: Logics in Artificial Intelligence. Volume 8761 of Lecture Notes in Computer Science. Springer International Publishing (2014) 573–581

3.3 Dynamic change of collaboration patterns: motivations and perspectives

Giacomo Cabri (University of Modena, IT)

License © Creative Commons BY 3.0 Unported license
© Giacomo Cabri

Joint work of Cabri, Giacomo; Capodieci, Nicola; Zambonelli, Franco; Puviani, Mariachiara
Main reference G. Cabri, N. Capodieci, “Runtime Change of Collaboration Patterns in Autonomic Systems: Motivations and Perspectives,” in Proc. of the 2013 27th Int’l Conf. on Advanced Information Networking and Applications Workshops (WAINA’13), pp. 1038–1043, IEEE, 2013.
URL <http://dx.doi.org/10.1109/WAINA.2013.82>

Today’s complex distributed systems must adapt to the unexpected execution conditions they face, in an autonomous way. This requires not only the adaptation feature at component level, but also the capability of adapting at the system level, modifying the collaboration pattern among components [3]. In this talk I will introduce the scenario, motivate the need for collaboration pattern changes at runtime [1], and propose some approaches to enact them, one based on formal models, one based on roles [4], and one bio-inspired [2].

References

- 1 Giacomo Cabri and Nicola Capodieci. *Runtime Change of Collaboration Patterns in Autonomic Systems: Motivations and Perspectives*. In *Advanced Information Networking and Applications Workshops (WAINA)*, 2013 27th International Conference on. IEEE, IEEE, Piscataway, NJ, USA, 1038–1043.
- 2 Nicola Capodieci, Emma Hart and Giacomo Cabri. *Artificial Immune System driven evolution in Swarm Chemistry*. Proceedings of The Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems SASO 2014.
- 3 Franco Zambonelli, Nicola Bicocchi, Giacomo Cabri, Letizia Leonardi and Mariachiara Puviani. *On self-adaptation, self-expression, and self-awareness in autonomic service component ensembles*. In *Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*, 2011 Fifth IEEE Conference on. IEEE, IEEE, Piscataway, NJ, USA, 108–113.
- 4 Mariachiara Puviani and Giacomo Cabri and Letizia Leonardi. *Enabling Self-expression: the Use of Roles to Dynamically Change Adaptation Patterns*. Proceedings of the Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshop(SASOW 14), London, UK, 8-12 September 2014.

3.4 A formal approach to autonomic systems programming: The SCEL Language

Rocco De Nicola (IMT Lucca, IT)

License © Creative Commons BY 3.0 Unported license
© Rocco De Nicola

The autonomic computing paradigm has been proposed to cope with size, complexity and dynamism of contemporary software-intensive systems. The challenge for language designers is to devise appropriate abstractions and linguistic primitives to deal with the large dimension of systems, and with their need to adapt to the changes of the working environment and to the evolving requirements. We introduced a set of programming abstractions that permit to represent behaviours, knowledge and aggregations according to specific policies, and to support programming context-awareness, self-awareness and adaptation. Based on these abstractions, we described SCEL (Software Component Ensemble Language), a kernel language whose

solid semantic foundations lay also the basis for formal reasoning on autonomic systems behaviour. To show expressiveness and effectiveness of SCEL's design, we presented a Java implementation of the proposed abstractions and showed how it has been exploited for programming a robotics scenario used as a running example for describing features and potentials of our approach.

3.5 Discrete Time Markovian Agents

Marco Gribaudo (Politecnico di Milano, IT)

License © Creative Commons BY 3.0 Unported license
© Marco Gribaudo

Markovian Agents is a formalism that has been used to model large systems composed by interacting entities. Agents interact using a mechanism based on what is called “Induction”: the states in which neighbor agents are, influences the transition rates. The concept is quite natural in continuous time, and it is supported by strong theory coming from mean-field analysis and spatial Poisson processes. The transition to discrete time however is not trivial, and opens new questions and new possibilities.

3.6 On bootstrapping sensori-motor patterns for a constructivist learning system in continuous environments

Salima Hassas (University Claude Bernard – Lyon, FR)

License © Creative Commons BY 3.0 Unported license
© Salima Hassas

Joint work of Mazac, Sébastien; Armetta, Frédéric; Hassas, Salima
Main reference S. Mazac, F. Armetta, S. Hassas, “On Bootstrapping Sensori-Motor Patterns for a Constructivist Learning System in Continuous Environments,” in Proc. of the 14th Int'l Conf. on the Synthesis and Simulation of Living Systems (ALIFE'14), pp. 160–167, MIT Press, 2014.
URL <http://dx.doi.org/10.7551/978-0-262-32621-6-ch028>

The theory of cognitive development from Jean Piaget (1923) is a constructivist perspective of learning that has substantially influenced cognitive science domain. Within AI, lots of works have tried to take inspiration from this paradigm since the beginning of the discipline. Indeed it seems that constructivism is a possible trail in order to overcome the limitations of classical techniques stemming from cognitivism or connectionism and create autonomous agents, fitted with strong adaptation ability within their environment, modelled on biological organisms. Potential applications concern intelligent agents in interaction with a complex environment, with objectives that cannot be predefined. Like robotics, Ambient Intelligence (AmI) is a rich and ambitious paradigm that represents a high complexity challenge for AI. In particular, as a part of constructivist theory, the agent has to build a representation of the world that relies on the learning of sensori-motor patterns starting from its own experience only. This step is difficult to set up for systems in continuous environments, using raw data from sensors without a priori modelling. With the use of multi-agent systems, we investigate the development of new techniques in order to adapt constructivist approach of learning on actual cases. Therefore, we use ambient intelligence as a reference domain for the application of our approach.

3.7 Challenges for Quantitative Analysis of Collective Adaptive Systems

Jane Hillston (University of Edinburgh, GB)

License © Creative Commons BY 3.0 Unported license
© Jane Hillston

Main reference J. Hillston, “Challenges for Quantitative Analysis of Collective Adaptive Systems,” in Proc. of the 8th Int’l Symp. on Trustworthy Global Computing (TGC’13), LNCS, Vol. 8358, pp. 14–21, Springer, 2014.

URL http://dx.doi.org/10.1007/978-3-319-05119-2_2

Quantitative analysis plays an important role in the design of systems as it allows us to predict their dynamic behaviour. Thus in addition to the functional properties that can be assessed by qualitative analysis we can also investigate properties such the timeliness of response and the efficient and fair access to resources. However the scale of collective adaptive systems imposes serious challenges on the usual approaches to quantitative analysis which are based on discrete state representations. In this talk I will talk about these challenges and explain how in some circumstances making a fluid approximation of the discrete state space can be beneficial.

3.8 Role-based Adaptation

Annabelle Klarl (LMU München, DE)

License © Creative Commons BY 3.0 Unported license
© Annabelle Klarl

Joint work of Klarl, Annabelle; Hennicker, Rolf


A self-adaptive component keeps track of its individual and shared goals, perceives its internal state as well as its environment, and adapts its behavior accordingly. Based on these characteristic features, we propose a pragmatic methodology to develop self-adaptive systems from specification to design. We specify the system’s adaptation logic by adaptation automata. A design model refines the specification by adding application logic and providing an architecture. We take inspiration from the autonomic manager pattern [2] where an adaptation manager is employed on an adaptable component to control appropriate behavioral adaptation in response to observations of the environment. To realize the architecture of the autonomic manager pattern, we make use of the Helena modeling approach [1] to encapsulate the manager, sensors of the environment, and different behavioral modes of the component into roles applied to the component. The system design therefore gets structured into self-contained roles providing a clear architecture separating adaptation logic and application logic.

References

- 1 Rolf Hennicker and Annabelle Klarl, Foundations for Ensemble Modeling – The Helena Approach, in Specification, Algebra, and Software, ser. LNCS, vol. 8373. Springer, 2014, pp. 359–381.
- 2 Mariachiara Puviani, Giacomo Cabri, and Franco Zambonelli, A taxonomy of architectural patterns for self-adaptive systems, in International C* Conference on Computer Science and Software Engineering. ACM, 2013, pp. 77–85.

3.9 Diversity, Heterogeneity and Dynamics in Collective Systems

Peter Lewis (Aston University Birmingham, UK)

License  Creative Commons BY 3.0 Unported license
© Peter Lewis

Diversity plays an important role in many natural and engineered systems. In this talk, I will describe two different forms of diversity present in engineered collective systems: (i) heterogeneity (genotypic/phenotypic diversity) and (ii) dynamics (temporal diversity). I will discuss these forms of diversity through two qualitatively different case studies (smart camera networks and particle swarm optimisation). The analysis shows that both forms of diversity can be beneficial in very different problem and application domains, and can indeed impact more than the ability of the collective to adapt. I will end by raising some questions regarding how to engineer effective diversity in collective systems.

3.10 Modelling Collective Adaptive Systems in CARMA

Michele Loreti (University of Firenze, IT)

License  Creative Commons BY 3.0 Unported license
© Michele Loreti

Joint work of Bortolussi, Luca; De Nicola, Rocco; Galpin, Vashti; Gilmore, Stephen; Hillston, Jane; Latella, Diego; Loreti, Michele; Massink, Mieke

In this talk we present CARMA, a language recently defined to support specification and analysis of collective adaptive systems. CARMA is a stochastic process algebra equipped with linguistic constructs specifically developed for modelling and programming systems that can operate in open-ended and unpredictable environments. This class of systems is typically composed of a huge number of interacting agents that dynamically adjust and combine their behaviour to achieve specific goals. A CARMA model, termed a collective, consists of a set of components, each of which exhibits a set of attributes. To model dynamic aggregations, which are sometimes referred to as ensembles, CARMA provides communication primitives that are based on predicates over the exhibited attributes. These predicates are used to select the participants in a communication. Two communication mechanisms are provided in the CARMA language: multicast-based and unicast-based. In the talk, we first introduce the basic principles of CARMA and then we show how our language can be used to support specification with a simple but illustrative example of a socio-technical collective adaptive system.

3.11 Stochastic Coordination in CAS: Expressiveness and Predictability

Stefano Mariani (Università di Bologna, IT)

License  Creative Commons BY 3.0 Unported license
© Stefano Mariani

Joint work of Omicini, Andrea; Mariani, Stefano

Recognising that (i) coordination is a fundamental concern when both analysing and modelling CAS, and that (ii) CAS often exhibit stochastic behaviours, stemming from probabilistic and time-dependent local (interaction) mechanisms, in this talk we argue that (a) measuring

expressiveness of coordination languages, and (b) predicting behaviours of stochastic systems based on coordination models are two fundamental steps in the quest for designing well-engineered CAS. As a concrete ground where to or discussion, we describe some of our current works as well as our ideas for further research.

3.12 On-the-fly Fast Mean Field Model Checking for Collective Adaptive Systems

Mieke Massink (CNR – Pisa, IT)

License © Creative Commons BY 3.0 Unported license

© Mieke Massink

Joint work of Latella, Diego; Loreti, Michele; Massink, Mieke

Main reference D. Latella, M. Loreti, M. Massink, “On-the-fly fast mean-field model-checking,” in Proc. of the 8th Int’l Symp. on Trustworthy Global Computing (TGC’13), LNCS, Vol. 8358, pp. 297–314, Springer, 2014.

URL http://dx.doi.org/10.1007/978-3-319-05119-2_17

Typical self-organising collective systems consist of a large number of interacting objects that coordinate their activities in a decentralised and often implicit way. Design of such systems is challenging and requires suitable, scalable analysis tools to check properties of proposed system designs before they are put into operation. Model checking has shown to be a successful technique for the verification of distributed and concurrent systems, but in the context of collective systems we need these techniques to be highly scalable. Model checking approaches can be divided into two broad categories: global approaches that determine the set of all states in a model M that satisfy a temporal logic formula F , and local approaches in which, given a state s in M , the procedure determines whether s satisfies F . When s is a term of a process language, the model-checking procedure can be executed “on-the-fly”, driven by the syntactical structure of s . For certain classes of systems, e.g. those composed of many parallel components, the local approach is preferable because, depending on the specific property, it may be sufficient to generate and inspect only a relatively small part of the state space. Recently global stochastic model-checking approaches for collective systems have been explored, combining fast simulation and fluid approximation in a continuous time setting, for example in the work by Bortolussi and Hillston. In this presentation we explore the use of on-the-fly techniques in this direction in a discrete time setting. We first present an efficient, probabilistic, on-the-fly, PCTL model checking procedure that is parametric with respect to the semantic interpretation of the language. The procedure comprises both bounded and unbounded until modalities. The correctness of the procedure is shown and its efficiency has been explored on a number of benchmark applications in comparison with the global PCTL model checker PRISM. We then show how to instantiate the procedure with a mean field semantics to verify bounded PCTL properties of selected individuals in the context of very large systems of independent interacting objects. The asymptotic correctness of the procedure is shown and some results of the application of a prototype implementation of the FlyFast model-checker will be presented.

References

- 1 Latella, D., Loreti, M., Massink, M.: On-the-fly fast mean-field model-checking. In: Abadi, M., Lluch-Lafuente, A. (eds.) Trustworthy Global Computing – 8th Int’l Symp., TGC 2013, Buenos Aires, Argentina, August 30–31, 2013, Revised Selected Papers. LNCS, vol. 8358, pp. 297–314. Springer (2014), http://dx.doi.org/10.1007/978-3-319-05119-2_17
- 2 Latella, D., Loreti, M., Massink, M.: On-the-fly probabilistic model-checking. In: Proceedings 7th Interaction and Concurrency Experience ICE 2014. EPTCS, vol. 166 (2014)

3.13 Declarative vs. Procedural Approach for SCSP with an Application to an E-mobility Optimization Problem

Ugo Montanari (University of Pisa, IT)

License © Creative Commons BY 3.0 Unported license
© Ugo Montanari

Main reference N. Hoch, G. V. Monreale, U. Montanari, M. Sammartino, A. Tcheukam Siwe, “From Local to Global Knowledge and Back,” in M. Wirsing, M. Hölzl, N. Koch, P. Mayer (eds.), “Software Engineering for Collective Autonomic Systems – The ASCENS Approach,” LNCS, Vol. 8998, pp. 185–220, Springer, 2015.

