CONFERENCE HANDBOOK

FM 2015

20TH INTERNATIONAL SYMPOSIUM ON FORMAL METHODS

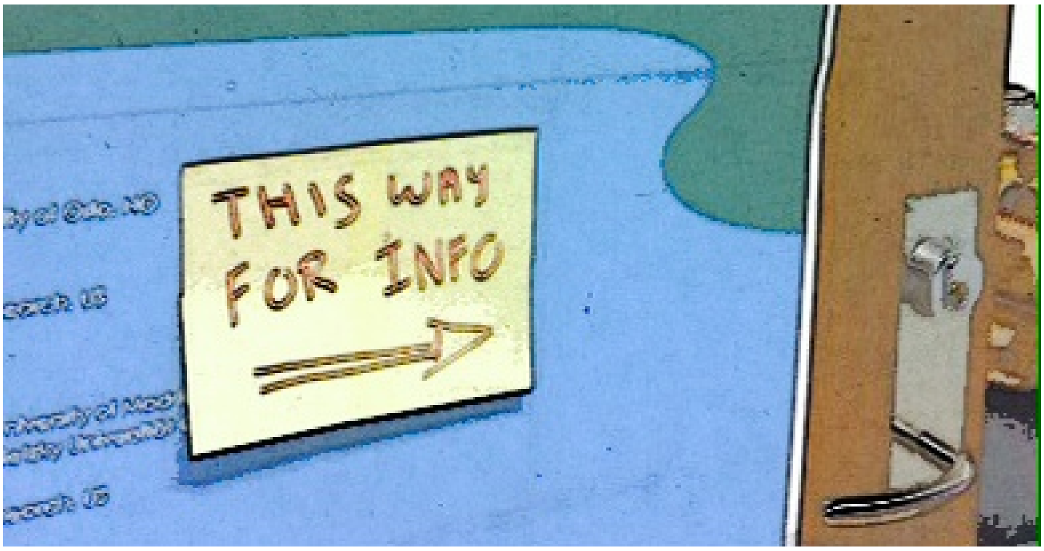
Oslo, Norway

UiO : Department of Informatics
University of Oslo

Formal Methods Europe

FM 2015

Oslo, Norway 22–26 June 2015



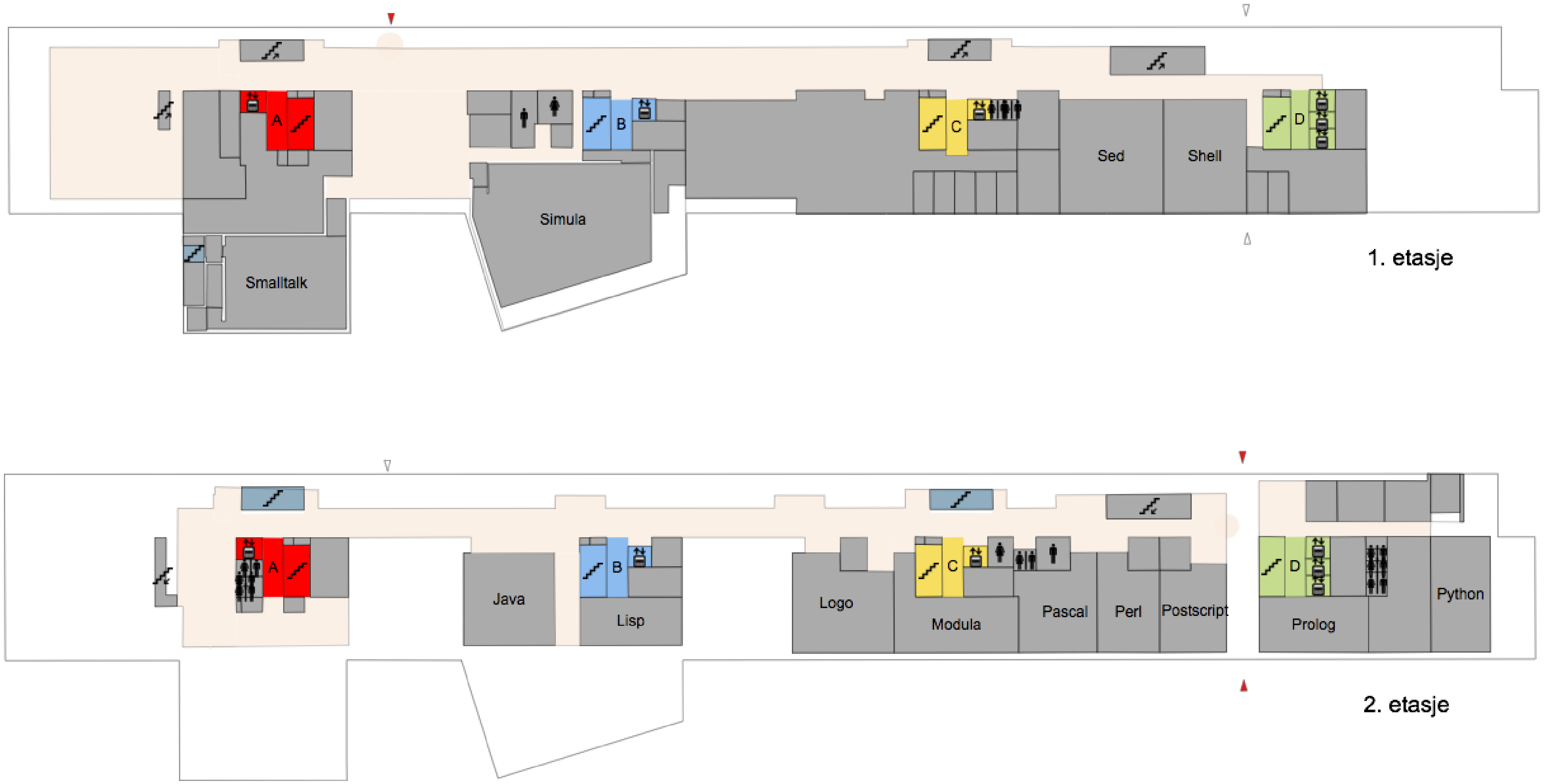


Figure 1: Map of the IFI building

Contents

List of Tables

List of Figures

General Information

Welcome to the University of Oslo

Welcome to the University of Oslo and the Department of Informatics, and to our new venue at the main campus of the university. I am very happy that you have chosen to gather here in Oslo to discuss cutting edge research on verification and formal methods and on the development of robust, reliable, and secure software systems.

Our building is named after Ole-Johan Dahl, who together with Kristen Nygaard is considered to be the father of object-oriented programming. Both Dahl and Nygaard were professors at our department and won the two most prestigious international prizes in computer science, the ACM Turing Award and the IEEE John von Neumann Medal.

Our department is an academically broad ICT department in with approximately 70 faculty members, about 200 registered PhD students and about 150 students finishing their master degree each year. In addition to computer science and communication technology, the scientific activities include e.g. microelectronics, natural language processing and information systems design, and main application areas are medicine and energy. There is a long tradition on formal methods and verification, also initiated by Ole Johan Dahl, who actually was invited speaker at VDM'90. This activity has gradually grown both in size and scope, and with a substantial increased activity in recent years with strong international collaborations, e.g. through EU projects.

I encourage you during conference breaks to have a look around the building, including at the many art installations found inside and outside. I also hope you will find time to enjoy some of what Oslo has to offer, such as the Vigeland Park — a large sculpture park, the Munch Museum, the Opera House and the Holmenkollen ski arena.

I wish you rewarding days at the conference and a pleasant stay in Oslo.

Knut Liestøl
University of Oslo
Department of Informatics
Head of Department

Welcome to FM 2015

This is the 20th edition of the International Symposia on Formal Methods, and the first to be hosted in Norway. FM is the first and largest conference covering the rapidly growing area of formal methods. The Symposia on Formal Methods are organized locally and coordinated by Formal Methods Europe (FME). FME is an independent, worldwide association bringing together researchers and practitioners in formal methods developing computing systems and software. Formal methods differ from many software engineering techniques in that they stress the importance of a rigorous semantic basis for the tools and notations used. Such sound foundations permit the analysis of computing system designs to a depth that is otherwise impossible to achieve. FME aims is to encourage formal methods research and application.

On behalf of the local organizing team, I would like to welcome you to Oslo and FM 2015. Oslo is the capital of Norway with approximately 600.00 habitants and home to the University of Oslo, Norway's main university, which hosts the FM 2015 conference. The city is located with the fjord on one side, and surrounded by forests and hills on the others. It is a fairly modern city with many attractions for visitors, including the Vigeland park, the Munch museum, the Opera house, the Polar and Viking Ship Museums, and the Holmenkollen ski jump. Computer science at the University of Oslo was established by Turing Award winner Ole-Johan Dahl, who was in fact an invited speaker at VDM'90, a precursor to the FM symposia.

This year the event includes two days of workshops and tutorials, a Doctoral symposium, and an Industry Track in addition to the main scientific track. In addition, FM 2015 includes the ceremony for the first FME Fellowship Award, a distinction that will be awarded approximately every three years, during an FME symposium, to researchers and practitioners of formal methods.

We gratefully acknowledge the support of our sponsors for FM2015: the Research Council of Norway, the City of Oslo, the Centre for Software Verification & Validation (CERTUS), the Centrum Wiskunde & Informatica (CWI), the Centre for Resilient Networks & Applications (CRNA), DNV GL, and Microsoft Research.

June is in principle an ideal time of the year to visit Oslo, with a comfortable climate and long Nordic summer nights. However, as I write these words, the Norwegian summer has been inexplicably delayed, and we are still waiting for the heat wave (by local standards) planned for the conference itself. Hopefully it has arrived by the time you receive this booklet, allowing you to maximally enjoy the conference both as an academic and a social event.