URL http://dx.doi.org/10.1007/978-3-319-16310-9_5

Large optimization problems tend to be overly complex to solve and often a globally optimal solution may be impossible to find. For this reason specific strategies are needed to solve them. We propose an approach for the coordination of declarative knowledge – that is the exact specification of the complete optimization problem – and of procedural knowledge – that is the specific knowledge about subproblems and their, possibly approximated, solution strategies. We consider Soft Constraint Satisfaction Problems (SCSPs) and we introduce a formalism, similar to a process calculus, for their specification. Cost functions are associated to terms and form a model of such specification, where operators are interpreted as optimization steps. We compare our approach with Courcelle’s approach for efficient monadic second-order evaluations on tree composable graphs. We apply our approach to a problem studied in the ASCENS e-mobility case study, for which we provide a model in terms of cost functions. The procedural part concerns heuristic choices about which dynamic programming strategy should be employed and how different ad-hoc approximation heuristics could be applied.

3.14 Procedural Justice and ‘Fitness for Purpose’ of Self-Organising Electronic Institutions

Jeremy Pitt (Imperial College London, UK)

License © Creative Commons BY 3.0 Unported license
© Jeremy Pitt

In many multi-agent systems, it is a commonplace requirement to distribute a pool of collectivised resources amongst those agents. One way to address typical problems, like unrestrained resource access, or to ensure some desirable property, like fairness, is for the agents to mutually agree a set of rules to self-organise and self-regulate the distribution process. We propose a framework for measuring the ‘fitness for purpose’ of such a set of rules, as represented in the Event Calculus. This framework encapsulates metrics for principles of participation, transparency and balancing, as derived from various conceptions of procedural justice. We define a metric for the empowerment aspect of the participation principle, and report some experimental results which show how this metric can reveal an inherent ‘fairness’ or ‘unfairness’ in the distribution of (institutionalised) power over time, and inform decision-making or rule-adaptation accordingly. We conclude with some discussion of how procedural justice can be used for analysis of collective adaptive systems.

3.15 LollyScript, a concurrent programming language to ensure that promises are kept

Christophe Scholliers (Free University of Brussels, BE)

License  Creative Commons BY 3.0 Unported license
© Christophe Scholliers

It is difficult to reason about the synchronisation between a set of concurrently executing tasks. One of the main reasons for this complexity is that the amount of possible states a system can be in increases exponentially with the amount of concurrent tasks that are executing at the same time. Programming languages abstractions can help the programmer to prune the state space by eliminating those states that lead to inconsistencies. In this talk we will focus on the use of promises and futures to structure the asynchronous communication between two tasks. Unfortunately, in current systems the use of promises can easily lead to deadlocks. For example, the programming model can not ensure that all the promises in the system will be resolved. In this talk we present a concurrent programming language with a linear type system to statically verify the correct use of promises in concurrent programs.

3.16 Testing as a useful complement to verification of SOAS?

Hella Seebach (Universität Augsburg, DE)

License  Creative Commons BY 3.0 Unported license
© Hella Seebach

Joint work of Seebach, Hella; Nafz, Florian; Eberhardinger, Benedikt; Reif, Wolfgang

Self-organization and adaptivity are important techniques for building flexible and robust systems. Though, applying verification and testing is crucial for their acceptance. We propose techniques (software engineering guideline, coalition formation, compositional reasoning, verified result checking, etc.) for the construction and partial verification of self-organizing resource-flow systems. These techniques allow for example to reason about global properties by verifying single agent properties. In this talk, I want to discuss in which way new techniques for testing SOAS can be a complement to further extend the quality assurance of our partially verified system.

References

- 1 Hella Seebach, Florian Nafz, Jan-Philipp Steghöfer, Wolfgang Reif *How to Design and Implement Self-organising Resource-Flow Systems*. Organic Computing – A Paradigm Shift for Complex Systems, Autonomic Systems, Birkhäuser, Springer
- 2 Florian Nafz, Jan-Philipp Steghöfer, Hella Seebach, Wolfgang Reif. *Formal Modeling and Verification of Self-* Systems Based on Observer/Controller-Architectures*. Springer-Verlag, Berlin, Heidelberg, 2013
- 3 Benedikt Eberhardinger, Hella Seebach, Alexander Knapp, Wolfgang Reif. *Towards Testing Self-Organizing, Adaptive Systems*. Proceedings of the 26th IFIP WG 6.1 International Conference (ICTSS 2014), Lecture Notes in Computer Science (accepted), Springer

3.17 How Collective and Adaptive our CAS are?

Nikola Serbedzija (FhG FOKUS – Berlin, DE)

License © Creative Commons BY 3.0 Unported license
© Nikola Serbedzija

Main reference N. B. Serbedzija, “Autonomous Systems and Their Impact on Us,” in R. de Nicola, R. Nennicker (eds.), “Software, Services, and Systems – Essays Dedicated to Martin Wirsing on the Occasion of His Retirement from the Chair of Programming and Software Engineering”, LNCS, Vol. 8950, pp. 662–675, Springer, 2015.

URL http://dx.doi.org/10.1007/978-3-319-15545-6_37

The talk examines the computing principles inspired by the nature (in a broader sense) and explores the design and deployment issues of technical systems that interfere with individuals and/or societies (placing humans directly into the processing loop). Due to inevitable impact that smart, mobile and web technologies have on individual and social behavior, inclusion of humanities in the early design phase is condition sine qua non. The talk further explore the possibilities of enriching current collective adaptive approaches with some concepts from social sciences and psychology.

3.18 Three Behavioural Equivalences for Chemical Reaction Networks

Mirco Tribastone (University of Southampton, UK)

License © Creative Commons BY 3.0 Unported license
© Mirco Tribastone

Joint work of Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin

Chemical reaction networks (CRNs) can be seen as a compact language for parallel computation, where the output of an algorithm is given by the concentration of the species at some point in time according to an underlying semantics based on continuous-time Markov chains (CTMCs) or on ordinary differential equations (ODEs).

Using a species-as-process analogy, we study behavioural equivalences over species of a CRN inspired by traditional approaches in established models of computation such as labelled transition systems. We define three equivalences in the Larsen-Skou style of probabilistic bisimulation that identify a partition of the species such that the dynamics of a CRN can be described only in terms of the equivalence classes. In Exact Fluid Lumpability, equivalent species have the same ODE solutions when starting from identical initial conditions. In Differential Species Bisimulation, each equivalence class represents the exact sum of the ODE trajectories of its member species. In Markovian Species Bisimulation, a partition over species identifies an exact aggregation in terms of ordinary lumpability over the states of the underlying CTMC.

For each equivalence relation we develop an efficient partition-refinement algorithm for computing the coarsest aggregations. Using a prototypal implementation, we find significant reductions in a number of models of biological processes available in the literature.

3.19 Engineering Autonomous Ensembles

Martin Wirsing (LMU München, DE)

License © Creative Commons BY 3.0 Unported license
© Martin Wirsing

Joint work of Wirsing, Martin; Hölzl, Matthias; Tribastone, Mirco; Zambonelli, Franco
Main reference M. Wirsing, M. M. Hölzl, M. Tribastone, F. Zambonelli, “ASCENS: Engineering Autonomic Service-Component Ensembles,” in Proc. of the 10th Int’l Symp. on Formal Methods for Components and Objects (FMCO’11), LNCS, Vol. 7542, pp. 1–24, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-35887-6_1

Today’s developers often face the demanding task of developing software for ensembles: systems with massive numbers of nodes, operating in open and non-deterministic environments with complex interactions, and the need to dynamically adapt to new requirements, technologies or environmental conditions without redeployment and without interruption of the system’s functionality. Conventional development approaches and languages do not provide adequate support for the problems posed by this challenge.

The goal of the ASCENS project is to develop a coherent, integrated set of methods and tools to build software for ensembles. To this end we research foundational issues that arise during the development of these kinds of systems, and we build mathematical models that address them. Based on these theories we design a family of languages for engineering ensembles, formal methods that can handle the size, complexity and adaptivity required by ensembles, and software-development methods that provide guidance for developers.

In this lecture I presented a systematic approach for engineering ensembles including an ensemble engineering process, the SOTA approach to ensemble requirements, the underlying formal model called GEM, the SCEL language for designing ensembles, and techniques for the quantitative analysis of ensembles.

3.20 Smart Cities as Heterogeneous Superorganisms: Scenarios and Challenges

Franco Zambonelli (University of Modena, IT)

License © Creative Commons BY 3.0 Unported license
© Franco Zambonelli

The smartness of future cities will be in their capabilities of working like immense heterogeneous superorganisms, bringing together a very diverse actors, from humans, to robots, to ICT devices of any kind.

Engineering the behavior of such systems for the good of our society as a whole and for the good of each individuals in it will be the key societal challenge of the future, and a source for fascinating research challenges in the area of computer science and collective adaptive systems.

4 Working Group Reports

4.1 Modelling, Specification, and Programming for Collective Adaptive Systems

Hella Seebach, Lenz Belzner, Marco Gribaudo, Anabelle Klarl, Michele Loreti, Ugo Montanari, Laura Nenzi, Rocco De Nicola, Christophe Scholliers, Petr Tuma, and Martin Wirsing

License © Creative Commons BY 3.0 Unported license
© Hella Seebach, Lenz Belzner, Marco Gribaudo, Anabelle Klarl, Michele Loreti, Ugo Montanari, Laura Nenzi, Rocco De Nicola, Christophe Scholliers, Petr Tuma, and Martin Wirsing

4.1.1 Introduction

Over the last decades we have witnessed a steep growth in the world population. This increase has a vast impact in the large scale on how cities operate. For example, how to route traffic in the city and where to place parking spots in such a way that the individuals commuting time is minimized. On a smaller scale, big events such as festivals have to be able to predict how the crowd will react in case of a major incident. It is needless to say that each of the individuals at a festival are autonomous entities, yet it is surprising to see that certain patterns can be observed from the group as a whole. Systems consisting out of a large number of individuals exhibiting group behaviour are called collective adaptive systems (CAS). While the collective adaptive systems described above consist solely out of humans the idea is that these systems can consist both out of human entities and/or ICT components.

While our understanding of CAS is getting better over time, the field is not widely understood by the big audience. CAS are omnipresent in current society and it is thus essential to be able to provide the correct set of abstraction in order to model, verify and implement them.

In this paper we show four typical domains of CAS (Sec. 4.1.2). Afterwards we give a description of what collective adaptive system are and how they can be characterized (Sec. 4.1.3). Section 4.1.4 shows how CAS can be modelled and implemented on a computer system. From this overview we conclude that each of the non-trivial collective adaptive systems has a vast need to reason over spatio-temporal properties. Surprisingly, most of the modelling and implementation techniques, do not provide spatio-temporal operations as first class entities. This means that programmers must encode these properties themselves which is time consuming and prone to error. We thus argue that in order to better reason about collective adaptive systems it is essential to focus on these operations. We conclude this paper with perspectives on future research and propose a set of challenges for future researchers to tackle elegantly.

4.1.2 Application Domains

Collective adaptive systems can be found in a lot of different domains. Each application domain naturally leads to different characteristics of CAS which will need multiple new enabling technologies. We just discussed four domains in this workshop which perfectly fit for developing and evaluating techniques for CAS.

Power Management Systems

In current power management systems, big power plants are controlled by electric utilities and other organisations in a flat hierarchy. Utilities and companies manage parts of the

overall power system independently from each other. For each of the big power plants, a schedule is created that postulates the output of the power plant at a given time. Schedules are coarse-grained, providing target values in 15 minute intervals. Small power plants and especially DERs (distributed energy resources) under the control of small cooperatives or individuals produce without external control and feed the power produced into the grid. This lack of control by the electric utilities is compensated by powerful controllable power plants. Current plans are to scale the controllable output further by installing more plants, especially flexible gas-powered ones. Geographical distribution is even increasing with the wide-spread installation of DERs such as biogas plants, solar plants, and wind farms. The relative stability in the network is an emergent behaviour. No single entity of the system can provide this stability in the face of load fluctuations, weather changes, and generator outages.

What makes power systems difficult to manage and optimize is the cost of storing energy: since energy consumption varies remarkably with time (day, week and season), the unit cost of production varies also, because less efficient plants are turned on only at peak time. On the other hand, DERs production capacity is also heavily dependent on weather conditions, and thus quite variable in time. Physical storage systems (hydro pumping stations, batteries, e.g. of electric vehicles) are not very practical, thus the best policy is to try to match consumption and production in an integrated or unbundled market. The goal is to make demand active, trying to move it, whenever possible, to slots where energy is less expensive. Therefore, there is a need for a future power grid in which even small power plants, consumers, as well as prosumers (entities that produce and consume power like an electric vehicle) can be controlled or participate in a scheduling scheme or market, since entrance requirements to power markets, such as the lower limit of 100 kW for contracts at the European Energy Exchange (EEX), exclude access for small organisations. Networked measuring equipment must be equipped to allow observing the grid status and make decisions based on current conditions. Power plants and consumers will be networked, too, and provide future production or consumption. Producers, consumers, and prosumers must be combined into groups, e.g., as aggregators [1] or Autonomous Virtual Power Plants (AVPPs) [2],[3],[4] that create schedules to cover a portion of the load (depending on the demand) aiming at: (i) lowering peak energy consumption by exploiting their flexibility; (ii) reducing electricity cost for the whole population of actors; (iii) increasing robustness by locally dealing with load and output fluctuations; and (iv) making a profit. Remaining research challenges comprise the robust autonomous scheduling of large-scale open heterogeneous systems (including spatial and temporal constraints) as well as security, privacy, and safety aspects, amongst others.

Cloud Computing

Contemporary cloud computing platforms rely on server farms that host a number of dedicated workload processing servers together with the necessary networking and storage infrastructure. The infrastructure does not expose the details of virtual server location at the level of individual server racks or individual network ports – these are managed transparently by the platform provider, possibly using mechanisms such as virtual machine migration [5] or software defined networks [6]. In contrast, higher granularity location is exposed – complex application deployment scenarios use it to make sure that both application code and application data is distributed appropriately to accommodate the (often conflicting) requirements on communication efficiency, failure resiliency, cost and other factors.

Although many cloud computing applications adopt somewhat conservative resource management techniques (such as limiting dynamic server allocation to manually selected server farms), many research directions seek to amplify the existing cloud computing benefits

by introducing mechanisms such as cloud federations or ad hoc and opportunistic clouds [7, 8].

On many levels, these directions strengthen the collective adaptive system characteristics of the cloud. There are multiple focus areas for research in the domain of cloud computing. *Efficient resource allocation*: Both the cloud platform and the cloud applications seek to maximize utility and minimize cost by sharing resources (servers, storage, network). Efficient resource allocation is a collective task where multiple adaptive entities (platform components and application components) allocate and release resources to meet their specific requirements. The domain offers research challenges in both cooperative and competitive resource allocation algorithms in presence of changing requirements [9, 10, 11, 12, 13, 14, 15]. In turn, these contain challenges in monitoring and predicting the impact of particular resource allocation, needed to perform allocation decisions. *Robustness against failures*: Especially in an open cloud with voluntary participation, node failures and application failures are expected rather than exceptional situations. The domain requires research into efficient failure resilient algorithms, behaviour modelling in presence of (possibly dependent) failures and other challenging problems [16]. *Security against abuse*: As an open environment with heavy resource sharing, cloud computing exposes many opportunities for abuse. These include not only the more traditional security related issues (virtual machine hijacking, data theft and other), but also the possibility of using the available resources beyond fair share or outright free-loading. There is a need for strategies and mechanisms that would prevent such abuse [17, 18, 19, 20]. *Preventing negative emergence*: The cloud environment incorporates entities whose behaviour is largely automated but rarely fully disclosed or even fully understood. Such an environment is easily prone to emergent behaviour with negative consequences, for example oscillations in the adaptive feedback mechanisms. The domain can benefit from research into preventing, detecting or managing cases of negative emergence.

As another practical benefit of the cloud computing domain, the difficulty of the identified challenges varies with the degree of openness and heterogeneity that is considered – a centralized resource allocation in a closed homogeneous cloud faces different issues than a cooperative distributed resource allocation in an open heterogeneous cloud with voluntary participation. Although multiple projects already started tackling some of the listed challenges [21, 22, 23, 24, 25, 26, 27, 28], there are still many topics waiting for advances.

Telecommunication – LTE Resource Allocation

LTE technology, used in 4G mobile phone communications, employs a channel resource allocation scheme based on orthogonal frequency-division multiplexing. In particular it supports for both time-division multiplexing and frequency-division multiplexing, splitting the communication spectrum in a set of resource blocks.

Each participating device is equipped by multiple antennas, and can transmit on more frequencies at the same time. The total bandwidth available to a device depends on the number of blocks it can use to transmit/receive data. The LTE technology envisage the possibility of dynamically allocating the resource blocks depending on the actual demand, to improve the performances that can be achieved by the single user. The service provider usually operates block allocation in a centralized manner: this however limits the bandwidth that could be achieved since interference might arise from the carrier being shared by different providers. An autonomous solution, where each mobile agent can acquire and release block resources, could improve the available bandwidth overcoming these difficulties. This however is not an easy task due to the characteristics of the wireless medium that is affected by limitations such as the hidden terminal problem [29]. The problem can become even more interesting when the cellular infrastructure is complemented with alternative wireless access

	Homogeneous Heterogeneous	Collaborative Competitive	Low ind. impact High ind. impact				
Cloud	<-----X	-----	-----	Space	Sync / async	Continuous / discrete	Open / closed
Wristband	X	·	·				
4G LTE	<·X	··	-----				
Power grids	<---X	-----	-----				

■ **Figure 1** Categorization of the considered applications.

technologies such as WiFi hotspots [30]. In this way data traffic can be offloaded whenever possible towards such hotspots, at the price of a possible degradation in the quality of service experienced by the users [31, 32]. Preliminary studies of such systems using CAS-based techniques have been proposed in [33, 34]. The key solutions in this direction will also be the basis to the next generation of wireless and cellular communications that exploits more advanced techniques such as *cognitive radio* [35] in which terminals and base stations harvest for unused radio frequencies and spectrum bandwidths to increase their transmission capacity.

Wearable Computational Devices

Due to the advances in hardware technology and the miniaturisation of electronic components it has become feasible to make wearable computational devices. These wearable devices open up opportunities for new and exciting applications. Also in the world of collective adaptive systems this new technology can be exploited. One particular application is the use of wristbands equipped with wireless near field communication. When a large number of people are equipped with such wristbands these wristbands could light up in order to provide additional functionalities to the users. One application of these wristbands could be to make figures at large scale events by lighting up the wristbands at synchronised moments in time. Additionally the same wristbands could be used at mass events to drive people to the exit in case of disaster. Challenging is amongst others the situation of simple, limited nodes (simple communication, limited resources) and the unpredictable position of the nodes.

4.1.3 Common Features and Characteristics of Application Domains

In the workshop, we identified multiple characteristics for CAS. The four mentioned application domains can be categorized against a set of different features that will guide in the selection of the most appropriate modelling technique. Figure 1 summarizes the results.

The first aspect we considered is the type of elements that compose the CAS. The elements can be *homogeneous*: all the cooperating entities can be considered to be identical. This is for example the case of the bracelets in the wristband application, where all the devices are exactly the same. *Heterogeneous* applications are instead composed of agents that are completely different one from the other. A typical example is the power grid scenario, where each producer or consumer is completely different from the others, of course depending on the level of abstraction. Both the cloud and the LTE scenario have some degree of heterogeneity due to the fact that they are composed by different devices, produced by different manufacturers. However all the devices are abstracted by the *role* they are playing in the application: this allows us to consider them homogeneous from a modelling perspective.

The second aspect we discussed is whether the agents are *collaborative* or *competitive*. For

example, the wristband application is a clear example of a collaborative system: each agent cannot gain any advantage by not cooperating with the others. The LTE network is instead an example of a competitive application, where each mobile device tries to acquire all the available bandwidth to improve its communication performance. Cloud computing can be either collaborative or competitive depending on the specific application we are considering. A Big Data application might be competitive to gain more resource to parallelize its execution and to reduce its running time. A Platform-as-a-service job can instead be consolidated with other applications to increase the chance of having idle machines that can be switched off, reducing the total energy consumption. The prosumer in the power management systems are first and foremost competitive to optimize their benefit. But in future energy grid scenarios they have to collaborate in organisational structures to be able to participate for example on the energy market.

The third feature is the impact that a local agent can have on the entire community. In the wristband application it can be minimal, since an agent can at most do not propagate a message and do not properly switch the colour of the bracelet. In the power grid example the impact is instead maximum, since other nodes might be relying on the production or consumption of energy of other participants in the network. 4G LTE might have a limited impact, since most of devices are autonomous. However, the presence of a shared environment (the communication spectrum) can have an impact on a large set of devices in case of malicious signals that could be generated to interfere with the regular transmissions. The particular characterization of the cloud-computing scenario depends on the considered application since it can have either a low impact (as the exclusion of a physical node in an Infrastructure-as-a-service scenario), or a high impact (for example when the shut down of a node in a distributed storage application makes a file no longer accessible).

Other important features that characterize an application from a modelling point of view are the presence of space, the fact of being synchronous or asynchronous, discrete or continuous, open or closed. All the examples that we consider in this work rely somehow on a concept of *space*. Space can be either a physical space or a logical space. For example, both the stadium where the concert is held in the wristband application, or the area where base stations are located in the LTE application, are examples of physical spaces. The nodes interconnected by the distribution network in the power grid application, and the servers and routers in the cloud application, are examples of logical space. In both cases the system topology can be described by a graph, where nodes of the system correspond to nodes on the graph. All the considered applications are *asynchronous*, but they build up some level of *synchronism* to allow the system to reach their goals: depending on the level of abstraction that we want to consider, different modelling techniques can be used to specifically target their synchronous or asynchronous features. Most of the applications are *discrete*, in the sense that they change their state according to events that happens in discrete time instants. The power grid application however, requires a *continuous* time approach since problems can arise and must be handled in few milliseconds. This leads to the requirement of radically different modelling techniques. Finally applications can be considered either *open* or *closed*. The wristband is a classical example of a closed application. Also in this case, depending on the level of abstraction and on the features we are interested to consider, different modelling techniques could be employed.

4.1.4 Methods to approach CAS

We identified three different main approaches to model CAS: systems of systems, autonomy, and aggregation. They are not exclusive; rather they should be regarded as a kind of

dimensions that a particular system or solution exhibits. Afterwards we discussed several modelling techniques for the different levels of abstraction.

Systems of Systems (Roles)

A core characteristic of CAS to be modelled are behavioural and communicational aspects, both on the individual and the collective level. A key challenge here is the specification of organisational, communicational or behavioural collectives. Also, modelling these structures has to take into account reconfiguration of collectives at runtime due to changing situations (i.e. adaptation). The properties stated above lead to the idea of considering systems of systems [36] as an appropriate way of modelling CAS. The collectives may be organized in hierarchical or overlapping ways, also depending on spatial aspects. In the workshop, aggregation and organization based on roles, communication patterns and spatio-temporal properties have been discussed.

Autonomy (Reasoning)

One way to drive adaptation is to provide learning and planning capabilities to individuals and collectives alike. Reasoning and planning provide ways for autonomic system reorganization according to current needs and system goals. In the context of CAS, this gives rise to questions about individual and collective knowledge gathering and transformation as well reasoning capabilities. In especial, if considering systems of systems the question arise how to compare and evaluate systems e.g. on different or equal levels of hierarchies [37] or in different locations leading to different circumstances.

Aggregation (Quantification)

CAS may consist of extreme numbers of individuals. Also, these numbers may be unknown at design-time and/or runtime. Quantitative analysis approaches identify and abstract away symmetries and structure of the system in order to allow for efficient computation of system properties. While this scalability is highly desirable, it comes at the cost of specializing towards a particular problem or situation – quantitative approaches are strongly coupled to the way a CAS is modelled. Thus, they should drive modelling approaches as well as respect any abstractions made when modelling CAS. Most of the techniques in this field rely on mean field solutions [38, 39]: the system is studied by considering variables that counts the number of elements in the same state, and by studying their evolution using a set of ordinary differential equations. The mean field approximation basically states that a large number of objects that randomly evolve tend to show a deterministic mean behaviour as their count tends to infinity. On this assumption, many higher level modelling techniques based on process algebra [40, 41], or Markovian agents [42] have been developed.

Each of the approaches already provides solutions for problems studied under particular aspects. What remains a mostly open challenge is the combination of modelling and solutions from the different perspectives as well as the integration of spatio-temporal aspects in the mentioned techniques. For example, modelling and collective organization formalisms have to (a) provide methods for integration of reasoning in the modelling process and (b) allow for autonomous, goal- or situation-based reconfiguration. On the other hand, reasoning has to account for structural changes, and has to infer about the collective structure. Also, it seems an interesting challenge how different quantitative approaches could be instrumented autonomously based on current system configuration and accounting for autonomous reconfiguration at runtime.

Modelling CAS at different level of abstractions

Different languages have been proposed or used to support modelling, analysis and deployment of CAS. Some of these are general purpose languages, like Java, that, while providing appropriate API, can be used to program and deploy systems also on complex distributed infrastructures. Others languages are domain specific [43, 44, 45, 46, 47] and are equipped with syntactic constructs specifically thought for modelling relevant aspects of CAS. These domain specific languages are typically more oriented to specification than to deployment and provide formal and automatic tools that can be used to support analysis.

This variety of tools and languages can be used by a CAS designer to tackle the system modelling at different level of abstractions. Indeed, each language can be used to describe and analyse a system from different perspective. However, to take a real advantage from this plethora of tools formal links between the considered formalism are definitively needed. The links, that can be rendered in terms of *model transformation*, will be first of all used to relate the different models described in the different languages. These relations will be then instrumental to use the results of analysis performed in a given language to update/improve the other models.

4.1.5 Conclusion and Open Challenges

After we discussed the different application domains for CAS and considered suitable modelling and specification techniques we identified a set of challenges that must be tackled before CAS will be integrated in today's or future ICT systems. As one main result the working group came to the conclusion that spatio-temporal properties are of great value for the modelling and implementation of CAS but are not yet appropriately integrated in the available methods. One concrete challenge: Investigation of the design of CAS learning spatio-temporal requirements. The idea is to optimise the specification and implementation of the space features in the model in such a way that the satisfiability of a spatio-temporal property is maximised. As we know, the verification of global properties on CAS is often an intractable task from a computational point of view. For this reason, such properties will have to be decomposed in a set of local requirements in the optimisation process.

Further the participants of CAS need methods to reason about local versus global or even conflicting goals of the system. These decisions strongly depend on the organisational structures and the presence or absence of central institutions in the CAS. One concrete challenge: When describing/implementing CAS two aspects are crucial: The specification of the actual actions that the different components have to perform (behavioural specification) and the specification of the goal that the single components or the collectives have to achieve (goal specification). Usually these two kinds of activities are performed by taking advantage of very different tools, the former are performed with classical imperative programming language while the latter rely on declarative specifications. The foreseen challenge is the reconciliation of these two approaches to, e.g., able to take decisions about the next actions after having measured how far the goal is and what is the best choice to get closer to it.

Quantitative approaches for modelling and analysis of CAS help to meet the challenge of state space explosion if considering large-scale CAS. Beside the mean field solutions outlined in the previous section, new physically inspired techniques could be applied. One example could come from fluid dynamics, leading to *fluid approximations to CAS modelling*. If we consider a very large number of agents, densely packed, their evolution can be approximated as the motion of a fluid. To give an idea, let us imagine that agents can evolve through a finite set of modes $i \in \{1, \dots, M\}$. Let us also focus on two-dimensional space where agents

can evolve. The state of whole system at time t can be characterized by a set of functions $p_i(x, y, t)$ that describes the density of agents (measured in agents per unit area) in state i at position (x, y) . The evolution of $p_i(x, y, t)$ can be described by a set of partial differential equations, similar to the one used by the mass continuity law in fluid dynamics. From the state density $p_i(x, y, t)$ several performance metrics can be derived. These measures can be used to assess several properties, such as for example determining if an emergent behaviour can appear, and which could be the convergence rate as function of the parameters of the model.