Einar Broch Johnsen
University of Oslo
FM2015 General Chair

GENERAL CHAIR:
Einar Broch Johnsen, University of Oslo, NO

PC CHAIRS:
Nikolaj Bjørner, Microsoft Research, US
Frank de Boer, CWI, NL

WORKSHOPS CHAIRS:
Marieke Huisman, Twente University, NL
Volker Stolz, University of Oslo, NO

INDUSTRY DAY CHAIRS:
Ralf Huuck, Red Lizard Software, AUS
Peter Gorm Larsen, Aarhus University, DK
Andreas Roth, SAP, DE

TOOL EXHIBITION CHAIRS:
Richard Bubel, TU Darmstadt, DE
Rudolf Schlatte, University of Oslo, NO

TUTORIAL CHAIRS:
Ferruccio Damiani, University of Torino, IT
Christian Johansen, University of Oslo, NO

DOCTORAL SYMPOSIUM CHAIRS:
Bernhard Aichernig, TU Graz, AT
Alessandro Rossini, Sintef, NO

FINANCE CHAIRS:
Arnaud Gotlieb, Simula Research Labs, NO
Ingrid Chieh Yu, University of Oslo, NO

PUBLICITY CHAIR:
Martin Steffen, University of Oslo, NO

LOCAL ORGANIZATION CHAIRS:
Violet Ka I Pun, University of Oslo, NO
S. Lizeth Tapia Tarifa, University of Oslo, NO

PROGRAM COMMITTEE:
Erika Ábrahám, RWTH Aachen University
Bernhard K. Aichernig, TU Graz
Gilles Barthe, IMDEA Software Institute
Nikolaj Bjørner, Microsoft Research
Marcello Bonsangue, LeidenUniversity
Michael Butler, University of Southampton
Andrew Butterfield, Trinity College Dublin
Ana Cavalcanti, University of York
David Clark, University College London
Frank S. de Boer, CWI
Jin Song Dong, National University of Singapore
Michael Emmi, IMDEA Software Institute
John Fitzgerald, Newcastle University
Nate Foster, Cornell University
Vijay Ganesh, University of Waterloo
Diego Garbervetsky, Dep. de Computación, U. de Buenos Aires
Dimitra Giannakopoulou, NASA Ames Research Center
Stefania Gnesi, ISTI-CNR
Ganesh Gopalakrishnan, University of Utah
Orna Grumberg, Technion, Israel Institute of Technology
Arie Gurfinkel, Carnegie Mellon University
Reiner Hähnle, Technical University of Darmstadt
Klaus Havelund, NASA Jet Propulsion Laboratory
Anne E. Haxthausen, Technical University of Denmark
Ian J. Hayes, University of Queensland
Gerard Holzmann, NASA Jet Propulsion Laboratory
Daniel Jackson, MIT
Cliff Jones, Newcastle University
Gerwin Klein, NICTA and UNSW
Laura Kovacs, Chalmers University of Technology
Marta Kwiatkowska, University of Oxford
Peter Gorm Larsen, Aarhus University
Yves Ledru, Lab. d'Informatique de Grenoble, U. Joseph Fourier
Rustan Leino, Microsoft Research
Martin Leucker, Universität zu Lübeck
Shaoying Liu, Hosei University
Tom Maibaum, McMaster University
Dominique Méry, Université de Lorraine, LORIA
Peter Müller, ETH Zürich
César Muñoz, National Aeronautics and Space Administration
David Naumann, Stevens Institute of Technology
Tobias Nipkow, TU München
José Oliveira, Universidade do Minho
Olaf Owe, University of Oslo
Sam Owre, SRI International

Andrei Paskevich, Université Paris-Sud 11, IUT d'Orsay
Grigore Roşu, University of Illinois at Urbana-Champaign
Kristin Yvonne Rozier, NASA Ames Research Center
Sanjit A. Seshia, UC Berkeley
Natasha Sharygina, Università della Svizzera Italiana
Viorica Sofronie-Stokkermans, Max-Planck Institute for Informatics
Jun Sun, Singapore University of Technology and Design
Kenji Taguchi, AIST
Margus Veanes, Microsoft Research
Ji Wang, National Lab. for Parallel and Distributed Processing
Alan Wassyng, McMaster University
Heike Wehrheim, University of Paderborn
Michael Whalen, University of Minnesota
Jim Woodcock, University of York
Gianluigi Zavattaro, University of Bologna
Pamela Zave, AT&T

Practical Information

Conference Venue

FM 2015 takes place in Ole-Johan Dahl's House, the modern Computer Science building on the main campus of the University of Oslo. Address: Gaustadalléen 23 B 0373 Oslo. See Figure 2 in the Map's section.

Getting There

Metro (T-bane):
From Jernbanetorget (Oslo Central Station) you can take line 4 (Ringen/Storo) or line 6 (Sognsvann). Disembark at Forskningsparken station. See ruter.no for more information. From the station in Ullevål Stadion you can take lines 4 (Bergkrystallen), or 6 (Ringen). Disembark at Forskningsparken (one stop). See ruter.no for more information.

Tram (trikk):
Line 17 or 18 from the stop, Stortorvet (by Glassmagasinet) direction to Rikshospitalet. Disembark at Forskningsparken stop. See ruter.no for more information.

Bus:
Line 23 between Lysaker and Simensbråthen (Ekeberg) stops at Ringveien. Disembark at Gaustad (in Store Ringvei). See ruter.no for more information.

Car:
Turn of Store ringvei by the exit towards the Rikshospitalet University Hospital.

Arrival from the west: turn left in the first and the second roundabout. Drive uphill and take the first exit to the left. You are now in Gaustadalléen, and Ole-Johan Dahl's House is on your lefthand side, approximately 500 metres further down the road.

Arrival from the east: follow the second exit after the Tåsen tunnel. Follow the second exit from the roundabout (slightly to the left). Drive uphill and take the first exit to the left. You are now in Gaustadalléen, and Ole-Johan Dahl's House is on your lefthand side, approximately 500 metres further down the road.

Airport express train (Flytoget):
If you use the Airport express train, disembark at Oslo Central Station (Oslo Sentralstasjon/Oslo S). Then switch to the metro or tram to get to the venue. Please see above for further information.

Train (tog):
If you travel by train (NSB), disembark at Oslo Central Station (Oslo Sentralstasjon/Oslo S). Then switch to the metro or tram to get to the venue. Please visit NSB for further information about train departures. NSB is cheaper than Flytoget, but not as frequent.

Conference Homepage

http://fm2015.ifi.uio.no/

Registration Desk

It is located in the 1st floor (ground floor) by the reception area near the rooms Simula and Smalltalk (See Figure 1 on page 4). The registration desk will be open from 08:15. There will be people attending the desk most of the time during the day.

Internet Access

Visitor wireless network options:

Eduroam. For visitors from universities and other educational institutions:

- Network Name (SSID): eduroam
- Login: username@institution.domain
- Password: your password from your institution

Conference network. For everyone:

- Network Name (SSID): conferences
- Network Key (password): universe

Rooms for the Conference/Tutorials/Workshops/Symposium

Monday and Tuesday: See Figure 1 on page 4.

- 1416 – Smalltalk (1st floor/ground floor)
- 2423 – Java (2nd floor)
- 2438 – Logo (2nd floor)
- 2452 – Pascal (2nd floor)
- 2453 – Perl (2nd floor)
- 2458 – Postscript (2nd floor)
- 2465 – Prolog (2nd floor)
- 2269 – Python (2nd floor)

Wednesday to Friday: See Figure 1 on page 4.

- 1423 – Simula (1st floor/ground floor)
- 1416 – Smalltalk (1st floor/ground floor)

Note: The ground floor is called 1st floor in Norway. The 2nd floor is one floor up from the ground floor.

Lunches

Lunches will be at Forskningsparken between 12:30 and 14:00, see Figure 3 in the Map's section (Please bring your ticket).

Coffee Breaks

Coffee breaks will be near the registration desk, and near the rooms Simula and Smalltalk. See Figure 1 on page 4.

Social Events

Reception

The Reception will be in Oslo's Town Hall, June 24 2015 at 18:00 sharp.
Address: Rådhusplassen 1. Near Nationaltheatret metro station. From the conference venue (Forskningsparken station) you can take metro line 4 (Bergkrystallen), or 6 (Ringen). Disembark at Nationaltheatret. See Figure 4 in the Map's section.

Note: Please bring your badge and be on time!

Banquet

The Banquet will be at Ingierstrand Bad, June 25 2015. Buses will leave at 17:40 from the conference venue. Address: Ingierstrandveien 30, 1420 Svartskog. See Figure 5 in the Map's section.

Useful Information about Oslo

Restaurants and Bars

Here follows a list of some recommended places to eat and drink downtown. For more comprehensive lists see web resources like tripadvisor.com or osloby.no (in Norwegian). Norway is a high cost country so you will probably find most prices to be rather high, particularly due to the high tax on alcoholic beverages.

Seafood Restaurants

We recommend to make a reservation in advance.

Solsiden: Fresh and good quality seafood with the possibility to enjoy the sunset over the Oslo Fjord. Open until 22:00. Address: Akershusstranda 13. Telephone: +47 22333630. Email: post@solsiden.no. See www.solsiden.no for more information.

Lofoten: Fresh and good quality seafood restaurant located on the edge of Aker Brygge, a modern area in Oslo (by the seaside). Three course menu for 535,- NOK. Address: Stranden 75. Telephone: +47 22830808. Email: lofoten@fiskerestaurant.no.
See www.lofoten-fiskerestaurant.no/english for more information.

Mares: Fresh and good quality seafood with a nice and delicate atmosphere. Four course menu for 665,- NOK. Address: Skovveien 1. Telephone: +47 22548980.
Email: mares@mares.no. See www.mares.no for more information.

Fjord: Fresh and good quality seafood restaurant located in Oslo city centre. Three course menu for 445,- NOK. Address: Kristian Augusts gate 11. Telefon: +47 22982150. Email: fjord@restauranteik.no. See restaurantfjord.no/en/ for more information.

Mid-Priced Restaurants and Bars

Amundsen Bryggeri & Spiseri: Microbrewery, also with eating options. Address: Stortingsgaten 20. Telephone: +47 24 20 09 00. Price range 150–300 NOK.

Nydalen Bryggeri & Spiseri: Microbrewery, also with eating options. Address: Nydalsveien 30A. Telephone: +47 22239440. Price range 160–350 NOK.

Delicatessen: Tapas bar. Address(Majorstuen): Vibes gate 8A. Address(Aker brygge): Holmens gate 2. Address(Grünerløkka): Søndregate 8. Telephone: +47 22467200. Price range 160–350 NOK.

Fyret: Nice little place with maritime atmosphere. Norwegian/Danish cuisine. Has an extensive range of the traditional Scandinavian Akevitt; recommended! Address: Youngstorget 6. Price range 150–700 NOK.

Villa Paradiso: Italian pizza place. Address(Grünerløkka): Olaf Ryes plass 8. Address(Frogner): Sommerrogata 17. Telephone: +47 22354060.

Hells Kitchen and Illegal Burger: A bar with good pizzas in a hip area. Gets very packed during weekends. Next door is a trendy burger place. Address: Møllergaten 23.

Internasjonalen: Trendy and large bar with good selection of drinks. Address: Youngstorget 2.

Olympen: Originally a "worker's pub" from 1892, the place has now been totally refurbished and serves traditional Norwegian food including an affordable three course menu for 349,- (good value/price). Extensive beer range with many local beers. Address: Grønlandsleiret 15. Telephone: +47 24 10 19 99 Price range 140–250 NOK.

Vognmand Nilsen: Norwegian ingredients in modern style. Affordable three course menu for 335,- (good value/price). Address: Rubina Ranas gate 3. Price range 200–300 NOK.

Summit 21: Bar on the 21st floor of the Radisson Blu Scandinavia Hotel with a great view. Address: Holbergs gate 30.

Vegetarian Restaurants

Vega Fair Food: Akersgata 74. Telephone: +47 47921214. Open until 21:00.

Krishnas Cuisine: Sørkedalsveien 10 B, Majorstuen. Telephone: +47 22692269. Open until 20.00.

Taxi Companies

Taxi2: Tel. 02202

Norgestaxi: Tel. 08000

Oslo Taxi: Tel. 02323

Payment: Pay in cash (Norwegian kroner only) or by credit card. If you want to pay by credit card, please inform the driver at the start of the trip. Taxis in Norway accept the most common cards such as VISA, American Express, Eurocard and MasterCard. Please be aware that non-chipped cards may not always be accepted.