References

- 1 Peeters, E., Belhomme, R., Batlle, C., Bouffard, F., Karkkainen, S., Six, D., Hommelberg, M.: ADDRESS: scenarios and architecture for active demand development in the smart grids of the future. In: Electricity Distribution-Part 1, 2009. CIRED 2009. 20th International Conference and Exhibition on, IET (2009) 1–4
- 2 Steghöfer, J.P., Anders, G., Siefert, F., Reif, W.: A system of systems approach to the evolutionary transformation of power management systems. In: GI-Jahrestagung. (2013) 1500–1515
- 3 Bremer, J., Rapp, B., Sonnenschein, M.: Encoding distributed search spaces for virtual power plants. In: Computational Intelligence Applications In Smart Grid (CIASG), 2011 IEEE Symposium on. (April 2011) 1–8
- 4 Becker, B., Allerdig, F., Reiner, U., Kahl, M., Richter, U., Pathmaperuma, D., Schmeck, H., Leibfried, T.: Decentralized energy-management to control smart-home architectures. In Müller-Schloer, C., Karl, W., Yehia, S., eds.: Architecture of Computing Systems – ARCS 2010. Volume 5974 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 150–161
- 5 Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., Warfield, A.: Live migration of virtual machines. In: Proceedings of NSDI 2005, USENIX
- 6 Casado, M., Koponen, T., Ramanathan, R., Shenker, S.: Virtualizing the network forwarding plane. In: Proceedings of PRESTO 2010, ACM
- 7 Grozev, N., Buyya, R.: Inter-cloud architectures and application brokering: Taxonomy and survey. Software: Practice and Experience (2014)
- 8 Kirby, G., Dearle, A., Macdonald, A., Fernandes, A.: An approach to ad hoc cloud computing
- 9 Warneke, D., Kao, O.: Exploiting dynamic resource allocation for efficient parallel data processing in the cloud. IEEE Transactions on Parallel and Distributed Systems **22**(6) (2011)
- 10 Ishakian, V., Sweha, R., Bestavros, A., Appavoo, J.: CloudPack. In: Proceedings of MIDDLEWARE 2012, Springer
- 11 Wang, J., Hua, R., Zhu, Y., Wan, J., Xie, C., Chen, Y.: RO-BURST: A robust virtualization cost model for workload consolidation over clouds. In: Proceedings of CCGRID 2012, IEEE
- 12 Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Journal of Future Generation Computing Systems **28**(5) (2012)
- 13 Li, J., Shuang, K., Su, S., Huang, Q., Xu, P., Cheng, X., Wang, J.: Reducing operational costs through consolidation with resource prediction in the cloud. In: Proceedings of CCGRID 2012, IEEE
- 14 Dong, J., Jin, X., Wang, H., Li, Y., Zhang, P., Cheng, S.: Energy-saving virtual machine placement in cloud data centers. In: Proceedings of CCGRID 2013, IEEE

- 15 Raghavendra, R., Ranganathan, P., Talwar, V., Wang, Z., Zhu, X.: No “power” struggles: Coordinated multi-level power management for the data center. In: Proceedings of ASPLOS 2008, ACM
- 16 Khalifa, A., Azab, M., Eltoweissy, M.: Resilient hybrid mobile ad-hoc cloud over collaborating heterogeneous nodes. In: Proceedings of COLLABORATECOM 2014, IEEE
- 17 Velloso, P.B., Laufer, R.P., de O. Cunha, D., Duarte, O.C.M.B., Pujolle, G.: Trust management in mobile ad-hoc networks using a scalable maturity-based model. *IEEE Transactions on Network and Service Management* **7**(3) (2010)
- 18 Huang, D., Zhang, X., Kang, M., Luo, J.: MobiCloud: Building secure cloud framework for mobile computing and communication. In: Proceedings of SOSE 2010, IEEE
- 19 He, Q., Wu, D., Khosla, P.: SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks. In: Proceedings of WCNC 2004, IEEE
- 20 Buchegger, S., Boudec, J.Y.L.: Self-policing mobile ad-hoc networks by reputation systems. *IEEE Communications* **43**(7) (2005)
- 21 Bernard, Y., Klejnowski, L., Hähner, J., Müller-Schloer, C.: Towards trust in desktop grid systems. In: 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGrid 2010, 17-20 May 2010, Melbourne, Victoria, Australia. (2010) 637–642
- 22 Bulej, L., Bureš, T., Horký, V., Keznl, J.: Adaptive deployment in ad-hoc systems using emergent component ensembles: Vision paper. In: Proceedings of ICPE 2013, ACM
- 23 Mayer, P., Velasco, J., Hennicker, R., Puviani, M., Tiezzi, F., Pugliese, R., Keznl, J., Bureš, T. In: *The Autonomic Cloud*. Springer (2015)
- 24 Amoretti, M., Grazioli, A., Senni, V., Zanichelli, F.: Towards a formal approach to mobile cloud computing. In: Proceedings of EUROMICRO PDP 2014
- 25 Sebastio, S., Amoretti, M., Lluch-Lafuente, A.: A computational field framework for collaborative task execution in volunteer clouds. In: Proceedings of SEAMS 2014, ACM
- 26 Celestini, A., Lluch-Lafuente, A., Mayer, P., Sebastio, S., Tiezzi, F.: Reputation-based cooperation in the clouds. In: Proceedings of IFIP TM 2014
- 27 Vassev, E., Hinchey, M., Mayer, P.: Formalizing self-adaptive clouds with knowlang. In: Proceedings of ISoLA 2014
- 28 Klarl, A., Mayer, P., Hennicker, R.: HELENA@Work: Modeling the science cloud platform. In: Proceedings of ISoLA 2014
- 29 Tsertou, A., Laurenson, D.I.: Revisiting the hidden terminal problem in a csma/ca wireless network. *Mobile Computing, IEEE Transactions on* **7**(7) (2008) 817–831
- 30 Trullols, O., Fiore, M., Casetti, C., Chiasserini, C., Ordinas, J.B.: Planning roadside infrastructure for information dissemination in intelligent transportation systems. *Computer Communications* **33**(4) (2010) 432–442
- 31 Han, B., Hui, P., Kumar, V.A., Marathe, M.V., Pei, G., Srinivasan, A.: Cellular traffic offloading through opportunistic communications: A case study. In: Proceedings of the 5th ACM Workshop on Challenged Networks. CHANTS '10, New York, NY, USA, ACM (2010) 31–38
- 32 Han, B., Hui, P., Kumar, V.S.A., Marathe, M.V., Shao, J., Srinivasan, A.: Mobile data offloading through opportunistic communications and social participation. *IEEE Transactions on Mobile Computing* **11**(5) (May 2012) 821–834
- 33 Gribaudo, M., Manini, D., Chiasserini, C.: Studying mobile internet technologies with agent based mean-field models. In: *Analytical and Stochastic Modelling Techniques and Applications – 20th International Conference, ASMTA 2013, Ghent, Belgium, July 8–10, 2013*. Proceedings. (2013) 112–126
- 34 Chiasserini, C., Gribaudo, M., Manini, D.: Traffic offloading/onloading in multi-rat cellular networks. In: Proceedings of the IFIP Wireless Days, WD 2013, Valencia, Spain, November 13-15, 2013. (2013) 1–7

- 35 Mitola, J., Maguire, G.Q., J.: Cognitive radio: making software radios more personal. *Personal Communications, IEEE* **6**(4) (Aug 1999) 13–18
- 36 Sage, A.P., Cuppan, C.D.: On the systems engineering and management of systems of systems and federations of systems. *Inf. Knowl. Syst. Manag.* **2**(4) (December 2001) 325–345
- 37 Schiendorfer, A., Steghöfer, J.P., Reif, W.: Synthesis and abstraction of constraint models for hierarchical resource allocation problems. In: *Proc. of the 6th International Conference on Agents and Artificial Intelligence (ICAART)*. Volume 2.
- 38 Benaim, M., Boudec, J.L.: A class of mean field interaction models for computer and communication systems. *Perform. Eval.* **65**(11-12) (2008) 823–838
- 39 Bobbio, A., Gribaudo, M., Telek, M.: Analysis of large scale interacting systems by mean field method. In: *Fifth International Conference on the Quantitative Evaluation of Systems (QEST 2008)*, 14-17 September 2008, Saint-Malo, France. (2008) 215–224
- 40 Bortolussi, L., Hillston, J., Latella, D., Massink, M.: Continuous approximation of collective system behaviour: A tutorial. *Perform. Eval.* **70**(5) (2013) 317–349
- 41 Hayden, R.A., Stefanek, A., Bradley, J.T.: Fluid computation of passage-time distributions in large markov models. *Theor. Comput. Sci.* **413**(1) (2012) 106–141
- 42 Cerotti, D., Gribaudo, M., Bobbio, A.: Markovian agents models for wireless sensor networks deployed in environmental protection. *Rel. Eng. & Sys. Safety* **130** (2014) 149–158
- 43 De Nicola, R., Loreti, M., Pugliese, R., Tiezzi, F.: A formal approach to autonomic systems programming: The SCEL language. *TAAS* **9**(2) (2014) 7
- 44 Feng, C., Hillston, J.: PALOMA: A process algebra for located markovian agents. In Norman, G., Sanders, W.H., eds.: *Quantitative Evaluation of Systems – 11th International Conference, QEST 2014, Florence, Italy, September 8–10, 2014. Proceedings*. Volume 8657 of *Lecture Notes in Computer Science.*, Springer (2014) 265–280
- 45 Alrahman, Y.A., Nicola, R.D., Loreti, M., Tiezzi, F., Vigo, R.: A calculus for attribute-based communication. In: *Proceedings of the ACM Symposium on Applied Computing, SAC 2015, ACM (2015) To appear.*
- 46 Bortolussi, L., Nicola, R.D., Galpin, V., Gilmore, S., Hillston, J., Latella, D., Loreti, M., Massink, M.: Caspa: a collective adaptive stochastic process algebra. In Bertrand, N., Tribastone, M., eds.: *Proceedings Twelfth International Workshop on Quantitative Aspects of Programming Languages and Systems, QAPL 2015, April 2015. EPTCS (2014) To appear.*
- 47 Viroli, M., Damiani, F.: A calculus of self-stabilising computational fields. In Eva Kühn, Pugliese, R., eds.: *Coordination Models and Languages – 16th IFIP WG 6.1 International Conference, COORDINATION 2014, Held as Part of the 9th International Federated Conferences on Distributed Computing Techniques, DisCoTec 2014, Berlin, Germany, June 3–5, 2014, Proceedings*. Volume 8459 of *Lecture Notes in Computer Science.*, Springer (2014) 163–178

4.2 Verification of CAS

Luca Bortolussi, Giacomo Cabri, Giovanna Di Marzo Serugendo, Vashti Galpin, Jane Hillston, Roberta Lanciani, Mieke Massink, Mirco Tribastone, and Danny Weyns

License © Creative Commons BY 3.0 Unported license

© Luca Bortolussi, Giacomo Cabri, Giovanna Di Marzo Serugendo, Vashti Galpin, Jane Hillston, Roberta Lanciani, Mieke Massink, Mirco Tribastone, and Danny Weyns

Verification is the process of assessing how well a system meets a specification or requirement. A variety of approaches have appeared in the literature, ranging from model checking to

static analysis of source code and theorem proving. In this working group, we primarily focused on verification based on model checking [2, 13], in which a state-based description of the system is assessed with respect to a property expressed in an appropriate specification language, like a temporal logic. In particular, we considered the challenges that arise in the model checking of Complex Adaptive Systems (CAS), which are systems comprised of a large number of heterogeneous agents which, interacting together, produce a complex array of collective behaviours.

In our working group in Dagstuhl, we first identified the major issues and more interesting challenges that arise in the process of verification of CAS. Afterwards, we divided in two subgroups to discuss in detail specific topics related to this general framework. In particular, we chose to investigate: (1) the quantitative or stochastic model checking in the presence of uncertainty, and (2) the specifications and logics which capture the spatial arrangements of systems, characterising the impact of those arrangements on collective behaviour. A brief account of each subgroup is given below.

4.2.1 Introduction

In our discussion, we first identified the numerous important issues and challenges that arise when we want to verify a CAS. In the following, we outline just a few of them.

Adaptation and verification

Adaptation is itself an interesting phenomenon which could be subjected to verification. In particular, we considered the issue of quantifying how adaptive the system is, and we identified a number of different measures that can be used to validate the adaptation of a CAS. For example:

Speed of adaptation – Once an adaptation is initiated, how rapidly does the system enter a stable behaviour? In this requirement the stable behaviour is not necessarily a single state, but could be a new stable equilibrium with a given probability distribution over a set of states.

Robustness – How often does the system adapt? Is there a danger of “thrashing”, meaning that the system alternates between different adaptations, achieving one, and shortly after pursuing another?

Effectiveness of adaptation – How closely does a system satisfy the revised property or goal after an adaptation?

Verifying global properties based on local behaviours

In many cases, the properties that are of interest to both system developers and system users are global requirements related to emergent behaviours of CAS. But the populations of CAS are comprised of a large number of single entities, whose local behaviour is usually more accessible and intuitive than the collective description of the system (due to the way CAS are implemented). Hence, there is a need for compositional approaches that are able to validate global properties of CAS, building on the verification of local requirements related to single agents or group of individuals. First promising results in this respect were achieved in [6] in the context of quantitative model checking of population models.

Verification in the presence of uncertainty

A characteristic feature of CAS is the uncertainty. For example, the structure of part of the system may be totally unknown or unknown at a particular instant in time. Moreover, the goals and objectives of a single agent may be hidden from the others, possibly due to an ongoing adaptation process. At a finer level of detail, the rates or probabilities that govern the dynamic behaviour of the system may be unknown, or changing in undefined ways, meaning that model of the CAS could be underspecified.

Scalability

Many verification techniques rely upon explicit representation of the state space of the model. In a CAS this is challenging in two respects. Firstly, not all possible states may be known due to future adaptation, as discussed above. Secondly, even if the “complete” state space is known or can be anticipated, the model will typically include too many states to be represented explicitly. Alternatives such as statistical model checking avoid constructing the whole state space at once, but then become computationally very expensive due to the sampling approach that must be adopted, necessitating many simulation runs before a verification can be approximated. An alternative is to use techniques based on fluid or mean field representation of the discrete state space [20, 4, 5, 6], but these approaches are still in their infancy.

Openness

Openness is an inherent property of CAS, as agents may join or leave the system throughout its lifetime. This poses severe challenges for state-based modelling techniques, particularly if there is the possibility that the population of the system grows unboundedly. In these scenarios, care is needed in phrasing the properties to be satisfied by the system. For example, it may be more appropriate to express goals in terms of proportions of agents rather than absolute numbers.

Quantified verification as a driver for adaptation

When the models contain quantitative information about the system, such as information about the timing and likelihood of events, it is possible to assess a system against a property not just in terms of boolean satisfaction but in a more quantified way. This can be viewed as measuring the degree to which a system satisfies a property, or the distance that a system is from satisfying the property. When this form of quantification is added to the verification process it is possible to see how verification can become a driver for adaptation. As the system undergoes adaptation, its progress towards a goal can be explicitly measured.

In recent years several authors have considered quantitative satisfaction of properties. In this framework, when a system is assessed against a property the result is a measure of distance indicating how close the system comes to satisfying the property [23, 21, 18]. For example, if the property is satisfied then the distance is zero. In this framework a system which fails to satisfy a property at distance 0.2 may be considered preferable to a system which fails to satisfy the property at distance 0.8. This approach has been used to conduct sensitivity analysis of model parameters with respect to desirable properties [24] and to seek parameters that bring a model closest to property satisfaction [25, 1]. This latter approach could be deployed to drive adaptation through verification.

Spatial aspects

In many CAS the location of agents, and bounded ranges of communication are an important factor in the design and realisation of the system. Thus it is essential that location and movement are treated as primitives in both modelling and verification techniques developed to support CAS. Often in existing techniques, if handled at all, space is treated only logically, and the relationships between locations are qualitative rather than quantitative. Thus a model may capture that locations are in some sense “adjacent” but not the actual distance between them. However, if agents are operating, for example, in a wireless sensor network, the actual distance between them will determine whether or not they are able to communicate, or the energy cost of doing so. Incorporating detailed spatial information means that model checking must consider spatio-temporal properties, a further level of challenge.

4.2.2 Motivation

To motivate our discussions in the working group we considered possible applications, where some of the discussed issues would have practical relevance. In particular, we identified the following:

Global adaptation: In this application, the adaptation takes place in the environment, while the agents operating within the system keep the same behaviour. An example of this would be a smart grid, where differential pricing is used to stimulate a change in the behaviour of the end users. In this scenario, a collective change in the dynamics of the system is obtained by acting on the environment (lowering the price of the energy during less busy period of the day), while the agents (the end users) keep the same goals and objectives (to buy energy at the cheapest possible price).

Agent adaptation: In these scenarios the agents change their behaviour based on local information, generating an effect on the collective behaviour of the system. An example of this is a peer-to-peer file sharing systems such as BitTorrent [14, 22]. In this application, the end users locally adapt to improve their own quality of service, while the environment, the BitTorrent protocol, remains unchanged. Moreover, the choices made by the single user affect its rate of uploading content to the network, thus altering the behaviour of the whole network.

4.2.3 Subgroup I: Uncertainty in CAS

When there is uncertainty in the behaviour of the system under consideration it makes the task of verifying a system even more challenging. Unfortunately in CAS, the inherent adaptability means that uncertainty is also inherent in the system. Broadly speaking, we identified two distinct approaches:

- Offline verification, before the system is deployed, tries to anticipate the possible range of behaviours that can be encountered.
 - Online verification, conducted while the system is running, reflects the observed behaviour.
- We anticipate that in many CAS both approaches may be needed.

Offline verification

In this case we might assume that a model is available which aims to capture all the possible behaviours, but that some aspect of the model, in particular the parameters corresponding to any given time or mode of behaviour, are unknown. To some extent this is the aim of probabilistic models which treat the parameters of models as random variations, with a

defined distribution function which gives an estimate of the range of possible values and the likelihood of each. Such approaches have long been used in performance and dependability modelling to abstract away from concrete instances of behaviour and capture instead a range of possible behaviours in a statistically meaningful way. However, the uncertainty that we are considering here means that even the random variables characterising the range of behaviours may be unknown, or unknowable. This arises because we are often interested in designing systems satisfying emergent properties. Uncertainty can emerge in many ways; probably one of the simplest ways is to consider models that are structurally defined, but which can have unspecified parameter values p , possibly belonging to a bounded set P . Furthermore, we assume that we are dealing with stochastic models, so that behaviours are satisfied with a certain probability, rather than always or never.

The question of how we can verify properties under such uncertainty is a difficult one. One approach is to compute the probability of satisfaction of a certain property ϕ (for instance, encoded as a linear temporal logic formula) as a function of the parameter values $p \in P$. Let us denote this satisfaction function by $f(p)$. An exhaustive numerical computation (or bounding) of $f(p)$ for $p \in P$ is infeasible even for simple models. Recently, a statistical method was proposed by Bortolussi *et al.* [7], leveraging machine learning ideas to characterise statistically the function f . This approach can be complemented with that of [3], where the authors use a similar statistical scheme to evaluate a robustness measure associated with a property ϕ , and use it for system design purposes.

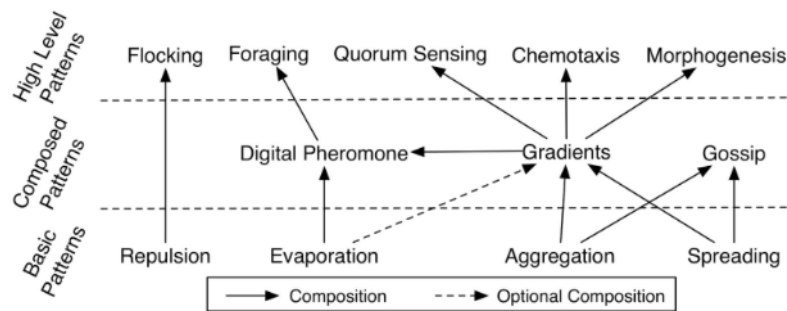
Applying these methods to CAS, however, still remains challenging. On one side, the size of such systems is so large that even fast statistical approaches can fail to provide an answer in a computationally acceptable time. In this respect, decomposition of the system into modules, seems an interesting direction for future work. The challenging problems here are how to identify such modules, and how to combine verification results of modules into a verification procedure (possibly providing bounds on the actual probabilities/robustness scores).

Another challenging problem is how to generalise the statistical approaches of [3, 7] to the case in which parameters are not only taking an unspecified value in a given set P , but they can also change over time (e.g. the arrival rate of customers at a bike or car sharing station will vary according to the time of day). This is a step towards verification of stochastic models of open CAS.

Finally, CAS are often subject to structural uncertainty, not only to parametrical one. In other words, the complete structure of the model itself may not be known. This seems to bring even more challenges to the analysis, particularly because structural changes of a model often result in discontinuous behaviour, which makes it much harder to exploit the statistical tools used in [3, 7] and similar work. Promising work on evolving models has recently been developed by Ghezzi *et al.* [17]. Whilst this was developed considering a specification that evolves as a system develops, it nevertheless has interesting ideas that could be applicable to the problem considered here.

Online verification

The importance of runtime verification has already been recognised by the software engineering community. In the simplest cases this might involve monitoring a system with respect to a formal specification of acceptable behaviour, such as a finite state machine, and raising an alarm if the observed behaviour deviates from the specification. However, systems such as CAS where the operating environment, the user requirements and the system itself are all changing over time, cannot be dealt with in such simplistic approaches. Moreover, functional



■ **Figure 2** Patterns for self-organisation [16].

correctness is no longer considered enough; in [9], the authors argue that quantitative aspects of behaviour must also be verified at runtime for self-adaptive software. In [19], the authors emphasise the need for formalising the adaptation components themselves, which is important to provide guarantees of correctness of the adaptation behaviour and changes of the adaptation logic to handle changing goals. Results from a related Dagstuhl seminar on Assurances for Self-adaptive Systems coined the term *perpetual assurances* [27] as an enduring process where new evidence is provided by combining system-driven and human-driven activities to deal with the uncertainties that the system faces across its lifetime.

In online verification a global model of the system may no longer be available, and if it is likely to be too computationally expensive to be used to give answers within the required timescale for runtime decision making. Thus it becomes attractive to develop compositional approaches to verification that allow results from lower levels, i.e. properties relating to one aspect of behaviour or relating to local behaviours, to be composed, building a global view from a number of local views. This may be addressed in a hierarchy, for example, with local verification giving rise to assurances about regional behaviour that can then be composed to give some assertion about global properties. It is likely that increasing levels of abstraction will be needed as we progress up the hierarchy, especially taking efficiency and response time into account. This remains a research challenge; whilst it is clear that boolean algebra provides the basis for composing verification results of qualitative analysis which lead to true/false answers, dealing with quantitative results of verification is more complex.

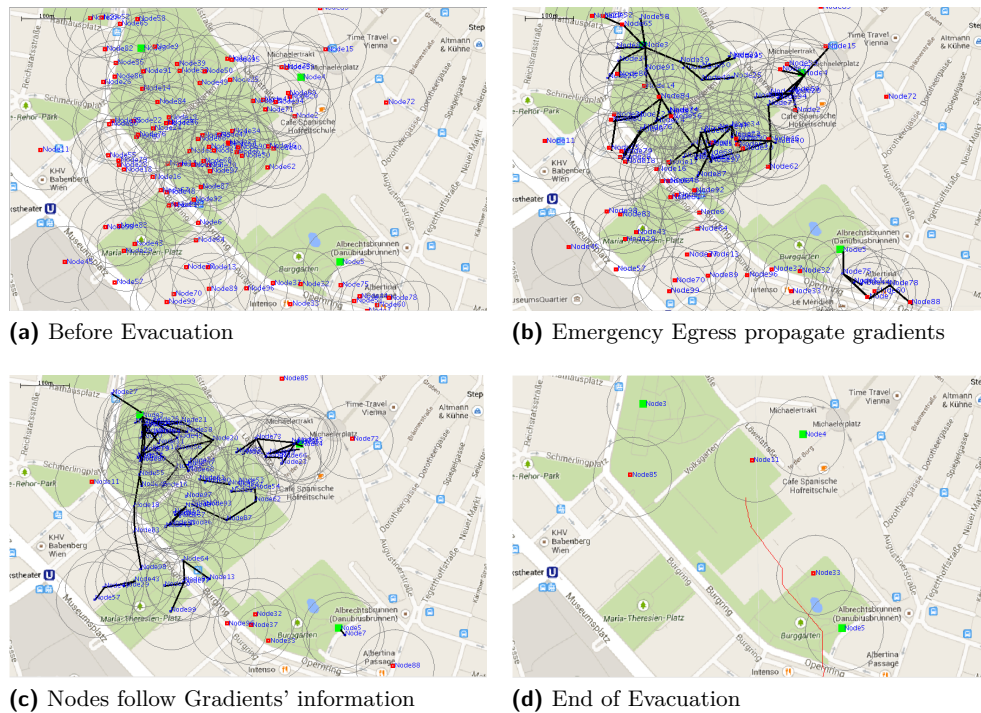
Another interesting approach for future work will be to combine statistical approaches with verification. For example, monitoring and verification could be combined with machine learning techniques to “learn” when a change in the system is likely to lead to an acceptable adaptation, or to guide adaptations in response to changes outside the system.

4.2.4 Subgroup II: Specification and verification of spatial self-organising patterns in CAS

In recent years a number of different frameworks have been proposed for the engineering of CAS. Here we particularly focus on the framework of self-organising patterns proposed in [16] and illustrated in Figure 2. Examples of these patterns include:

Spreading: a copy of the information (received or held by an agent) is sent to neighbours and propagated over the network from one node to another. Information spreads progressively over the system.

Aggregation: information is distributively processed in order to reduce the amount of information and to obtain meaningful information. Aggregation consists of locally applying



■ **Figure 3** Four snapshots of the emergency egress scenario: The emergency exits are the green boxes, the people that have not been reached yet are indicated by red boxes and the circles around people show the radius within which they can get in touch with their neighbours. After some time the dark lines indicate the gradient structure.

a fusion operator to synthesise macro information (filtering, merging, aggregating, or transforming).

Gossip: in large-scale systems, agents need to reach an agreement, shared among all agents, with only local perception and in a decentralised way. Information spreads to neighbours, where it is aggregated with local information. Aggregates are spread further and their value progressively reaches the agreement.

Gradient: information spreads from the location where it is initially deposited and aggregates when it meets other information. During spreading, additional information about the sender's distance and direction is provided: either through a distance value (incremented or decremented); or by modifying the information to represent its concentration (lower concentration when information is further away).

These patterns may be combined to design a system with collective adaptive agents who achieve a desired high-level outcome. For example, consider a scenario of emergency egress in a city. People are assumed to have handheld devices on which they can receive real-time information about the directions to follow to the nearest exit. This information is propagated via their neighbours. However, these devices have only a limited radius for local communication. The idea is therefore to create a dynamic ad-hoc network that aims to eventually reach everyone and provide the required information using dynamic gradients. Figure 3 shows four snapshots of a particular evolution of the system in the initial state and some later times.

When we consider verification there are a number of properties that are of interest in this case study. We mention just a few examples.

- As the mechanism of the communication is dependent on the gradient, it is important to assess its functional properties such as whether all nodes will be reachable.
- By chemotaxis, the agent should follow the gradient through the shortest path. Verification can assess whether all agents will reach the source of the gradient.
- We can also consider quantitative aspects of the induced behaviour such as the speed with which information is spread, the bandwidth consumed and the probability of reaching a certain fraction of people within a certain time.
- Spatio-temporal properties are also of concern such as whether at some future time all reachable nodes receive the gradient information (the spatial dispersion of information at a given time), or conversely at some given location all reachable nodes will receive the gradient information within a given time.
- Invariant properties may be important such as ensuring that the shortest paths are built and that the generated gradient structure does not have loops.

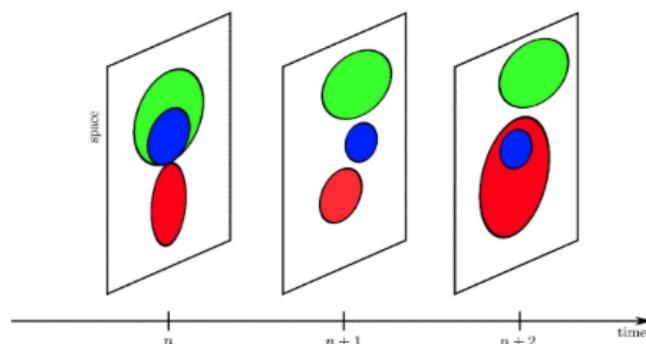
When an application is developed based on such patterns, the details of implementation are often delegated to an agent or service that implements the pattern in question. Therefore it is important that we have the means to independently verify the properties of the supplied patterns (local behaviours) as well as checking the emergent properties at the global level. In particular, it would be hugely beneficial to be able to develop mechanisms for compositional verification to use the verified properties of each pattern to derive the properties of the higher-level patterns, and ultimately properties of the applications build using those patterns. For example, considering the high-level patterns in Figure 2, is it possible to derive the correctness of a flocking pattern in a straightforward manner given the correctness of the repulsion pattern used to implement it?

Spatio-temporal verification via model-checking

Properties of different patterns at the collective level emerge from the coordination at the local level. Considering the example of dynamic gradients used to guide people to the nearest exits in an emergency situation such as that shown in Figure 3 [15], various global properties of the collective mechanism are of interest. For example, as already mentioned, one might like to be sure that all people involved at any time do receive gradient updates on directions within a given time.

Some such properties are spatio-temporal in nature and spatial [12, 10] and spatio-temporal model-checking [11] could be one of the techniques to be considered to automatically verify such properties. Spatio-temporal model-checking is a technique that requires a spatio-temporal model, on the one hand, and a spatio-temporal property, on the other. One way in which the model can be conceived is to consider it as a composition of a Kripke structure (S, T) to model the temporal evolution of the system and a spatial model for each state of the Kripke structure that reflects the spatial situation at the particular time. The latter can also be imagined as a “snapshot” of the spatial situation in a state of the Kripke structure. The spatial model can be conveniently chosen to be a closure space (X, C) , where X is a set of points, and C a closure operator which has its origin in topological spatial logics [26]. Such a choice covers a wide range of specific choices for the spatial structure, encompassing graphs, regular grids and also images similar to those shown in Figure 3.