Public Transport and Tickets

Ruter's tickets can be used on the Metro, trams, buses, ferries and trains in Oslo and Akershus, but are not valid on airport shuttle buses, the Airport Express Train, TIMEkspressen or the Bygdøy ferries. Buy your ticket using your mobile phone (See ruter.no/en/mobile-apps/), at kiosks (such 7-Eleven, Deli de Luca, etc.), from ticket machines (outside the metro stations), or at Service Points (Central Station, Majorstuen, etc.). More information at ruter.no.

Tickets and fares (for zone 1):
Single Ticket: 30 (50 if you buy a single ticket on the bus/tram!)
24-hours Ticket: 90
7-days Ticket: 240

Tourist Information Centre

Oslo Visitor Centre: Mon–Sun 09.00–18.00 (all year)
Address: Oslo Visitor Centre is located in Østbanehallen next to Oslo Central Station. Entrance from Jernbanetorget (the square with the tiger sculpture) or from inside the hall.
Call centre: +47 815 30 555 Mon–Fri 09.00–16.00
See more at: http://www.visitoslo.com/en/tourist-information-centre/

Pharmacies

Vitusapotek Blindern:
Problemveien 9, 0313. Telephone: 23195030. Opening hours: 08:30 – 16:30
Vitusapotek Majorstuen:
Kirkeveien 64B. Telephone: 21544470. Opening hours: 08:00 – 19:00
Vitusapotek Jernbanetorget:
Jernbanetorget 4B. Telephone: 23358100. Open 24 hours.

Emergency Numbers

110 – Fire
112 – Police
113 – Ambulance

Program Overview

	Monday - June 22	**Tuesday - June 23**	**Wednesday - June 24**	**Thursday - June 25**	**Friday - June 26**
09:00 – 10:00	Parallel Sessions	Parallel Sessions	Session 1 Keynote: Elvira Albert	Session 5 Keynote: Werner Damm	Session 9 Keynote: Valérie Issarny
10:00 – 10:30	Coffee break	Coffee break	Coffee break	Coffee break	Coffee break
10:30 – 12:30	Parallel Sessions	Parallel Sessions	Session 2A Session 2B	Session 6A Session 6B (Ind. Track)	Session 10
12:30 – 14:00	Lunch	Lunch	Lunch	Lunch	Lunch
14:00 – 15:30	Parallel Sessions	Parallel Sessions	Session 3A Session 3B	Session 7A Session 7B (Ind. Track)	Session 11
15:30 – 16:00	Coffee break	Coffee break	Coffee break	Coffee break	Coffee break
16:00 – 17:30	Parallel Sessions	Parallel Sessions	Session 4 Keynote: Leslie Lamport	Session 8 FME Fellowship Award Ceremony	Session 12
17:30 – ...		FME AGM	Town Hall Reception	Banquet	

Table 1: Time Overview

Monday – June 22	Tuesday – June 23	Wednesday – June 24	Thursday – June 25	Friday - June 26
Workshops - **FMICS** **Time:** 09:00 – 17:00 **Room:** 2438 Logo - **Refinement** **Time:** 10:00 – 17:30 **Room:** 2269 Python - **ESSS** **Time:** 09:00 – 17:00 **Room:** 2423 Java - **F-IDE** **Time:** 09:00 – 17:45 **Room:** 2453 Perl	**Workshops** - **FMICS** **Time:** 09:00 – 17:00 **Room:** 2438 Logo - **Overture** **Time:** 09:00 – 17:45 **Room:** 2453 Perl - **WWV** **Time:** 09:00 – 17:30 **Rooms:** 2423 Java 2438 Logo (keynotes) - **USE/SETS** **Time:** 09:00 – 17:15 **Room:** 2269 Python - **FMSEET** **Time:** 09:00 – 17:30 **Room:** 2458 Postscript	**Main Conference** **Time:** 09:00 – 17:00 **Rooms:** 1423 Simula 1416 Smalltalk	**Main Conference** **Time:** 09:00 – 16:00 **Room:** 1423 Simula	**Main Conference** **Time:** 09:00 – 17:30 **Room:** 1423 Simula
Tutorial - **Modelling and Analysis of Communicating Systems** **Time:** 09:00 – 17:30 **Room:** 2465 Prolog	**Tutorials** - **Abstract Behavioral Specification** **Time:** 14:00 – 17:30 **Room:** 2452 Pascal - **Theory and Practice of Runtime Verification** **Time:** 09:00 – 17:30 **Room:** 1416 Smalltalk	**Social Event** - **Reception** **Time:** 18:00 sharp **Place:** Town Hall	**Industry Track** **Time:** 09:00 – 16:00 **Rooms:** 1423 Simula (keynote) 1416 Smalltalk	
Doctoral Symposium **Time:** 09:00 – 17:30 **Room:** 1416 Smalltalk	**FME AGM** **Time:** 18:00 – ... **Room:** 1416 Smalltalk		**FME Fellowship Award Ceremony** **Time:** 16:00 – 17:30 **Room:** 1423 Simula	
			Social Event - **Banquet** **Time:** Bus at 17:40 (From the venue) **Place:** Ingierstrand Bad	

Table 2: Program Overview

Main Conference

Program for Wednesday, June 24

Program Overview – Track A

Location: Smalltalk, Simula*

09:00 – 10:00	**Session 1 – Keynote** (*Location: Simula*)
	Resource Analysis: From Sequential to Concurrent and Distributed Programs *Elvira Albert*
10:00 – 10:30	**Coffee break**
10:30 – 12:30	**Session 2A – Probabilistic and Hybrid Systems**
10:30 – 11:00	**Direct formal verification of liveness properties in continuous and hybrid dynamical systems** *Andrew Sogokon and Paul Jackson*
11:00 – 11:30	**Counterexamples for Expected Rewards** *Tim Quatmann, Nils Jansen, Christian Dehnert, Ralf Wimmer, Erika Ábráham, Joost-Pieter Katoen and Bernd Becker*
11:30 – 12:00	**Probabilistic Bisimulation for Realistic Schedulers** *Christian Eisentraut, Jens Chr. Godskesen, Holger Hermanns, Lei Song and Lijun Zhang*
12:00 – 12:30	**Abstraction of Elementary Hybrid Systems by Variable Transformation** *Jiang Liu, Naijun Zhan, Hengjun Zhao and Liang Zou*
12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 3A – Temporal Logic**
14:00 – 14:30	**Using Real-Time Maude to Model Check Energy Consumption Behavior** *Shin Nakajima*
14:30 – 15:00	**Trace-Length Independent Runtime Monitoring of Quantitative Policies in LTL** *Xiaoning Du, Yang Liu and Alwen Tiu*
15:00 – 15:30	**Parameter Synthesis through Temporal Logic Specifications** *Tommaso Dreossi, Thao Dang and Carla Piazza*
15:30 – 16:00	**Coffee break**
16:00 – 17:00	**Session 4 – Keynote** (*Location: Simula*)
	What Should Math Have to do with Building Complex Distributed Systems? *Leslie Lamport*

* The two joint keynote sessions are located in Room Simula

Program Overview – Track B

Location: Simula

09:00 – 10:00	**Session 1 – Keynote**
	Resource Analysis: From Sequential to Concurrent and Distributed Programs *Elvira Albert*
10:00 – 10:30	**Coffee break**
10:30 – 12:30	**Session 2B – Security**
10:30 – 11:00	**Certified Reasoning with Infinity** *Asankhaya Sharma, Shengyi Wang, Andreea Costea, Aquinas Hobor and Wei-Ngan Chin*
11:00 – 11:30	**Detection of Design Flaws in the Android Permission Protocol through Bounded Verification** *Hamid Bagheri, Eunsuk Kang, Sam Malek and Daniel Jackson*
11:30 – 12:00	**Privacy by design in practice: reasoning about privacy properties of biometric system architectures** *Julien Bringer, Hervé Chabanne, Daniel Le Métayer and Roch Lescuyer*
12:00 – 12:30	**Verifying Parameterized Timed Security Protocols** *Li Li, Jun Sun, Yang Liu and Jin Song Dong*
12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 3B – Model Checking and Runtime Verification**
14:00 – 14:30	**Verifying the Safety of a Flight-Critical System** *Temesghen Kahsai, Falk Howar, Dimitra Giannakopoulou, Guillaume Brat, Misty Davies and David Bushnell*
14:30 – 15:00	**A Specification Language for Static and Runtime Verification of Data and Control Properties** *Wolfgang Ahrendt, Jesus Mauricio Chimento, Gordon Pace and Gerardo Schneider*
15:00 – 15:30	**Safety, Liveness and Run-time Refinement for Modular Process-Aware Information Systems with Dynamic Sub Processes** *Søren Debois, Thomas Hildebrandt and Tijs Slaats*
15:30 – 16:00	**Coffee break**
16:00 – 17:00	**Session 4 – Keynote**
	What Should Math Have to do with Building Complex Distributed Systems? *Leslie Lamport*

Detailed Program

09:00 – 10:00 Location:	**Session 1: Keynote** *Simula*

09:00 Elvira Albert

Resource Analysis: From Sequential to Concurrent and Distributed Programs

ABSTRACT. Resource analysis aims at automatically inferring upper/lower bounds on the worst/best-case cost of executing programs. Ideally, a resource analyzer should be parametric on the cost model, i.e., the type of cost that the user wants infer (e.g., number of steps, amount of memory allocated, amount of data transmitted, etc.). The inferred upper bounds have important applications in the fields of program optimization, verification and certification. In this talk, we will review the basic techniques used in resource analysis of sequential programs and the new extensions needed to handle concurrent and distributed systems.

10:30 – 12:30 Location:	**Session 2A: Probabilistic and Hybrid Systems** *Smalltalk*

10:30 Andrew Sogokon and Paul Jackson

Direct formal verification of liveness properties in continuous and hybrid dynamical systems

ABSTRACT. This paper addresses the problem of formally verifying the temporal property of eventuality (a type of liveness) in systems given by ordinary differential equations (ODEs) under evolution constraints. This problem is of a more general interest to hybrid system verification, where reasoning about temporal properties in the continuous fragment is often a bottleneck. Much of the difficulty in handling continuous systems stems from the fact that closed-form solutions to non-linear ODEs are rarely available. We present a general method for proving eventuality properties that works with the differential equations directly, without the need to explicitly compute the solutions. Our method is intuitively simple, yet much less conservative than previously reported approaches to solving the eventuality verification problem, making it highly amenable to use as a sound rule of inference in a formal proof calculus for hybrid systems.

11:00 Tim Quatmann, Nils Jansen, Christian Dehnert, Ralf Wimmer, Erika Abraham, Joost-Pieter Katoen and Bernd Becker

Counterexamples for Expected Rewards

ABSTRACT. The computation of counterexamples for probabilistic systems has gained a lot of attention in research during the last few years. However, all of the proposed methods focus on the situation when the probabilities of certain events are too high. In this paper we investigate how counterexamples for properties concerning expected costs (or, equivalently, expected rewards) of events can be computed. We propose methods to extract a minimum subsystem which already leads to costs beyond the allowed bound. Besides these exact methods, we present heuristic approaches based on path search and on best-first search, which are applicable to very large systems when deriving a minimal subsystem becomes infeasible due to the system size. Experiments show that we can compute counterexamples for systems with millions of states.