Spatio-temporal properties address both the evolution in time and the spatial properties of a point in the model, i.e. a point (node) in the space at a particular state in the Kripke



■ **Figure 4** Schematic view of the evolution of a temporal path.

structure. In each possible world there is a different valuation of atomic propositions, inducing a different “snapshot” of the spatial situation which “evolves” over time. This is made clear along a temporal path. A path in the Kripke structure denotes a sequence of digital pictures indexed by instants of time. This is illustrated more schematically in the Figure 4, showing three different states of the Kripke structure at time step n , $n + 1$ and $n + 2$.

Spatio-temporal operators in STLCS (Spatio-Temporal Logic for Closure Spaces) [10] feature the usual Boolean operators (negation, disjunction etc), the CTL path quantifiers A (“for all paths”), E (“exists a path”), which must be followed by path-specific temporal operators XF (“in the next step”), $F1 U F2$, (“eventually $F2$ holds, but until then $F1$ must hold”), where F , $F1$ and $F2$ are STLCS formulas, and the spatial operators closure C and spatial until $F1 S F2$ (“ $F1$ surrounded by $F2$ ”). The two derived temporal operators G (“always”) and F (“eventually”) are also very useful.

Let us proceed with a few simple examples.

- Consider the STLCS formula $EG(\text{green } S \text{ blue})$. This formula is satisfied at a point x in the graph, associated with the initial state s_0 , if there exists a (possible) evolution of the system, starting from s_0 , in which point x is *always*, i.e. in every state in the path, *green* and surrounded by *blue*. The prototype spatio-temporal model-checker described in [10] will return (or colour) all the points x that satisfy the formula.
- A further, more complicated, nested example is the STLCS formula

$$EF(\text{green } S (AX \text{blue})).$$

This formula is satisfied at a point x in the graph associated with the initial state s_0 , if there is a (possible) evolution of the system, starting from s_0 , in which point x is eventually *green* and surrounded by points y that, for every possible evolution of the system from then on, will be *blue* in the next step.

- A simple example concerning the emergency egress case is the formula $AF(!\text{red } S \text{ false})$, where $!$ denotes negation. This formula is satisfied at a point x in the initial state if all possible evolutions of the system eventually reach a state in which there are no red points. Recall that red points correspond to nodes representing people who did not receive the directions to the nearest exit. So when this formula is satisfied it means that there is a point in time at which all people are being updated by the system. The particular expression $!\text{red } S \text{ false}$ is satisfied if none of the pixels are red because the surround operator is a spatial variant of a weak until operator.

This is an area of on-going work and there are a number of open issues for this line of research.

- Are the STLCS spatio-temporal operators sufficient to express the spatio-temporal properties that are relevant to the self-organising patterns used to design and implement CAS?
- If not, which other operators would be needed? Here we can think of operators that address performance aspects, for example to express that the probability that 90% of the people in the emergency egress example have been reached within a certain time-bound T , or more collective aspects such as that a set of points satisfies a certain property.
- What would be the right set of basic operators that on the one hand provide satisfactory expressiveness, or at least cover an interesting class of properties, and on the other hand are such that efficient model-checking algorithms can be found to verify them?
- Which derived operators and property templates are convenient to facilitate formulation of relevant properties in this context?

Much interesting and challenging work remains to find answers to these questions. Furthermore, there are other proposals that consider spatial and spatial-temporal logics. As an example, we would like to mention the spatial signal temporal logic [8]. This is a linear logic for the specification of behavioural properties of continuous signals which has recently been extended with some spatial operators. This logic has been applied in the domain of epidemiology to analyse the spread of a virus.

4.2.5 Concluding remarks

An important aspect of the engineering of CAS is providing evidence that the system requirements are satisfied, despite the uncertainty that may affect the system, its goals, and its environment. In this working group, we primarily focussed on verification of CAS to assess how well a system meets its specification or requirements. The specifics and characteristics of CAS poses several challenges to the problem of verification, including:

- How to express emergent properties of CAS and verify them?
- How to provide evidence in the face of uncertainty, in particular structural uncertainty, where the complete structure of the model of the system may not be known?
- How to enable runtime verification for CAS for which a global model is not available?
- How to enable compositional verification that uses verified properties of patterns at lower levels to derive the properties of higher-level patterns and ultimately properties of CAS built using those patterns?
- How to express spatio-temporal properties relevant for self-organising systems used to design and implement CAS?
- How to blend offline with online verification to provide the evidence that the system requirements are satisfied during the entire lifetime of CAS?

Whilst we enjoyed and benefited from a number of stimulating discussions around the topic of verification of CAS during the Dagstuhl seminar 14512, the time available did not allow us to make any significant developments beyond deepening our understanding and mutual appreciation. Nevertheless, the discussion helped us to hone our ideas and identify a number of exciting topics for future research, several of which are now being actively pursued by members of the working group.

References

- 1 Eugène Asarin, Alexandre Donzé, Oded Maler, and Dejan Nickovic. Parametric identification of temporal properties. In *Runtime Verification – Second International Conference, RV 2011, San Francisco, CA, USA, September 27–30, 2011, Revised Selected Papers*, volume 7186 of *Lecture Notes in Computer Science*, pages 147–160. Springer, 2011.
- 2 C. Baier and J.P. Katoen. *Principles of Model Checking*. MIT press, 2008.
- 3 Ezio Bartocci, Luca Bortolussi, Laura Nenzi, and Guido Sanguinetti. On the robustness of temporal properties for stochastic models. In *Proceedings Second International Workshop on Hybrid Systems and Biology, HSB 2013, Taormina, Italy, 2nd September 2013.*, volume 125 of *EPTCS*, pages 3–19, 2013.
- 4 Luca Bortolussi and Jane Hillston. Fluid model checking. In *CONCUR 2012 – Concurrency Theory - 23rd International Conference, CONCUR 2012, Newcastle upon Tyne, UK, September 4–7, 2012. Proceedings*, volume 7454 of *Lecture Notes in Computer Science*, pages 333–347. Springer, 2012.
- 5 Luca Bortolussi and Roberta Lanciani. Model checking Markov population models by central limit approximation. In *Quantitative Evaluation of Systems - 10th International Conference, QEST 2013, Buenos Aires, Argentina, August 27–30, 2013. Proceedings*, volume 8054 of *Lecture Notes in Computer Science*, pages 123–138. Springer, 2013.
- 6 Luca Bortolussi and Roberta Lanciani. Stochastic approximation of global reachability probabilities of Markov population models. In *Computer Performance Engineering – 11th European Workshop, EPEW Florence, Italy*, volume 8721 of *Lecture Notes in Computer Science*, pages 224–239. Springer, 2014.
- 7 Luca Bortolussi, Dimitrios Millios, and Guido Sanguinetti. Smoothed model checking for uncertain continuous time Markov chains. *CoRR*, abs/1402.1450, 2014.
- 8 Luca Bortolussi and Laura Nenzi. Specifying and monitoring properties of stochastic spatio-temporal systems in signal temporal logic. In *VALUETOOLS 2014*, In Press.
- 9 Radu Calinescu, Carlo Ghezzi, Marta Z. Kwiatkowska, and Raffaella Mirandola. Self-adaptive software needs quantitative verification at runtime. *Commun. ACM*, 55(9):69–77, 2012.
- 10 Vincenzo Ciancia, Stephen Gilmore, Diego Latella, Michele Loreti, and Mieke Massink. Data verification for collective adaptive systems: spatial model-checking of vehicle location data. In *2nd FoCAS Workshop on Fundamentals of Collective Systems*, IEEE Eight International Conference on Self-Adaptive and Self-Organizing Systems, page to appear. IEEE Computer Society, 2014.
- 11 Vincenzo Ciancia, Gianluca Grilletti, Diego Latella, Michele Loreti, and Mieke Massink. A spatio-temporal model-checker, <http://blog.inf.ed.ac.uk/quanticol/technical-reports/>. Technical report, The QUANTICOL project, 2014.
- 12 Vincenzo Ciancia, Diego Latella, Michele Loreti, and Mieke Massink. Specifying and Verifying Properties of Space. In Springer, editor, *The 8th IFIP International Conference on Theoretical Computer Science, TCS 2014, Track B*, volume 8705 of *Lecture Notes in Computer Science*, pages 222–235, 2014.
- 13 Edmund M Clarke, Orna Grumberg, and Doron Peled. *Model checking*. MIT press, 1999.
- 14 Bram Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.
- 15 Jose Luis Fernandez-Marquez and Giovanna Di Marzo Serugendo. Assessment of robustness of spatial structures in crowd steering scenarios. Technical Report SAPERE TR.WP2.2013.07, SAPERE Project, 2013.
- 16 Jose Luis Fernandez-Marquez and Giovanna Di Marzo Serugendo. From self-organizing mechanisms to design patterns to engineering self-organizing applications. In *7th IEEE*

- International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2013, Philadelphia, PA, USA, September 9-13, 2013*, pages 267–268. IEEE Computer Society, 2013.
- 17 Carlo Ghezzi, Claudio Menghi, Amir Molzam Sharifloo, and Paola Spoletini. On requirement verification for evolving statecharts specifications. *Requir. Eng.*, 19(3):231–255, 2014.
 - 18 Thomas A. Henzinger. Quantitative reactive modeling and verification. *Computer Science – R&D*, 28(4):331–344, 2013.
 - 19 M. Usman Iftikhar and Danny Weyns. Activforms: Active formal models for self-adaptation. In *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2014*, pages 125–134, New York, NY, USA, 2014. ACM.
 - 20 Diego Latella, Michele Loreti, and Mieke Massink. On-the-fly fast mean-field model-checking. In *Trustworthy Global Computing – 8th International Symposium, TGC 2013, Buenos Aires, Argentina, August 30–31, 2013, Revised Selected Papers*, volume 8358 of *Lecture Notes in Computer Science*, pages 297–314. Springer, 2013.
 - 21 Radu Mardare, Luca Cardelli, and Kim G. Larsen. Continuous Markovian logics – axiomatization and quantified metatheory. *Logical Methods in Computer Science*, 8(4), 2012.
 - 22 Dongyu Qiu and Rayadurgam Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 367–378. ACM, 2004.
 - 23 Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology. In *Computational Methods in Systems Biology, 6th International Conference, CMSB 2008, Rostock, Germany, October 12-15, 2008. Proceedings*, volume 5307 of *Lecture Notes in Computer Science*, pages 251–268. Springer, 2008.
 - 24 Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 25(12), 2009.
 - 25 Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. Continuous valuations of temporal logic specifications with applications to parameter optimization and robustness measures. *Theor. Comput. Sci.*, 412(26):2827–2839, 2011.
 - 26 Johan van Benthem and Guram Bezhanishvili. Modal logics of space. In Marco Aiello, Ian Pratt-Hartmann, and Johan van Benthem, editors, *Handbook of Spatial Logics*, pages 217–298. Springer, 2007.
 - 27 Danny Weyns, Nelly Bencomo, Radu Calinescu, Javier Camara, Carlo Ghezzi, Vincenzo Grassi, Lars Grunske, Poala Inverardi, Jean-Marc Jezequel, Sam Malek, Raffaella Mirandola, Marco Mori, and Giordano Tamburrelli. Perpetual assurances in self-adaptive systems. In *Assurances for Self-Adaptive Systems, Dagstuhl Seminar 13511*, 2014.

4.3 Humans-in-the-Loop in Collective Adaptive Systems

Jake Beal, Peter Lewis, Stefano Mariani, Jeremy Pitt, Nikola Serbedzija, Franco Zambonelli

License © Creative Commons BY 3.0 Unported license

© Jake Beal, Peter Lewis, Stefano Mariani, Jeremy Pitt, Nikola Serbedzija, and Franco Zambonelli

4.3.1 Introduction

Autonomous and autonomic systems have proved highly effective for self-* management of resource allocation in open, distributed computer systems and networks. Examples range from the ‘Ur’ application of autonomic systems in data centre management [8], to autonomous

virtual power plants for autonomous energy supply [1], to self-organising electronic institutions for resource provision and appropriation in multi-agent systems [21].

These systems are exemplars of collective adaptive systems (CAS): collective, because it requires cooperative, coordinated, synchronised or orchestrated effort of individual entities acting as a group; and adaptive, because it may change its shape, structure, functions, components, rules etc. at run-time, either reactively in response to changes in the environment, or pro-actively, in anticipation of such changes. The operation of such systems is, not unexpectedly, largely if not completely hidden from human users, and may be considered as being composed of purely ‘technical’ components. It is the speed, frequency and complexity of decision-making that precludes operator intervention, and necessitates these autonomous and autonomic approaches to control.

However, leaving data centre management or high-frequency trading algorithms to their own devices (as it were) is one thing, but it is something different when considering so-called ‘smart’ systems (smart homes, smart grids, smart cities, etc.), where the decision-making has a direct impact on qualitative human concerns, or actively requires the intervention of human agency.

The key question then, is: how can the design principles and operating principles for collective adaptive systems composed purely of ‘technical’ components be successfully transferred to resolve corresponding problems in *socio-technical* collective adaptive systems, i.e. CAS composed of both ‘human’ and ‘technical’ components, and involving human-human interaction, computer-mediated human-human interaction, human-technical component interaction (technical components including sensor, device, software agent, robot, etc.), and technical-technical component interaction. In other words, these are systems with ‘humans in the loop’, in which people interact with an electronically saturated infrastructure, or with each other through an electronically-mediated interface, especially when trying to achieve some collective action or common purpose.

This Working Group was constituted to address this precise question. In three main working sessions, we primarily considered three issues: scenarios which highlight the human factors that have to be taken into account in ‘programming’ socio-technical CAS for smart-* applications; a *general intervention framework* that might be used at design-time, run-time or even both, to account for these factors given the special operating conditions of socio-technical CAS; and finally a preliminary ontology for the specific *shaping mechanisms* which might be used to instantiate the general framework.

This report is structured with summaries of each working session, followed by some observations on research challenges and outcomes, and some concluding remarks. Note that for the purposes of this report, the term *socio-technical CAS* is taken to denote the same type of object as *CAS with “Humans-in-the-Loop”*.