11:30 Christian Eisentraut, Jens Chr. Godskesen, Holger Hermanns, Lei Song and Lijun Zhang

Probabilistic Bisimulation for Realistic Schedulers

ABSTRACT. Weak distribution bisimilarity is an equivalence notion on probabilistic automata, originally proposed for Markov automata. It has gained some popularity as the coarsest behavioral equivalence enjoying valuable properties like preservation of trace distribution equivalence and compositionality. This holds in the classical context of arbitrary schedulers, but it has been argued that this class of schedulers is unrealistically powerful. This paper studies a strictly coarser notion of bisimilarity, which still enjoys these properties in the context of realistic subclasses of schedulers: Trace distribution equivalence is implied for partial information schedulers, and compositionality is preserved by distributed schedulers. The intersection of the two scheduler classes thus spans a coarser and still reasonable compositional theory of behavioral semantics.

12:00 Jiang Liu, Naijun Zhan, Hengjun Zhao and Liang Zou

Abstraction of Elementary Hybrid Systems by Variable Transformation

ABSTRACT. Elementary hybrid systems (EHSs) are those hybrid systems (HSs) containing elementary functions such as exp, ln, sin, cos, etc. EHSs are very common in practice, especially in safety-critical domains. Due to the non-polynomial expressions which lead to undecidable arithmetic, verification of EHSs is very hard. Existing approaches based on partition of state space or over-approximation of reachable sets suffer from state explosion or inflation of numerical errors. In this paper, we propose a symbolic abstraction approach that reduces EHSs to polynomial hybrid systems (PHSs), by replacing all non-polynomial terms with newly introduced variables. Thus the verification of EHSs is reduced to the one of PHSs, enabling us to apply all the well-established verification techniques and tools for PHSs to EHSs. In this way, it is possible to avoid the limitations of many existing methods. We illustrate the abstraction approach and its application in safety verification of EHSs by several real world examples.

10:30 – 12:30 **Session 2B: Security**
Location: *Simula*

10:30 Asankhaya Sharma, Shengyi Wang, Andreea Costea, Aquinas Hobor and Wei-Ngan Chin

Certified Reasoning with Infinity

ABSTRACT. We demonstrate how infinities improve the expressivity, power, readability, conciseness, and compositionality of a program logic. We prove that adding infinities to Presburger arithmetic enables these improvements without sacrificing decidability. We develop Omega++, a Coq-certified decision procedure for Presburger arithmetic with infinity and benchmark its performance. Both the program and proof of Omega++ are paramaterized over user-selected semantics for the indeterminate terms (such as $0 * \infty$)

11:00 Hamid Bagheri, Eunsuk Kang, Sam Malek and Daniel Jackson

Detection of Design Flaws in the Android Permission Protocol through Bounded Verification

ABSTRACT. The ever increasing expansion of mobile applications into nearly every aspect of modern life, from banking to healthcare systems, is making their security more important than ever. Modern smartphone operating systems (OS) rely substantially on the permission-based security model to enforce restrictions on the operations that each application can perform. In this paper, we perform an analysis of the permission protocol implemented in Android, a popular OS for smartphones.

We propose a formal model of the Android permission protocol, and describe a fully automatic analysis that identifies potential flaws in the protocol. A study of hundreds of real-world Android applications corroborates our finding that the flaws in the Android permission protocol can have severe security implications, in some cases allowing the attacker to bypass the permission checks entirely.

11:30 Julien Bringer, Hervé Chabanne, Daniel Le Métayer and Roch Lescuyer

Privacy by design in practice: reasoning about privacy properties of biometric system architectures

ABSTRACT. The work presented in this paper is the result of a collaboration between academics, industry and lawyers to show the applicability of the privacy by design approach to biometric systems and the benefit of formal methods to this end. The choice of particular techniques and the role of the components (central server, secure module, terminal, smart card, etc.) in the architecture have a strong impact on the privacy guarantees provided by a biometric system. However, existing proposals were made on a case by case basis, which makes it difficult to compare them and to provide a rationale for the choice of specific options. In this paper, we show that a general framework for the definition of privacy architectures can be used to specify these options and to reason about them in a formal way.

12:00 Li Li, Jun Sun, Yang Liu and Jin Song Dong

Verifying Parameterized Timed Security Protocols

ABSTRACT. Quantitative timing is often explicitly used in systems for better security, e.g., the credentials for automatic website logon often has limited lifetime. Verifying timing relevant security protocols in these systems is very challenging as timing adds another dimension of complexity compared with the untimed protocol verification. In our previous work, we proposed an approach to check the correctness of the timed authentication in security protocols with fixed timing constraints. However, a more difficult question persists, i.e., given a particular protocol design, whether the protocol has security flaws in its design or it can be configured secure with proper parameter values? In this work, we answer this question by proposing a parameterized verification framework, where the quantitative parameters in the protocols can be intuitively specified as well as automatically analyzed. Given a security protocol, our verification algorithm either produces the secure constraints of the parameters, or constructs an attack that works for any parameter values. The correctness of our algorithm is formally proved. We implement our method into a tool called PTAuth and evaluate it with several security protocols. Using PTAuth, we have successfully found a timing attack in Kerberos V which is unreported before.

14:00 – 15:30 **Session 3A: Temporal Logic**
Location: *Smalltalk*

14:00 Shin Nakajima

Using Real-Time Maude to Model Check Energy Consumption Behavior

ABSTRACT. Energy consumption is one of the primary non-functional concerns, especially for application programs running on systems that have limited battery capacity. A model-based analysis of the problem is introduced at early stages of development. As rigorous formal models of this, the power consumption automaton and a variant of linear temporal logic are proposed. Detecting unexpected energy consumption is then reduced to a model checking problem, which is unfortunately undecidable in general. This paper introduces some restrictions to the logic formulas representing energy consumption properties so that an automatic analysis is possible with Real-Time Maude.

14:30 Xiaoning Du, Yang Liu and Alwen Tiu

Trace-Length Independent Runtime Monitoring of Quantitative Policies in LTL

ABSTRACT. Linear temporal logic (LTL) has been widely used to specify runtime policies. Traditionally this use of LTL is to capture the qualitative aspects of the monitored systems, but recent developments in metric LTL and its extensions with aggregate operators allow some quantitative policies to be specified. Our interest in LTL-based policy languages is driven by applications in runtime Android malware detection, which requires the monitoring algorithm to be independent of the length of the system event traces so that its performance does not degrade as the traces grow. We propose a policy language based on a past-time variant of LTL, extended with an aggregate operator called the counting quantifier to specify a policy based on the number of times some sub-policies are satisfied in the past. We show that a broad class of policies, but not all policies, specified our language can be monitored in a trace-length independent way, and provide a concrete algorithm to do so. We implement and test our algorithm in an existing Android monitoring framework and show that our approach can effectively specify and enforce quantitative policies drawn from real-world Android malware studies.

15:00 Tommaso Dreossi, Thao Dang and Carla Piazza

Parameter Synthesis through Temporal Logic Specifications

ABSTRACT. Parameters are often used to tune mathematical models and capture nondeterminism and uncertainty in physical and engineering systems. This paper is concerned with parametric nonlinear dynamical systems and the problem of determining the parameter values that are consistent with some expected properties. In our previous works, we proposed a parameter synthesis algorithm limited to safety properties and demonstrated its applications for biological systems. Here we consider more general properties specified by a fragment of STL (Signal Temporal Logic), which allows us to deal with complex behavioral patterns that biological processes exhibit. We propose an algorithm for parameter synthesis w.r.t. a property specified using the considered logic. It exploits reachable set computations and forward refinements. We instantiate our algorithm in the case of polynomial dynamical systems exploiting Bernstein coefficients and we illustrate it on an epidemic model.

14:00 – 15:30 Session 3B: Model Checking and Runtime Verification

Location: *Simula*

14:00 Temesghen Kahsai, Falk Howar, Dimitra Giannakopoulou, Guillaume Brat, Misty Davies and David Bushnell

Verifying the Safety of a Flight-Critical System

ABSTRACT. This paper describes our work on demonstrating verification and certification technologies on a flight critical system. Performed in the context of a major NASA milestone, this study required us to tackle a system of realistic functionality, size, and complexity. Our work targeted a commercial aircraft control system called Transport Control Model (TCM). The work involved several stages: formalizing and disambiguating requirements in collaboration with domain experts; processing models for their use by formal verification tools; applying compositional techniques at the architectural and component level to scale verification. This was one of the most challenging and substantial studies that our group has performed, and it took several person months to complete it. In this paper, we describe the overall process and report the lesson learned.

14:30 Wolfgang Ahrendt, Jesus Mauricio Chimento, Gordon Pace and Gerardo Schneider

A Specification Language for Static and Runtime Verification of Data and Control Properties

ABSTRACT. Static verification techniques can verify properties across all executions of a program, but powerful judgements are hard to achieve automatically. In contrast, runtime verification enjoys full automation, but cannot judge future and alternative runs. In this paper we present a novel approach in which data-centric and control-oriented properties may be stated in a single formalism, amenable to both static and dynamic verification techniques. We develop and formalise a specification notation, ppDATE, extending the control-flow property language used in the runtime verification tool LARVA with pre/post-conditions and show how specifications written in this notation can be analysed both using the deductive theorem prover KeY and the runtime verification tool LARVA. Verification is performed in two steps: KeY first partially proves the data-oriented part of the specification, simplifying the specification which is then passed on to LARVA to check at runtime for the remaining parts of the specification including the control-centric aspects. We apply the approach to Mondex, an electronic purse application.

15:00 Søren Debois, Thomas Hildebrandt and Tijs Slaats

Safety, Liveness and Run-time Refinement for Modular Process-Aware Information Systems with Dynamic Sub Processes

ABSTRACT. We study modularity, run-time adaptation and refinement under safety and liveness constraints in event-based process models with dynamic sub-process instantiation. The study is part of a larger programme to provide semantically well-founded technologies for modelling, implementation and verification of flexible, run-time adaptable process-aware information systems, moved into practice via the Dynamic Condition Response (DCR) Graphs notation co-developed with our industrial partner. Our key contributions are: (1) A formal theory of dynamic sub-process instantiation for declarative, event-based processes under safety and liveness constraints, given as the DCR* process language, equipped with a compositional operational semantics and conservatively extending the DCR Graphs notation; (2) an expressiveness analysis revealing that the DCR* process language is Turing-complete, while the fragment corresponding to DCR Graphs (without dynamic sub-process instantiation) characterises exactly the languages that are the union of a regular and an omega-regular language; (3) a formalisation of run-time refinement and adaptation by composition for DCR* processes and a proof that such refinement is undecidable in general; and finally (4) a decidable and practically useful sub-class of run-time refinements. Our results are illustrated by a running example inspired by a recent Electronic Case Management solution based on DCR Graphs and delivered by our industrial partner. An online prototype implementation of the DCR* language (including examples from the paper) and its visualisation as DCR Graphs can be found at http://tiger.itu.dk:8018/.

16:00 – 17:00 **Session 4: Keynote**
Location: *Simula*

16:00 Leslie Lamport

What Should Math Have to do with Building Complex Distributed Systems?

Program for Thursday, June 25

Program Overview – Track A

Location: Simula

09:00 – 10:00	**Session 5 – Keynote**
	AVACS: Automatic Verification and Analysis of Complex Systems Highlights and Lessons Learned *Werner Damm*
10:00 – 10:30	**Coffee break**
10:30 – 12:30	**Session 6A – Case Studies: Verification in Practice**
10:30 – 11:00	**Semantics-Preserving Simplification of Real-World Firewall Rule Sets** *Cornelius Diekmann, Lars Hupel and Georg Carle*
11:00 – 11:30	**Automated Verification of RPC Stub Code** *Matthew Fernandez, Andronick June, Gerwin Klein and Ihor Kuz*
11:30 – 12:00	**Rigorous Estimation of Floating-Point Round-off Errors with Symbolic Taylor Expansions** *Alexey Solovyev, Charlie Jacobsen, Zvonimir Rakamaric and Ganesh Gopalakrishnan*
12:00 – 12:30	**A Fully Verified Container Library** *Nadia Polikarpova, Julian Tschannen and Carlo A. Furia*
12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 7A – Memory Models**
14:00 – 14:30	**Verifying Opacity of a Transactional Mutex Lock** *John Derrick, Brijesh Dongol, Gerhard Schellhorn, Oleg Travkin and Heike Wehrheim*
14:30 – 15:00	**A framework for correctness criteria on weak memory models** *John Derrick and Graeme Smith*
15:00 – 15:30	**Property-Driven Fence Insertion using Reorder Bounded Model Checking** *Saurabh Joshi and Daniel Kroening*
15:30 – 16:00	**Coffee break**
16:00 – 17:30	**Session 8 – FME Fellowship Award Ceremony**

Program Overview – Track B (Industry Track)

Location: Smalltalk, Simula*

09:00 – 10:00	**Session 5 – Keynote** *(Location: Simula)*
	AVACS: Automatic Verification and Analysis of Complex Systems Highlights and Lessons Learned *Werner Damm*
10:00 – 10:30	**Coffee break**
10:30 – 12:30	**Session 6B – Industry Track**
10:30 – 10:50	**Using Simulink Design Verifier for Automatic Generation of Requirements-based Tests** *Rodrigo Reis, Bruno Miranda and Henrique Masini*
10:50 – 11:10	**Practices for Formal Models as Documents: Evolution of VDM Application to "Mobile FeliCa" IC Chip Firmware** *Taro Kurita, Fuyuki Ishikawa and Keijiro Araki*
11:10 – 11:30	**Formalizing the Concept Phase of Product Development at Philips Healthcare** *Mathijs Schuts and Jozef Hooman*
11:30 – 11:50	**Analyzing the Restart Behavior of Industrial Control Applications** *Stefan Hauck-Stattelmann, Sebastian Biallas, Bastian Schlich, Stefan Kowalewski and Raoul Jetley*
11:50 – 12:10	**Software Development and Authentication for Arms Control Information Barriers** *Neil Evans*
12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 7B – Industry Track**
14:00 – 14:20	**Autofunk: an inference-based formal model generation framework for production systems** *William Durand and Sébastien Salva*
14:20 – 14:40	**Eliminating Static Analysis False Positives using Loop Abstraction and Bounded Model Checking** *Bharti Chimdyalwar, Priyanka Darke, Anooj Chavda, Sagar Vaghani and Avriti Chauhan*
14:40 – 15:00	**Formal Virtual Modelling and Data Verification for Supervision Systems** *Thierry Lecomte*
15:00 – 15:20	**Case Study: Static Security Analysis of the Android Goldfish Kernel** *Ralf Huuck and Tao Liu*
15:30 – 16:00	**Coffee break**
16:00 – 17:30	**Session 8 – FME Fellowship Award Ceremony** *(Location: Simula)*

* The joint keynote session and FME Fellowship Award Ceremony are located in Room Simula

Detailed Program

09:00 – 10:00 **Session 5: Keynote**
Location: *Simula*

09:00 Werner Damm

AVACS: Automatic Verification and Analysis of Complex Systems Highlights and Lessons Learned

10:30 – 12:30 **Session 6A: Case Studies: Verification in Practice**
Location: *Simula*

10:30 Cornelius Diekmann, Lars Hupel and Georg Carle

Semantics-Preserving Simplification of Real-World Firewall Rule Sets

ABSTRACT. The security provided by a firewall for a computer network almost completely depends on the rules it enforces. For over a decade, it has been a well-known and unsolved problem that the quality of many firewall rule sets is insufficient. Therefore, there are many tools to analyze them. However, we found that none of the available tools could handle typical, real-world iptables rulesets. This is due to the complex chain model used by iptables, but also to the vast amount of possible match conditions that occur in real-world firewalls, many of which are not understood by academic and open source tools.

In this paper, we provide algorithms to transform firewall rulesets. We reduce the execution model to a simple list model and use ternary logic to abstract over all unknown match conditions. These transformations enable existing tools to understand real-world firewall rules, which we demonstrate on four decently-sized rulesets. Using the Isabelle theorem prover, we formally show that all our algorithms preserve the firewall's filtering behavior.

11:00 Matthew Fernandez, Andronick June, Gerwin Klein and Ihor Kuz

Automated Verification of RPC Stub Code

ABSTRACT. Formal verification has been successfully applied to provide strong correctness guarantees of software systems, but its application to large code bases remains an open challenge. The technique of component-based software development, traditionally employed for engineering benefit, also aids reasoning about such systems. While there exist compositional verification techniques that leverage the separation implied by a component system architecture, they implicitly rely on the component platform correctly implementing the isolation and composition semantics they assume. Any property proven using these techniques is vulnerable to being invalidated by a bug in the code of the platform itself. In this paper, we show how this assumption can be eliminated by automatically generating machine-checked proofs of the correctness of a component platform's generated Remote Procedure Call (RPC) code. We demonstrate how these generated proofs can be composed with hand-written proofs to yield a system-level property with equivalent assurance to an entirely hand-written proof. This technique forms the basis of a scalable approach to formal verification of large software systems.

11:30 Alexey Solovyev, Charlie Jacobsen, Zvonimir Rakamaric and Ganesh Gopalakrishnan

Rigorous Estimation of Floating-Point Round-off Errors with Symbolic Taylor Expansions

ABSTRACT. Rigorous estimation of maximum floating-point round-off errors is an important capability central to many formal verification tools. Unfortunately, available techniques for this task often provide overestimates. Also, there are no rigorous approaches for transcendental functions. We have developed a new approach called Symbolic Taylor Expansions that avoids this difficulty, and implemented a new tool called FPTaylor embodying this approach. Key to our approach is the use of rigorous global optimization, instead of the more familiar interval arithmetic, affine arithmetic, and/or SMT solvers. In addition to providing far tighter upper bounds of round-off error in a vast majority of cases, FPTaylor also emits analysis certificates in the form of HOL Light proofs. We release FPTaylor along with our benchmarks for evaluation.

12:00 Nadia Polikarpova, Julian Tschannen and Carlo A. Furia

A Fully Verified Container Library

ABSTRACT. The comprehensive functionality and nontrivial design of realistic general-purpose container libraries pose challenges to formal verification that go beyond those of individual benchmark problems mainly targeted by the state of the art. We present our experience verifying the full functional correctness of EiffelBase2: a container library offering all the features customary in modern language frameworks, such as external iterators, and hash tables with generic mutable keys and load balancing. Verification uses the automated deductive verifier AutoProof, which we extended as part of the present work. Our results indicate that verification of a realistic container library (135 public methods, 8,400 LOC) is possible with moderate annotation overhead (1.4 lines of specification per LOC) and good performance (0.2 seconds per method on average).

10:30 – 12:30 **Session 6B: Industry Track**
Location: *Smalltalk*

10:30 Rodrigo Reis, Bruno Miranda and Henrique Masini

Using Simulink Design Verifier for Automatic Generation of Requirements-based Tests

10:50 Taro Kurita, Fuyuki Ishikawa and Keijiro Araki

Practices for Formal Models as Documents: Evolution of VDM Application to "Mobile FeliCa" IC Chip Firmware

11:10 Mathijs Schuts and Jozef Hooman

Formalizing the Concept Phase of Product Development at Philips Healthcare

11:30 Stefan Hauck-Stattelmann, Sebastian Biallas, Bastian Schlich, Stefan Kowalewski and Raoul Jetley

Analyzing the Restart Behavior of Industrial Control Applications

11:50 Neil Evans

Software Development and Authentication for Arms Control Information Barriers

14:00 – 15:30 Location:	**Session 7A: Memory Models** *Simula*

14:00 John Derrick, Brijesh Dongol, Gerhard Schellhorn, Oleg Travkin and Heike Wehrheim

Verifying Opacity of a Transactional Mutex Lock

ABSTRACT. Software transactional memory (STM) provides programmers with a high-level programming abstraction for synchronization of parallel processes, allowing blocks of codes that execute in an interleaved manner to be treated as an atomic block. This atomicity property is captured by a correctness criterion called opacity. Opacity relates histories of a sequential atomic specification with that of STM implementations.
In this paper we prove opacity of a recently proposed STM implementation (a Transactional Mutex Lock) by Dalessandro et al. The proof is done within the interactive verifier KIV and proceeds via the construction of an intermediate level in between sequential specification and implementation, leveraging existing proof techniques for linearizability.

14:30 John Derrick and Graeme Smith

A framework for correctness criteria on weak memory models

ABSTRACT. The implementation of weak (or relaxed) memory models is standard practice in modern multiprocessor hardware. For efficiency, these memory models allow operations to take effect in shared memory in a different order from that which they occur in a program. A number of correctness criteria have been proposed for concurrent objects operating on such memory models, each reflecting different constraints on the objects which can be proved correct. In this paper, we provide a framework in which correctness criteria are defined in terms of two components: the first defining the particular criterion (as it would be defined in the absence of a weak memory model), and the second defining the particular weak memory model. The framework facilitates the definition and comparison of correctness criteria, and encourages reuse of existing definitions. The latter enables properties of the criteria to be proved using existing proofs. We illustrate the framework via the definition of correctness criteria on the TSO (Total Store Order) weak memory model.

15:00 Saurabh Joshi and Daniel Kroening

Property-Driven Fence Insertion using Reorder Bounded Model Checking

ABSTRACT. Modern architectures provide weaker memory consistency guarantees than sequential consistency. These weaker guarantees allow programs to exhibit behaviours where the program statements appear to have executed out of program order. Fortunately, modern architectures provide memory barriers (fences) to enforce the program order between a pair of statements if needed. Due to the intricate semantics of weak memory models, the placement of fences is challenging even for experienced programmers. Too few fences lead to bugs whereas overuse of fences results in performance degradation. This motivates automated placement of fences. Tools that restore sequential consistency in the program may insert more fences than necessary for the program to be correct. Therefore, we propose a property-driven technique that introduces reorder-bounded exploration to identify the smallest number of program locations for fence placement. We implemented our technique on top of CBMC; however, in principle, our technique is generic enough to be used with any model checker. Our experimental results show that our technique is faster and solves more instances of relevant benchmarks as compared to earlier approaches.

14:00 – 15:30 Location:	**Session 7B: Industry Track** *Smalltalk*
14:00	William Durand and Sébastien Salva **Autofunk: an inference-based formal model generation framework for production systems**
14:20	Bharti Chimdyalwar, Priyanka Darke, Anooj Chavda, Sagar Vaghani and Avriti Chauhan **Eliminating Static Analysis False Positives using Loop Abstraction and Bounded Model Checking**
14:40	Thierry Lecomte **Formal Virtual Modelling and Data Verification for Supervision Systems**
15:00	Ralf Huuck and Tao Liu **Case Study: Static Security Analysis of the Android Goldfish Kernel**
16:00 – 17:30 Location:	**Session 8: FME Fellowship Award Ceremony** *Simula*

Program for Friday, June 26

Program Overview

Location: Simula

09:00 – 10:00	**Session 9 – Keynote**
	The key role of formal methods to overcome the interoperability challenge *Valérie Issarny*
10:00 – 10:30	**Coffee break**
10:30 – 12:30	**Session 10 – Model Checking**
10:30 – 11:00	**Certificates for Parameterized Model Checking** *Sylvain Conchon, Alain Mebsout and Fatiha Zaidi*
11:00 – 11:30	**Proving Safety with Trace Automata and Bounded Model Checking** *Daniel Kroening, Matt Lewis and Georg Weissenbacher*
11:30 – 11:45	**QPMC: A Model Checker for Quantum Programs and Protocols** *Yuan Feng, Ernst Moritz Hahn, Andrea Turrini and Lijun Zhang*
11:45 – 12:15	**Automated Circular Assume-Guarantee Reasoning** *Karam Abdelkader, Orna Grumberg, Corina Pasareanu and Sharon Shoham*
12:15 – 12:30	**Model-Based Problem Solving for University Timetable Validation and Improvement** *David Schneider, Michael Leuschel and Tobias Witt*
12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 11 – Static Analysis**
14:00 – 14:30	**Narrowing operators on template abstract domains** *Gianluca Amato, Simone Di Nardo Di Maio, Maria Chiara Meo and Francesca Scozzari*
14:30 – 15:00	**Static Differential Program Analysis for Software-Defined Networks** *Tim Nelson, Andrew D. Ferguson and Shriram Krishnamurthi*
15:00 – 15:30	**Static Optimal Scheduling for Synchronous Data Flow Graphs with Model Checking** *Xue-Yang Zhu, Rongjie Yan, Yu-Lei Gu, Jian Zhang, Wenhui Zhang and Guangquan Zhang*
15:30 – 16:00	**Coffee break**
16:00 – 17:30	**Session 12 – Logics and Semantics**
16:00 – 16:30	**The Semantics of Cardinality-based Feature Models via Formal Languages** *Aliakbar Safilian, Tom Maibaum and Zinovy Diskin*
16:30 – 17:00	**Towards Formal Verification of Orchestration Computations Using the K Framework** *Musab A. Alturki and Omar Alzuhaibi*
17:00 – 17:30	**Typed First-Order Logic** *Peter Schmitt and Mattias Ulbrich*

Detailed Program

09:00 – 10:00 **Session 9: Keynote**
Location: *Simula*

09:00 Valérie Issarny

The key role of formal methods to overcome the interoperability challenge

ABSTRACT. Given the highly dynamic and extremely heterogeneous software systems composing the Future Internet, automatically achieving interoperability between software components without modifying them is more than simply desirable, it is quickly becoming a necessity. Although much work has been carried out on interoperability, existing solutions have not fully succeeded in keeping pace with the increasing complexity and heterogeneity of modern software, and meeting the demands of runtime support. On the one hand, solutions at the application layer target higher automation and loose coupling through the synthesis of intermediary entities, mediators, to compensate for the differences between the interfaces of components and coordinate their behaviours, while assuming the use of the same middleware solution. On the other hand, solutions to interoperability across heterogeneous middleware technologies do not reconcile the differences between components at the application layer. In order to allow for interoperability across layers, the "emergent middleware" paradigm was introduced. Emergent middleware leverages formal methods so as to be able to rigorously abstract and concretize protocols, and further reason about protocol mismatches so as to synthesize the necessary mediators that reconcile the differences between software components from the application down to the middleware layers. In this talk, I will review the foundations of emergent middleware and will then concentrate on its application to the development of distributing software systems contributing to the realization of the smart city vision, which involves the composition of highly heterogeneous systems.

10:30 – 12:30 **Session 10: Model Checking**
Location: *Simula*

10:30 Sylvain Conchon, Alain Mebsout and Fatiha Zaidi

Certificates for Parameterized Model Checking

ABSTRACT. This paper presents a technique for the certification of Cubicle, a model checker for proving safety properties of parameterized systems. To increase the confidence in its results, Cubicle now produces a proof object (or certificate) that, if proven valid, guarantees that the answer for this specific input is correct. The main challenges addressed in this paper are (1) the production of such certificates without degrading the performances of the model checker and (2) the construction of these proof objects so that they can be independently and efficiently verified by an SMT solver. Since the burden of correctness insurance now relies on this external solver, a stronger guarantee is obtained by the use of multiple backend automatic provers for redundancy. Experiments show that our approach does not impact Cubicle's performances and that we were able to verify certificates for industrial size verification problems. As a byproduct these certificates allowed us to find subtle and critical implementation bugs in Cubicle.

11:00 Daniel Kroening, Matt Lewis and Georg Weissenbacher

Proving Safety with Trace Automata and Bounded Model Checking

ABSTRACT. Loop under-approximation is a technique that enriches C programs with additional branches that represent the effect of a (limited) range of loop iterations. While this technique can speed up the detection of bugs significantly, it

introduces redundant execution traces which may complicate the verification of the program. This holds particularly true for verification tools based on Bounded Model Checking, which incorporate simplistic heuristics to determine whether all feasible iterations of a loop have been considered.
We present a technique that uses *trace automata* to eliminate redundant executions after performing loop acceleration. The method reduces the diameter of the program under analysis, which is in certain cases sufficient to allow a safety proof using Bounded Model Checking. Our transformation is precise—it does not introduce false positives, nor does it mask any errors. We have implemented the analysis as a source-to-source transformation, and present experimental results showing the applicability of the technique.

11:30 Yuan Feng, Ernst Moritz Hahn, Andrea Turrini and Lijun Zhang

QPMC: A Model Checker for Quantum Programs and Protocols

ABSTRACT. We present QPMC (Quantum Program/Protocol Model Checker), an extension of the probabilistic model checker ISCASMC to automatically verify quantum programs and quantum protocols. QPMC distinguishes itself from the previous quantum model checkers proposed in the literature in that it works for general quantum programs and protocols, not only those using Clifford operations.

11:45 Karam Abdelkader, Orna Grumberg, Corina Pasareanu and Sharon Shoham

Automated Circular Assume-Guarantee Reasoning

ABSTRACT. Compositional verification techniques aim to decompose the verification of a large system into the more manageable verification of its components. In recent years, compositional techniques have gained significant successes following a breakthrough in the ability to automate assume-guarantee reasoning. However, automation is still restricted to simple acyclic assume-guarantee rules.
In this work, we focus on automating circular assume-guarantee reasoning in which the verification of individual components mutually depends on each other. We use a sound and complete circular assume-guarantee rule and we describe how to automatically build the assumptions needed for using the rule. Our algorithm accumulates joint constraints on the assumptions based on (spurious) counterexamples obtained from checking the premises of the rule, and uses a SAT solver to synthesize minimal assumptions that satisfy these constraints. To the best of our knowledge, our work is the first to automate circular assume-guarantee reasoning. We implemented our approach and compared it with an established learning-based method that uses an acyclic rule. In all cases, the assumptions generated for the circular rule were significantly smaller, leading to smaller verification problems. Further, on larger examples, we obtained a significant speedup as well.

12:15 David Schneider, Michael Leuschel and Tobias Witt

Model-Based Problem Solving for University Timetable Validation and Improvement

ABSTRACT. Constraint satisfaction problems can be expressed very elegantly in state-based formal methods such as B. However, can such specifications be directly used for solving real-life problems? We will try and answer this question in the present paper with regard to the university timetabling problem. We report on an ongoing project to build a formal model-based curriculum timetable validation tool where we use a formal specification as the basis to validate timetables from a student's perspective and to support incremental modification of timetables. In this article we focus on expressing the problem domain, the formalization in B and our approach to execute the formal model in a production system using ProB.

14:00 – 15:30	**Session 11: Static Analysis**
Location:	*Simula*

14:00 Gianluca Amato, Simone Di Nardo Di Maio, Maria Chiara Meo and Francesca Scozzari

Narrowing operators on template abstract domains

ABSTRACT. In the theory of abstract interpretation, narrowing operators are used to improve the precision of the analysis after a post-fixpoint has been reached. This is especially true on numerical domains, since they are generally endowed with infinite descending chains which may lead to a non-terminating analysis in the absence of narrowing. We provide an abstract semantics which improves the analysis precision and show that, for a large class of numerical abstract domains over integer variables (such as intervals, octagons and template polyhedra), it is possible to avoid infinite descending chains and omit narrowing. Moreover, we propose a new family of narrowing operators for real variables which improves the analysis precision.

14:30 Tim Nelson, Andrew D. Ferguson and Shriram Krishnamurthi

Static Differential Program Analysis for Software-Defined Networks

ABSTRACT. Networks are increasingly controlled by software, and bad updates can bring down an entire network. Network operators therefore need tools to determine the impact of changes. To address this, we present static differential analysis of software-defined network (SDN) controller programs. Given two versions of a controller program our tool, Chimp, builds atop Alloy to produce a set of concrete scenarios where the programs differ in their behavior. Chimp thus enables network developers to exploit the power of formal methods tools without having to be trained in formal logic or property elicitation. Furthermore, we show that there are many interesting properties that one can state about the changes themselves. Our evaluation shows that Chimp is fast, returning scenarios in under a second on several real applications.

15:00 Xue-Yang Zhu, Rongjie Yan, Yu-Lei Gu, Jian Zhang, Wenhui Zhang and Guangquan Zhang

Static Optimal Scheduling for Synchronous Data Flow Graphs with Model Checking

ABSTRACT. Synchronous data flow graphs (SDFGs) are widely used to model digital signal processing and streaming media applications. In this paper, we present exact methods for static optimal scheduling and mapping of SDFGs on a heterogenous multiprocessor platform. The optimization criteria we consider are throughput and energy consumption, taking into account the combination of various constraints such as auto-concurrency and buffer sizes. We present a concise and flexible (priced) timed automata semantics of system models, which include an SDFG and a multiprocessor platform, and formulate the optimization goals as temporal logic formulas. The optimization and scheduling problems are then transformed to model checking problems, which are solved by UPPAAL (CORA). Thanks to the exhaustive exploration nature of model checking and the facility of the tools, we obtain throughput-optimal schedules that have a best energy consumption, and energy consumption-optimal schedules that have a best throughput. The approach is applied to two real applications, which shows that our approach can deal with moderate models within reasonable execution time and reveal the impacts of different constraints on optimization goals.

16:00 – 17:30 **Session 12: Logics and Semantics**
Location: *Simula*

16:00 Aliakbar Safilian, Tom Maibaum and Zinovy Diskin

The Semantics of Cardinality-based Feature Models via Formal Languages

ABSTRACT. A feature model is a graphical structure presenting a hierarchical decomposition of features, called a feature diagram, with some possible crosscutting constraints between them. Cardinality-based feature models provide the most expressive language among the existing feature modeling languages. In this paper, we provide a reduction process, which allows us to go from a cardinality-based feature diagram to an appropriate regular expression. As for crosscutting constraints, we propose a formal language interpretation of them. In this way, we provide a formal language-based semantics for cardinality-based feature models. Accordingly, we describe a computational hierarchy of feature models, which guides us in how feature models can be constructively analyzed. We also characterize some existing analysis operations over feature models in terms of on languages and discuss the corresponding decidability problems.