4.3.2 ‘Programming’ Socio-Technical CAS

Modern ICT systems are de facto large-scale socio-technical systems whose goal is to collectively and adaptively behave in certain ways. In this context, the focus of the first discussion sessions was to analyze systematically the peculiar differences that such systems exhibit with respect to traditional ICT systems in engineering their behaviours, starting from existing works with user-centric aspects in the system design and or operation [21, 4, 5, 30, 2, 18, 9, 25, 16]. The session then focused on Humans-in-the-Loop from the perspective firstly of scenarios, and secondly of human factors, i.e. those aspects of human behaviour and psychology which distinguish socio-technical CAS from ‘technical’ CAS.

Scenarios

Smart Cities. It is an increasingly common practice to prefix the qualifier ‘smart’ to particular applications. The applications include smart grids (including smart meters), smart homes, smart cars, and so on. ‘Smart’, in this context, has come to mean systems that are characterised by three ‘i’s: instrumented, interconnected and intelligent. These applications all exhibit collectivity, adaptivity, and indeed other self-* computing properties. The various applications converge in the unifying concept of the Smart City, in which multiple interacting CAS will enable the energy distribution, transportation systems, and other city-wide infrastructure to continuously adapt and cope with the dynamism of the environment (changing traffic conditions, stochastic energy supply and demand, utilisation of common urban spaces, etc.).

However, there is, arguably, a fourth ‘i’ – interactive. All these applications also, at some point, have ‘humans in the loop’ for decision-making and even computation, have people as the recipients, as well as data sources, of a service (for example, participatory sensing applications), or have people’s goals, benefits or values as their focal point (for example, fairness and sustainability in energy or water distribution). Therefore this ecosystem is essentially one of socio-technical applications, and there is a requirement to take the people into account – in short, to recognise that Smart Cities are also places where citizens have to live.

In respect of this, the interleaving of autonomic, adaptive and ubiquitous computing, providing the foundations for self-* computing properties, and human-infrastructure interaction, providing the basis for innovative interfaces, affordances and ergonomics, will be critical in the design, and co-design, of ‘fit for purpose’ ‘user friendly’ socio-technical collective adaptive systems, and sub-systems, for Smart Cities.

Shared Living Spaces. Any shared living space, such as a communal flat, an open-plan office, or even a public space such as a park, require people to share a common space. This is a collective adaptive system: the ambience of the living space is both a goal of and function of the collective; but many aspects of the collective change over time, in particular the human ‘components’, but also the mutually-agreed conventional rules governing the use of the shared space.

Furthermore, violation of (implicitly or explicitly stated) these conventional rules, or social norms, can cause instances of incivility [22]. Such incivility, characterised by a low-intensity form of deviance from accepted norms, can be difficult to detect and resolve, but is also very harmful for the people who experience it regularly. Therefore, it is a pressing problem in both ergonomics and urban planning to reduce the negative side-effects of incivility.

One technological solution that has been proposed for addressing the incivility problem, is MACS (Affective Conditioning System): a system that attempts to avoid, reduce and/or resolve incivility in workplace environment before it escalates into a higher-intensity situation, e.g. conflict or aggression [24]. MACS is intended to emphasise stakeholder engagement and empower collective choice: firstly by avoiding micro-management, as incivility episodes are resolved between stakeholders (i.e. the occupants of the shared space themselves), and only as a last resort by appeal to higher authorities; and secondly by providing social support, through a network of communication and mutual obligations, via the collective selection, monitoring and enforcement of the stakeholders’ own social norms and pro-social processes such as forgiveness [28].

In MACS, the shared living space is envisioned as a *common pool resource* which we seek to manage according to the institutional design principles of Elinor Ostrom [17]. In this

respect, the metaphor we are pursuing is that the (intangible) ‘office ambience’ is a pooled resource which the office occupants can deplete by anti-social behaviour and re-provision by pro-social behaviour. Furthermore, what is (and is not) anti-social behaviour is determined by the occupants themselves – a specific instantiation of Ostrom’s third principle (that those affected by collective choice arrangements participate in their selection). Consequently, MACS implements a voting system for social norms, which allows for those (and only those) admitted to a shared space to vote positively or negatively for a norm. It also allows people to suggest new norms, as the dynamic nature of offices might mean there is a constant need to change norms, so MACS provides support for this process.

The Social Computer. The idea of the *social computer* [11] or *social computation* is for the designers of applications which synthesise the intelligence of human and automated computational units to tackle so-called ‘wicked’ problems [26]. Prototype examples of this type of collective adaptive system can be seen in crowdsourcing applications, such as Amazon’s Mechanical Turk, etc., with people doing the work that computers are not so good at, and computers the work at which people are not so competent.

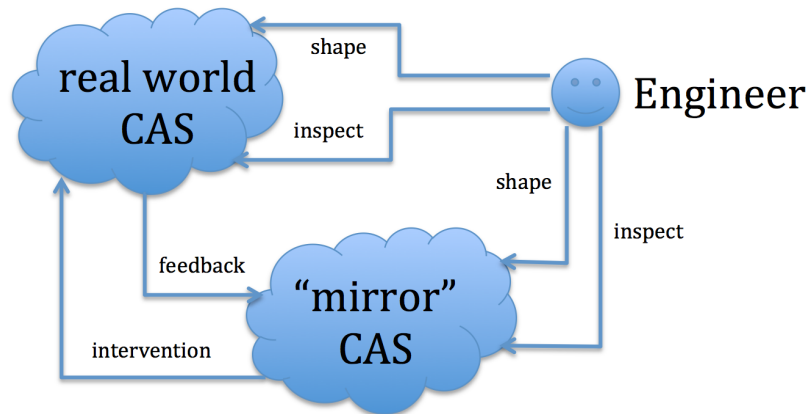
One of the prime examples of the potential of such systems is ReCAPTCHA, which leverages a security requirement to supplement the incompleteness of optical character recognition in digitising text. Ideally, the design and operation of CAS with humans-in-the-loop will thrive on achieving such externalities, i.e. where work required for one purpose will provide beneficial input for another.

Human Factors

Although this WG was composed of scientists and technologists rather than psychologists or sociologists, there was some experience of human-computer interaction and the discussion of (potentially disruptive) human factors identified the following features:

- People engage in micro-level behaviours, actions and decision-making which produces (potentially unexpected) macro-level outcomes, i.e. socio-technical CAS are also complex systems [7];
- Participation and engagement are critical in empowerment and politics but ‘attention’ remains a limited resource [6];
- Population change over times, and with it so do attitudes, cultures, fashions, etc.;
- People have different access to, perception of, and skills with technology;
- People don’t comply or not-comply with policies, they react to incentives implied by the policy [10], or find ways of interpreting a policy so that they consider themselves compliant;
- People are not equivalent to programmable components [11];
- People innovate themselves, and in particular utilise generative technology in unexpected ways [31];
- There is a trade-off between values, incentives and principles, often manifested in the form of social capital [20];
- A ‘spectrum’ of errors is to be expected, from accidents and triage, through to low-grade non-compliance and (regrettably) absolute malice;
- Governance is critical.

One conclusion drawn was that the design, deployment and evaluation of socio-technical CAS should be a multi-disciplinary, if not trans-disciplinary, process; and that the disciplines of psychology (and adaptation) [13, 12, 15], sociology (and collective intelligence) [29] and



■ **Figure 5** 3I Life-Cycle in Design and Monitoring of Socio-Technical CAS.

legal anthropology [3] can all make essential contributions to the study of socio-technical CAS.

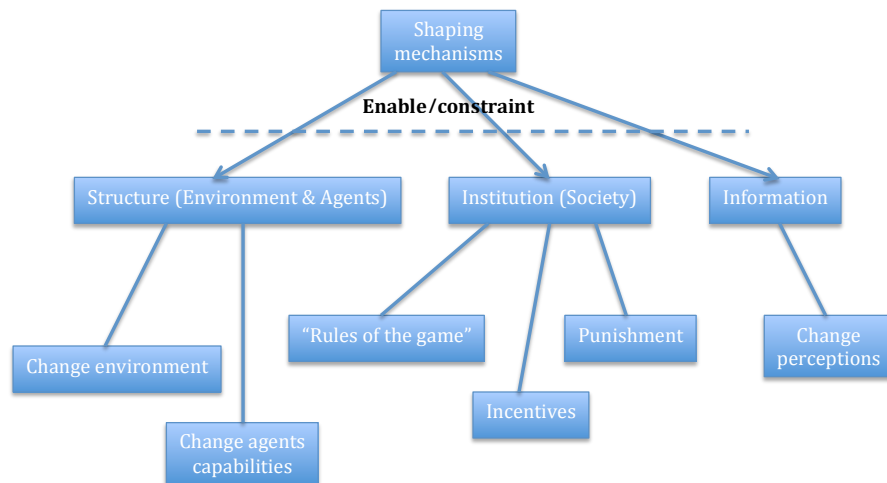
4.3.3 (Towards A) General Intervention Framework

CAS mostly are existing systems we have to simultaneously model (observe) and engineer (build) so as to intervene (influence) on the behaviour of both the collectives and the individuals. Thus, there is a need to embrace (model and engineer) uncertainty, error and unpredictability of behaviours and interactions, with the goal of drawing the boundaries within which individuals as well as collectives can behave and interact, be influenced and influence each other and the CAS as a whole. The main challenge here is that engineers cannot program the individuals (especially if humans); the path to pursue is that, on the other hand, engineers may shape the environment where individuals live to influence the way in which they behave and interact.

When humans enter the picture, it is no longer true that technology is neutral w.r.t. individuals' principles and values and in terms of their reaction to it (emotional, economic, etc.). Further, the impact of technology is also unpredictable and/or uncertain. Thus, there is a need to take into account individuals values at different scales (individuals, collective, CAS) and to recognize that whenever you offer a given technology to people you are simultaneously enabling and constraining (shaping) their capabilities of behaving and interacting within the CAS. The main challenge here is that failure, error, unforeseen emergent phenomena and misbehaviour (such as incivility) are an intrinsic part of the system.

Due to unavoidable uncertainty in modelling CAS and limited intervention opportunities and capabilities for engineers, simulation, model checking, etc. are no longer enough to ensure CAS sustainability. Thus, there is a need for live testing environments, running in a "mirror world" reflecting real-world data, behaviours and interactions while protecting real individuals, continuously providing feedbacks on CAS behaviour and absorbing patches, upgrades, intervention of engineers. The main challenge here is how to conceive, design and deploy these test environments.

The conclusion of the discussion led to the (preliminary) proposal of the the 3I life-cycle (Inspection-Innovation-Intervention) for the the design and operational management and governance of socio-technical CAS (see Figure 5). The idea is that system designers can inspect the behaviour of the system, innovate improvements which are tested in the "mirror" CAS or some sub-system of the CAS, and then make timely interventions in the "actual" CAS.



■ **Figure 6** A Shaping Mechanism Ontology for the General Intervention Framework.

Essentially, though what is required here is *general intervention framework* for socio-technical CAS. The specific mechanisms to shape or make interventions are discussed in the next section.

One further observation of this discussion, though, was that if any socio-technical CAS should be its own testbed, then consideration should be given to instrumentation, observability and running trials for sub-systems. In one sense, this could provide the basis for *evidence-based policy-making*, although how to design and run a controlled double-blind randomised experiment with policies (rather than, say, health treatments) – and not fall foul of ethical requirements – remains an open question.

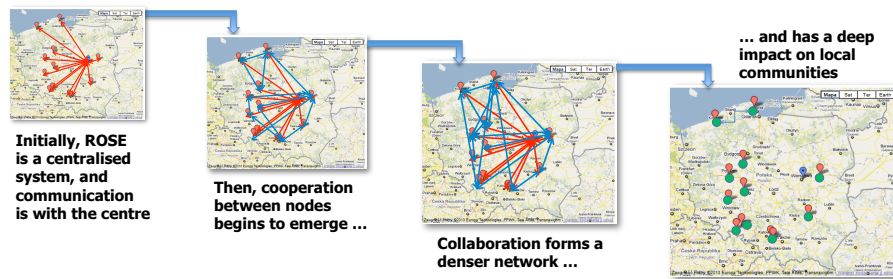
4.3.4 Shaping Mechanisms

The discussion in this session focused on the mechanisms for shaping intervention in the ‘landscape’ of a socio-technical CAS. Throughout, the group were seeking an analogy to the way that a city planner might intervene in traffic shaping by various mechanisms: for example physical mechanisms such as roundabouts (replacing traffic lights) and speed bumps, and policy-based mechanisms such as congestion charging and low-emission zones.

A Preliminary Ontology

Figure 6 illustrates a preliminary ontology of shaping mechanisms to instantiate the general intervention framework for designing, deploying and managing socio-technical CAS at run-time.

Figure 6 identifies three categories of mechanism, environmental (structural), institutional and informational. Environmental mechanisms includes changing the environment or the components. Institutional mechanisms includes changing the conventional rules, the incentives or the punishments (the system of retributive justice dealing with sanctions, or fear of sanctions (deterrence)). One informational mechanism is in the ‘sensory’ apparatus of the CAS. This has been referred to an *interoceptive collective awareness*, i.e. it is a sense that comes from within the collective as a whole, is concerned with the well-being of the collective, and is a precursor to collective action [20].



■ **Figure 7** Emerging Meso-level Structures in Project ROSE.

As an exemplar of a shaping mechanism, we briefly describe the experience of the ROSE project in Poland.

Exemplar: Project ROSE

To illustrate the principles and potential of a shaping mechanism for socio-technical CAS, we examine the experience of Project ROSE (Regional Centres of E-learning). The project started in 2004 at the Institute for Psychology of Informatics and Communication, directed by Professor Andrzej Nowak, of Warsaw School of Social Sciences and Humanities. The challenge was to promote the use of ICT, especially the Internet, in education in Poland. However, the rapid advances of ICT usually render any non-evolving educational program obsolete in just a few years. The solution was to create a learning community in the form of an expanding network of teachers that constantly adapt to new developments in ICT.

ROSE was based on the idea that teacher enhancement is a social change process rather than a transfer of knowledge. The Bubble Theory of Social Change [12] specifies how a sustainable social change may be achieved by concentrating on changing fragments of social networks (clusters or bubbles) rather than separate individuals. ROSE is therefore a mixture of face-to-face workshops and Internet mediated interactions. The workshops enabled the teachers to learn to collaborate with each other and to develop trust. From each workshop several individuals were selected as natural leaders to seed the ROSE network. After the initial workshop the training was conducted over the Internet using an e-learning platform. The communication structure resembled a star with the university performing the role of the central hub, and each school being a spoke (see Figure 7).