16:30 Musab A. Alturki and Omar Alzuhaibi

Towards Formal Verification of Orchestration Computations Using the K Framework

ABSTRACT. Orchestration provides a general model of concurrent computations, although it is more often referred to in the context of service orchestrations describing the composition and management of (web) services. A minimal yet expressive theory of orchestration is provided by Orc, in which computations are modeled by site calls and their orchestrations through a few combinators. Using Orc, formal verification of correctness of orchestrations amounts to devising an executable formal semantics of Orc and leveraging existing tool support. Despite its simplicity and elegance, giving formal semantics to Orc capturing precisely its intended behaviors is far from trivial and has been of interest since Orc's inception primarily due to the challenges posed by concurrency, timing and the distinction between internal and external actions. This paper presents a semantics-based approach for formally verifying Orc orchestrations using the K framework. Unlike previously developed operational semantics of Orc, the K semantics is not directly based on the interleaving semantics given by the reference SOS specification of Orc. Instead, the K semantics takes full advantage of true concurrency enabled by K. It also utilizes various K facilities to arrive at a clean, minimal and elegant semantic specification. To demonstrate the usefulness and applicability of the proposed approach, we also describe a specification for a robotics case study and provide initial formal verification results.

17:00 Peter Schmitt and Mattias Ulbrich

Typed First-Order Logic

ABSTRACT. This paper contributes to the theory of typed first-order logic. We present a sound and complete axiomatization lifting restriction imposed by previous results. As a second contribution this paper provides complete axiomatizations for the type predicates instance_T, exactInstance_T, and functions cast_T indispensable for reasoning about object-oriented programming languages.

Workshops and Doctoral Symposium

Monday – June 22	**Tuesday – June 23**
FMICS Time: 09:00 – 17:00 Room: 2438 Logo	
Refinement Time: 10:00 – 17:30 Room: 2269 Python	**Overture** Time: 09:00 – 17:45 Room: 2453 Perl
ESSS Time: 09:00 – 17:00 Room: 2423 Java	**WWV** Time: 09:00 – 17:30 Room: 2423 Java 2438 Logo (keynotes)
F-IDE Time: 09:00 – 17:45 Room: 2453 Perl	**USE** Time: 09:00 – 12:30 Room: 2269 Python
Doctoral Symposium Time: 09:00 – 17:30 Room: 1416 Smalltalk	**SETS** Time: 14:00 – 17:15 Room: 2269 Python
	FMSEET Time: 09:00 – 17:30 Room: 2458 Postscript

Table 6: Overview of Workshops and Doctoral Symposium

Keynotes of Workshops and Doctoral Symposium

Monday – June 22	Tuesday – June 23
Formal Verification of Industrial Critical Software *Marielle Petit-Doche (Systerel)* **Event: FMICS** **Time:** 09:00 – 10:00	**Formal Patterns for Web and Cloud Computing** *José Meseguer (University of Illinois at Urbana-Champaign, USA)* **Event: FMICS / WWV** **Time:** 09:00 – 10:00
From Timed Automata to Stochastic Hybrid Games *Kim G. Larsen (Aalborg University)* **Event: FMICS** **Time:** 14:00 – 15:00	**Moving fast with software verification** *Dino Distefano (Queen Mary University, London, UK & Facebook)* **Event: FMICS / WWV** **Time:** 14:00 – 15:00
Verification of Concurrent Software *Marieke Huisman (University of Twente)* **Event: ESSS** **Time:** 09:00 – 10:00	**Title: TBA** *Taro Kurita (Sony Felica)* **Event: Overture** **Time:** 09:15 – 10:00
Modelling Reliability with Degrees of Uncertainty based on Subjective Logic *Audun Jøsang (University of Oslo)* **Event: ESSS** **Time:** 14:00 – 15:00	**Symbolic Execution and Advanced Test Coverage Criteria** *Nikolaï Kosmatov* **Event: USE** **Time:** 09:15 – 10:00
Title: TBA *John Fitzgerald (Newcastle University, UK)* **Event: F-IDE** **Time:** 09:10 – 10:00	**Why, how and what should be taught about Formal Methods?** *Maximiliano Cristia (CIFASIS, Universidad Nacional de Rosario, Argentinia)* **Event: FMSEET** **Time:** 09:05 – 10:00
Proving that Android's, Java's and Python's sorting algorithm is broken (and showing how to fix it) *Stijn de Gouw (CWI and SDL, the Netherlands)* **Event: Doctoral Symposium** **Time:** 09:00 – 10:00	

Doctoral Symposium

Room: 1416 Smalltalk **Monday – June 22**

09:00 – 10:00	**Opening** **Session 1 – Keynote**
	Proving that Android's, Java's and Python's sorting algorithm is broken (and showing how to fix it) *Stijn de Gouw (CWI and SDL, the Netherlands)*
10:00 – 10:30	**Coffee break**
10:30 – 12:30	**Session 2**
10:30 – 11:00	**Properties of Communicating Controllers for Safe Traffic Manoeuvres** *Maike Schwammberger (University of Oldenburg, Germany)*
11:00 – 11:30	**Real-time systems modelling with UML state machines and coloured Petri nets** *Mohamed Mahdi Benmoussa (Université Paris 13, France)*
11:30 – 12:00	**Test-Case Generation via Language Inclusion for Non-Deterministic Networks of Timed Automata** *Florian Lorber (Graz University of Technology, Austria)*
12:00 – 12:30	**Trace-length Independent Runtime Monitoring** *Xiaoning Du (Nanyang Technological University, Singapore)*
12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 3**
14:00 – 14:30	**Inheritance and refinement of trustworthy component-based systems** *José Dihego (Universidade Federal de Pernambuco, Brazil)*
14:30 – 15:00	**Component-based CPS Verification: A Recipe for Reusability** *Andreas Müller (Johannes Kepler University Linz, Austria)*
15:00 – 15:30	**A Novel and Faithful Semantics for Feature Modeling** *Aliakbar Safilian (McMaster University, Canada)*
15:30 – 16:00	**Coffee break**
16:00 – 17:30	**Session 4**
16:00 – 16:30	**A Formal Model for the Safety-Critical Java Level 2 Paradigm** *Matthew Luckcuck (University of York, United Kingdom)*
16:30 – 17:00	**A Code Generator for VDM-RT models** *Miran Hasanagic (Aarhus University, Denmark)*
17:00 – 17:30	**Privacy-Preserving Social Networks** *Raúl Pardo (Chalmers University of Technology, Sweden)*

FMICS – Formal Methods for Industrial Critical Systems

Room: **2438 Logo**	**Monday – June 22**	**Tuesday – June 23**
Sessions:	**Invited Talks**	
09:00 – 10:00	**Formal Verification of Industrial Critical Software** *Marielle Petit-Doche (Systerel)*	**Formal Patterns for Web and Cloud Computing (shared with WWV 2015)** *José Meseguer (University of Illinois at Urbana-Champaign, USA)*
10:00 – 10:30	**Coffee break**	
Sessions:	**Applications**	**Specification and Analysis**
10:30 – 11:00	**A Case Study on Formal Verification of the Anaxagoros Hypervisor Paging System with Frama-C** *Allan Blanchard, Nikolai Kosmatov, Matthieu Lemerre and Frédéric Loulergue*	**Require, Test and Trace IT** *Bernhard K. Aichernig, Klaus Hörmaier, Florian Lorber, Dejan Nickovic and Stefan Tiran*
11:00 – 11:30	**Intra-Procedural Optimization of the Numerical Accuracy of Programs** *Nasrine Damouche, Matthieu Martel and Alexandre Chapoutot*	**Applying Finite State Process Algebra to Formally Specify a Computational Model of Security Requirements in the Key2phone-Mobile Access Solution** *Sunil Chaudhary, Linfeng Li, Eleni Berki, Marko Helenius, Juha Kela and Markku Turunen*
11:30 – 12:00	**Successful Use of Incremental BMC in the Automotive Industry** *Peter Schrammel, Daniel Kroening, Martin Brain, Ruben Martins, Tino Teige and Tom Bienmüller*	**Timed Mobility and Timed Communication for Critical Systems** *Bogdan Aman and Gabriel Ciobanu*
12:00 – 12:30	**Formal Analysis and Testing of Real-time Automotive Systems using UPPAAL Tools** *Jin Hyun Kim, Kim G. Larsen, Petur Olsen, Brian Nielsen and Marius Mikucionis*	**On the Formal Analysis of Photonic Signal Processing Systems** *Umair Siddique, Sidi Mohamed Beillahi and Sofiene Tahar*
12:30 – 14:00	**Lunch**	
Sessions:	**Invited Talks**	
14:00 – 15:00	**From Timed Automata to Stochastic Hybrid Games** *Kim G. Larsen (Aalborg University)*	**Moving fast with software verification (shared with WWV 2015)** *Dino Distefano (Queen Mary University, London, UK & Facebook)*
15:00 – 15:30	Time for Discussions etc.	
15:30 – 16:00	**Coffee break**	
Sessions:	**Protocols**	**Verification**
16:00 – 16:30	**Colored Petri Net Modeling of the Publish/Subscribe Paradigm in the Context of Web Services Resources** *Valentin Valero, Hermenegilda Macia, Gregorio Díaz and M. Emilia Cambronero*	**Automated Verification of Nested DFS** *Jaco van de Pol*
16:30 – 17:00	**Modeling and Verification for the Server-Side Netpay Protocol** *Kaylash Chaudhary and Ansgar Fehnker*	**On the Formal Verification of Optical Quantum Gates in HOL** *Mohamed Yousri Mahmoud, Prakash Panangaden and Sofiene Tahar*

Refinement

Room: 2269 Python

Monday – June 22

10:00 – 10:30	**Coffee break**
10:30 – 12:30	**Session 1**
10:30 – 11:00	**SCJ-Circus: a refinement-oriented formal notation for Safety-Critical Java** *Alvaro Miyazawa and Ana Cavalcanti*
11:00 – 11:30	**Denotational Semantics of Channel Mobility** *Gerard Ekembe Ngondi and Jim Woodcock*
11:30 – 12:00	**A Theory of Service Dependency** *Luigia Petre and Mats Neovius*
12:00 – 12:30	**Formal refinement of extended state machines** *Thomas Fayolle, Marc Frappier, Frederic Gervais and Regine Laleau*
12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 2**
14:00 – 14:30	**Program Derivation by Correctness Enhancements** *Nafi Diallo, Wided Ghardallou, Jules Desharnais and Ali Mili*
14:30 – 15:00	**Reversible Computing and Refinement** *Frank Zeyda, Steve Dunne and Bill Stoddart*
15:00 – 15:30	**Linking linearizability and contextual trace refinement** *Lindsay Groves and Brijesh Dongol*
15:30 – 16:00	**Coffee break**
16:00 – 17:30	**Session 3**
16:00 – 16:30	**A logic for n-dimensional hierarchical refinement** *Alexandre Madeira, Manuel A. Martins and Luis Barbosa*
16:30 – 17:00	**Programming language features for refinement** *Rustan Leino and Jason Koenig*
17:00 – 17:30	**Big Data refinement** *Eerke Boiten*

ESSS – Engineering Safety and Security Systems

Room: 2423 Java

Monday – June 22

09:00 – 10:00	**Session 1 – Invited Talk**
	Verification of Concurrent Software *Marieke Huisman (University of Twente)*
10:00 – 10:30	**Coffee break**
10:30 – 12:00	**Session 2**
10:30 – 11:00	**Verification of Railway Interlocking Systems** *Simon Busard, Quentin Cappart, Christophe Limbree, Charles Pecheur and Pierre Schaus*
11:00 – 11:30	**Formal Verification of Real-Time Function Blocks Using PVS** *Linna Pang, Chen-Wei Wang, Mark Lawford, Alan Wassyng, Josh Newell, Vera Chow and David Tremaine*
11:30 – 12:00	**Automatic Generation of Minimal Cut Sets** *Sentot Kromodimoeljo and Peter Lindsay*
12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 3 – Invited Talk**
14:00 – 15:00	**Modelling Reliability with Degrees of Uncertainty based on Subjective Logic** *Audun Jøsang (University of Oslo)*
15:00 – 15:30	**Indefinite Waitings in MIRELA Systems** *Johan Arcile, Raymond Devillers, Jean-Yves Didier, Hanna Klaudel and Artur Rataj*
15:30 – 16:00	**Coffee break**
16:00 – 17:00	**Session 4**
16:00 – 16:30	**Breaking Dense Structures-Proving Stability of Densely Structured Hybrid Systems** *Eike Moehlmann and Oliver Theel*
16:30 – 17:00	**Using Indexed and Synchronous Events to Model and Validate Cyber-Physical Systems** *Chen-Wei Wang, Jonathan Ostroff and Simon Hudon*

F-IDE – Formal Integrated Development Environment

Room: 2453 Perl

Monday – June 22

09:00 – 09:10	**Opening**
09:10 – 10:00	**Session 1 – Invited Talk** *John Fitzgerald (Newcastle University, UK)*
10:00 – 10:30	**Coffee break**
	Session 2
10:30 – 12:30	**The AutoProof Verifier: Usability by Non-Experts and on Standard Code** *Carlo A. Furia, Christopher M. Poskitt and Julian Tschannen*
	Towards Enabling Overture as a Platform for Formal Notation IDEs *Luís Diogo Couto, Peter Gorm Larsen, Miran Hasanagic, Georgios Kanakis, Kenneth Lausdahl and Peter W. V. Tran-Jørgensen*
	Software Architecture of Code Analysis Frameworks Matters: The Frama-C Example *Julien Signoles*
	An experimental Study using ACSL and Frama-C to formulate and verify Low-Level Requirements from a DO-178C compliant Avionics Project *Frank Dordowsky*
12:30 – 14:00	**Lunch**
	Session 3
14:00 – 15:30	**Building an IDE for the Calculational Derivation of Imperative Programs** *Dipak L. Chaudhari and Om Damani*
	Formal Reasoning Using an Iterative Approach with an Integrated Web IDE *Nabil M. Kabbani, Daniel Welch, Caleb Priester, Stephen Schaub, Blair Durkee, Yu-Shan Sun, and Murali Sitaraman*
	A Holistic Approach to Embedded Systems Development information *Bojan Nokovic and Emil Sekerinski*
15:30 – 16:00	**Coffee break**
16:00 – 17:30	**Session 4** **Demos and Discussion**
17:30 – 17:45	**Closing**

Overture – Overture/VDM

Room: 2453 Perl

Tuesday – June 23

09:00 – 09:15	**Opening:** *Peter Gorm Larsen and Fuyuki Ishikawa*
09:15 – 10:00	**Session 1 – Invited Talk** *Taro Kurita (Sony Felica)*
10:00 – 10:30	**Coffee break**
	Session 2
10:30 – 12:30	**JODTool on the Overture Tool to manage formal requirement dictionaries** *Yoichi Omori, Keijiro Araki and Peter Gorm Larsen*
	VDM Animation for a Wider Range of Stakeholders *Tomohiro Oda, Yasuhiro Yamamoto, Kumiyo Nakakoji, Keijiro Araki and Peter Gorm Larsen*
	Integrating the PVSio-web modelling and prototyping environment with Overture *Paolo Masci, Luis Diogo Couto, Peter Gorm Larsen and Paul Curzon*
	Extending the Overture code generator towards Isabelle syntax *Luís Diogo Couto and Peter Würtz Vinter Tran-Jørgensen*
12:30 – 14:00	**Lunch**
	Session 3
14:00 – 15:30	**Code Generation of VDM++ Concurrency** *Georgios Kanakis, Peter Gorm Larsen and Peter Würtz Vinter Tran-Jørgensen*
	Generating Java RMI code for the distributed aspects of VDM-RT models *Miran Hasanagic, Peter Gorm Larsen and Peter Würtz Vinter Tran-Jørgensen*
	Improving Time Estimates in VDM-RT Models *Morten Larsen, Peter Würtz Vinter Tran-Jørgensen and Peter Gorm Larsen*
15:30 – 16:00	**Coffee break**
	Session 4
16:00 – 17:30	**Case Studies on Combination of VDM and Test-Driven Approaches: Application, Model Finding and Refinement** *Fuyuki Ishikawa*
	Pacemaker Parameter Tuning using Crescendo *Carl Gamble, Martin Mansfield, John Fitzgerald and Peter Gorm Larsen*
	TASTE for Overture to keep SLIM *Marcel Verhoef and Maxime Perrotin*
17:30 – 17:45	**Closing**

WWV – Automated Specification and Verification of Web Systems

Room: 2423 Java, 2438 Logo*

Tuesday – June 23

09:00 – 10:00	**Session 1 – Invited Talk** (*Room Logo*)
	Formal Patterns for Web and Cloud Computing (shared with FMICS 2015) *José Meseguer (University of Illinois at Urbana-Champaign, USA)*
10:00 – 10:30	**Coffee break**
10:30 – 12:00	**Session 2 – Languages for Safety and Security** (*Room Java*)
10:30 – 11:00	**A Calculus of Mobility and Communication for Ubiquitous Computing** *Nosheen Gul*
11:00 – 11:30	**Unlocking Blocked Communicating Processes** *Adrian Francalanza, Marco Giunti, and António Ravara*
11:30 – 12:00	**On Properties of Policy-Based Specifications** *Andrea Margheri, Rosario Pugliese, and Francesco Tiezzi*
12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 3 – Invited Talk** (*Room Logo*)
14:00 – 15:00	**Moving fast with software verification (shared with FMICS 2015)** *Dino Distefano (Queen Mary University, London, UK & Facebook)*
15:00 – 15:30	Time for Discussions etc.
15:30 – 16:00	**Coffee break**
16:00 – 17:30	**Session 4 – Web Analysis** (*Room Java*)
16:00 – 16:30	**Personalised Web Search: one step beyond** *Rocco De Nicola, Francesco Tiezzi, Marinella Petrocchi, Angelo Spognardi, and Van Tien Hoang*
16:30 – 17:00	**Semantics-based Automated Web Testing** *Hai-Feng Guo, Qing Ouyang, and Harvey Siy*
17:00 – 17:30	**Using a Machine Learning Approach to Evaluate Product Line Features** *Davide Bacciu, Stefania Gnesi, and Laura Semini*

* The two joint keynote sessions are located in Room Logo

USE – Usages of Symbolic Execution

Room: 2269 Python

Tuesday – June 23

09:00 – 09:15	**Opening**
09:15 – 10:00	**Session 1 – Invited Talk**
	Symbolic Execution and Advanced Test Coverage Criteria *Nikolaï Kosmatov*
10:00 – 10:30	**Coffee break**
10:30 – 12:30	**Session 2**
10:30 – 11:00	**Test Data Generation for Cyclic Executives with CBMC and Frama-C : A Case Study** *Omer Landry Nguena Timo and Guillaume Langelier*
11:00 – 11:30	**Symbolic Input-Output Conformance Checking for Model-Based Mutation Testing** *Bernhard K. Aichernig and Martin Tappler*
11:30 – 12:00	**An Illustrative Use Case of the DIVERSITY Platform based on UML Interaction Scenarios** *Mathilde Arnaud, Boutheina Bannour and Arnault Lapitre*
12:00 – 12:30	**Bound Analysis for Whiley Programs** *Min-Hsien Weng, Mark Utting and Bernhard Pfahringer*
12:30 – 14:00	**Lunch**

SETS – Sets and Tools

Room: 2269 Python

Tuesday – June 23

12:30 – 14:00	**Lunch**
14:00 – 15:30	**Session 1**
14:00 – 14:30	**Encoding TLA+ set theory into many-sorted first-order logic** *S. Merz, H. Vanzetto*
14:30 – 15:00	**Is hyper-extensionality preservable under deletions of graph elements?** *A. Casagrande, C. Piazza, A. Policriti*
15:00 – 15:30	**Adding Partial Functions to Constraint Logic Programming with Sets** *M. Cristià, G. Rossi*
15:30 – 16:00	**Coffee break**
16:00 – 17:15	**Session 2**
16:00 – 16:30	**Specification and Validation of the MODAM Module Manager** *M. Utting, F. Boulaire*
16:30 – 17:00	**First Steps in Integrating log into Z/EVES** *M. Cristià, G. Rossi, C. Frydman*
17:00 – 17:15	**Closing session**

FMSEET – Formal Methods in Software Engineering Education and Training

Room: 2458 Postscript

Tuesday – June 23

09:00 – 09:05	**Welcome**
09:05 – 10:00	**Session 1 – Invited Talk** Chair: Andreas Bollin
	Why, how and what should be taught about Formal Methods? *Maximiliano Cristia (CIFASIS, Universidad Nacional de Rosario, Argentinia)*
10:00 – 10:30	**Coffee break**
	Session 2 Chair: Andreas Bollin
10:30 – 12:30	**Helping Programmers to Adopt Set-Based Specifications** *Maximiliano Cristia, Gianfranco Rossi and Claudia Frydman*
	The Role of Modelling in Teaching Formal Methods for Software Engineering *Tony Cowling*
	Foundations of Semantics and Model Checking in a Software Engineering Course *Henning Bordihn, Anna-Lena Lamprecht and Tiziana Margaria*
	Making Formal Methods Popular: The Crux is Math Education! *Franz Lichtenberger*
12:30 – 14:00	**Lunch**
	Session 3 Chair: Tiziana Margaria
14:00 – 15:30	**Keys and Roles of Formal Methods Education for Industry: 10 Year Experience with Top SE Program** *Fuyuki Ishikawa, Nobukazu Yoshioka and Yoshinori Tanabe*
	Considerations in Event-B Refinement toward Industrial Use *Naoto Sato and Fuyuki Ishikawa*
	Well-defined Software Process as Vehicle to Understand Effectiveness of Formal Methods *Shigeru Kusakabe, Yoich Omori and Keijiro Araki*
	Evaluation Using a Formal Reasoning Concept Inventory *Joseph Hollingsworth and Murali