The leaders in each school initially worked with teachers from their own school but in the next stage schools already in ROSE collaborated with each other in the preparation of programmes for other schools. Meso-level structures (formal groupings with rules, roles, processes, designated groups responsible for decisions in specific areas, etc.; and informal groupings based on friendship circles, interest groups, etc.) emerged as clusters of collaborating schools, local administration and businesses etc. Afterwards, the meso-level structures grew stronger and bigger as more common initiatives were undertaken. The role of the university decreased as the network became increasingly decentralized.

This is a demonstration of using institutions as the shaping mechanism for a socio-technical CAS. This exemplar also again emphasises that disciplines from the social sciences, in this case dynamic social psychology [14], have highly relevant and significant insight to offer the development of socio-technical CAS.

4.3.5 Research Challenges

In the final working session, the WG identified and discussed a number of issues that remain unresolved in devising socio-technical CAS. This includes:

- Dispute resolution mechanisms for the various conflicts that may occur when, for example, members belonging to several communities with incompatible goals or notions of fairness collide;
- The “social ergonomics” that will need to be evaluated and refined: for example, even if the macro-objectives emerging at any one time are fair with respect to a society’s common good, and even if fairness is ensured in the long-term for each individual, this will not necessarily imply an acceptable *experience* for each individual in that society. Users may be willing to trade-off optimality for stability;
- The attention deficit: having algorithmic controls at their fingertips, individuals participating in a group may feel that they have no choice but to engage in a process of continuous negotiation and adaptation to rule-sets and social norms. The system’s affordances would engender an open cycle of societal self-adaptations and frequent change, inducing individual and collective stress and fatigue;
- Data: as observed in [27], the power of Big Data and associated tools for analytical modelling “... should not remain the preserve of restricted government, scientific or corporate élites, but be opened up for societal engagement and critique. To democratise such assets as a public good, requires a sustainable ecosystem enabling different kinds of stakeholder in society”.
- Economic security: how vulnerable would such CAS be to ‘hijacking’, by external parties and what could be the consequences? This especially concerns those socio-technical CAS which have an economic dimension and might see work at the edge but value only in the network or middleware [23];
- Governance security: a model of ‘good governance’ must be installed and maintained, and be equally robust to hostile takeover by minority interests;
- Externalities: are there any collateral costs, as well as benefits that such a system would place on society?

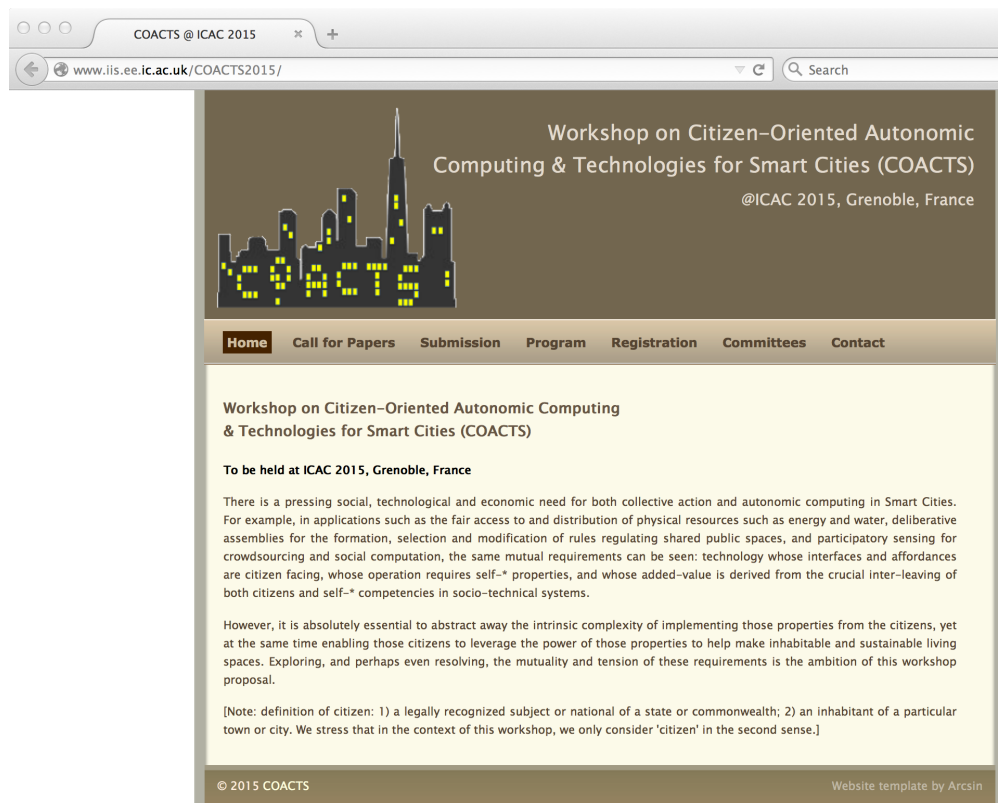
Such questions and the ensuing design requirements must be carefully considered before irreversibly embedding socio-technical CAS in the very fabric of everyday life. Since all possible scenarios cannot be predicted and addressed in advance, the ICT system itself must be sufficiently flexible to enable its evolution in parallel to the society it serves, which is why not just the 3I life-cycle is so important, but so too the integrity of the designers, programmers and managers, highlighting the need for *design contractualism* [19]

4.3.6 Outcomes

At the time of submission, there have been two significant outcomes of the Dagstuhl Seminar and the Working Group on Humans-in-the-Loop in particular.

The first is that the deliberations and discussions of the WG were highly influential in drafting a workshop proposal that has been accepted at ICAC 2015 (International Conference on Autonomic Computing), entitled COACTS (Citizen-Oriented Autonomic Computing and Technologies for SmartCities), see <http://www.iis.ee.ic.ac.uk/COACTS2015/> and Figure 8.

The second is the structure of a paper, co-authored by the WG participants, entitled *Towards a General Intervention Framework for Shaping Socio-Technical Collective Adaptive Systems*, which will be informed by this report.



■ **Figure 8** COACTS at ICAC2015 Website Home Page.

4.3.7 Summary and Conclusions

The WG has made a significant contribution to the Seminar's goal understanding quantitative and qualitative analysis and modelling of CAS through an in-depth discussion of socio-technical CAS, or *CAS with Humans-in-the-Loop*. In particular, two specific contributions can be highlighted:

- a critical analysis of CAS scenarios with 'humans in the loop', and the identification of human factors which need to be taken into account concerning the design and operational principles of such CAS; and
- three innovative proposals for quantitative and qualitative analysis and modelling of socio-technical CAS: (i) a general intervention framework, (ii) the 3I life-cycle, and (iii) the ontology of shaping mechanisms.

As well as identifying some key research challenges, the WG believes that whatever else happens, collective adaptive systems for socio-technical applications with 'humans in the loop' need to be engineered properly, deployed responsibly, and managed with 'good governance'.

References

- 1 Anders, G., Steghöfer, J.P., Siefert, F., Reif, W.: A trust- and cooperation-based solution of a dynamic resource allocation problem. In: SASO. pp. 1–10 (2013)
- 2 Beal, J., Dulman, S., Usbeck, K., Viroli, M., Correll, N.: Organizing the aggregate: Languages for spatial computing. CoRR abs/1202.5509 (2012), <http://arxiv.org/abs/1202.5509>

- 3 Casanovas, P., Pagallo, U., Palmirani, M., Sartor, G.: Law, social intelligence, nmas and the semantic web: An overview. In: *AI Approaches to the Complexity of Legal Systems – AICOL*. pp. 1–10 (2013)
- 4 Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J.R., Mellouli, S., Nahon, K., Pardo, T., Scholl, H.: Understanding smart cities: An integrative framework. In: *IEEE Hawaii International Conference on System Sciences*. Maui (HI), USA (2012)
- 5 Conti, M., Das, S., Bisdikian, C., Kumar, M., Ni, L., Passarella, A., Roussos, G., Troster, G., Tsudik, G., Zambonelli, F.: Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber-physical convergence. *Pervasive and Mobile Computing* 8(1), 2–21 (2012)
- 6 Ferscha, A., Paradiso, J., Whitaker, R.: Attention management in pervasive computing. *IEEE Pervasive Computing* 13(1), 19–21 (2014)
- 7 Gell-Mann, M.: What is complexity? *Complexity* 1(1), 16–19 (1995)
- 8 Kusic, D., Kephart, J., Hanson, J., Kandasamy, N., Jiang, G.: Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing* 12(1), 1–15 (2009)
- 9 Lewis, P.R., Marrow, P., Yao, X.: Resource allocation in decentralised computational systems: an evolutionary market-based approach. *Autonomous Agents and Multi-Agent Systems* 21(2), 143–171 (2010), <http://springerlink.metapress.com/content/h4t672m316678605/>
- 10 López, E.: *The Pursuit of Justice: Law and Economics of Legal Institutions*. New York, NY: Palgrave MacMillan (2010)
- 11 Miorandi, D., Maggi, L.: “Programming” social collective intelligence. *Technology & Society Magazine* 33(3), 55–61 (2014)
- 12 Nowak, A., Lewenstein, M., Szamrej, J.: Bable modelem przemian społecznych (bubbles: a model of social transition). *Swiat Nauki (Scientific American Polish Edition)* 12 (1993)
- 13 Nowak, A., Szamrej, J., Latane, B.: From private attitude to public opinion: a dynamic theory of social impact. *Psychological Review* 97, 362–376 (1990)
- 14 Nowak, A., Vallacher, R., Kus, M., Urbaniak, J.: The dynamics of societal transition: modeling non-linear change in the Polish economic system. *International Journal of Sociology* 35, 65–68 (2005)
- 15 Nowak, A., Rychwalska, A., Szamrej, J.: Social, psychological and technological determinants of energy use. *IEEE Technol. Soc. Mag.* 33(3), 42–47 (2014)
- 16 Omicini, A., Mariani, S.: Agents & multiagent systems: En route towards complex intelligent systems. *Intelligenza Artificiale* 7(2), 153–164 (2013), <http://dx.doi.org/10.3233/IA-130056>
- 17 Ostrom, E.: *Governing the commons: The evolution of institutions for collective action*. Cambridge, UK: Cambridge University Press (1990)
- 18 Pitt, J., Bourazeri, A., Nowak, A., Roszczynska-Kurasinska, M., Rychwalska, A., Santiago, I.R., Sanchez, M.L., Florea, M., Sanduleac, M.: Transforming big data into collective awareness. *Computer* 46(6), 40–45 (2013)
- 19 Pitt, J.: Design contractualism for pervasive/affective computing. *IEEE Technol. Soc. Mag.* 31(4), 22–29 (2012), <http://dx.doi.org/10.1109/MTS.2012.2225458>
- 20 Pitt, J., Nowak, A.: Collective awareness and the new institution science. In: Pitt, J. (ed.) *The Computer After Me*, chap. 11. IC Press (2014)
- 21 Pitt, J., Schaumeier, J., Artikis, A.: Axiomatisation of socio-economic principles for self-organising institutions: Concepts, experiments and challenges. *ACM Trans. Auton. Adapt. Syst.* 7(4), 39:1–39:39 (Dec 2012)
- 22 Porath, C., Pearson, C.: The price of incivility. *Harvard Business Review* 91(1-2), 114 (2013)

- 23 Reich, R.: The sharing economy is hurtling us backwards. Salon (February 2015)
- 24 Santos, M., Pitt, J.: Emotions and norms in shared spaces. In: Balke, T., Dignum, F., van Riemsdijk, M.B., Chopra, A. (eds.) COIN. LNCS, vol. 8386, pp. 157–176. Springer (2013)
- 25 Serbedzija, N.B., Fairclough, S.H.: Reflective pervasive systems. TAAS 7(1), 12 (2012), <http://doi.acm.org/10.1145/2168260.2168272>
- 26 Serugendo, G.D.M., Risoldi, M., Solemayni, M.: The social computer. In: Pitt, J. (ed.) *The Computer After Me*, chap. 11. IC Press (2014)
- 27 Shum, S.B., Aberer, K., Schmidt, A., Bishop, S., Lukowicz, P., Anderson, S., Charalabidis, Y., Domingue, J., de Freitas, S., Dunwell, I., Edmonds, B., Grey, F., Haklay, M., Jelasity, M., Karpistenko, A., Kohlhammer, J., Lewis, J., Pitt, J., Sumner, R., Helbing, D.: Towards a global participatory platform: Democratising open data, complexity science and collective intelligence. *European Physical Journal: Special Topics* 214 (2012)
- 28 Vasalou, A., Hopfensitz, A., Pitt, J.: In praise of forgiveness: Ways for repairing trust breakdowns in one-off online interactions. *Int. J. Hum.-Comput. Stud.* 66(6), 466–480 (2008)
- 29 Wood, L., Büscher, M., van Veelen, J.B., van Splunter, S.: Agile response and collaborative agile workflows. *IJISCRAM* 5(3), 1–19 (2013), <http://dx.doi.org/10.4018/ijiscram.2013070101>
- 30 Zambonelli, F.: Toward sociotechnical urban superorganisms. *IEEE Computer* 45(8), 76–78 (2012)
- 31 Zittrain, J.: *The Future of the Internet – And How to Stop It*. New Haven, CT: Yale University Press (2008)

Participants

- Jacob Beal
BBN Technologies –
Cambridge, US
- Lenz Belzner
LMU München, DE
- Luca Bortolussi
Universität des Saarlandes, DE
- Giacomo Cabri
University of Modena, IT
- Rocco De Nicola
IMT – Lucca, IT
- Giovanna Di Marzo Serugendo
University of Geneva, CH
- Vashti Galpin
University of Edinburgh, GB
- Marco Gribaudo
Politecnico di Milano Univ., IT
- Salima Hassas
University Claude Bernard –
Lyon, FR
- Jane Hillston
University of Edinburgh, GB
- Annabelle Klarl
LMU München, DE
- Roberta Lanciani
IMT – Lucca, IT
- Peter Lewis
Aston Univ. – Birmingham, GB
- Michele Loreti
University of Firenze, IT
- Stefano Mariani
Università di Bologna, IT
- Mieke Massink
CNR – Pisa, IT
- Ugo Montanari
University of Pisa, IT
- Laura Nenzi
IMT – Lucca, IT
- Jeremy Pitt
Imperial College London, GB
- Christophe Scholliers
Free University of Brussels, BE
- Hella Seebach
Universität Augsburg, DE
- Nikola Serbedzija
FhG FOKUS – Berlin, DE
- Mirco Tribastone
University of Southampton, GB
- Petr Tuma
Charles University – Prague, CZ
- Danny Weyns
Linnaeus University – Växjö, SE
- Martin Wirsing
LMU München, DE
- Franco Zambonelli
University of Modena, IT

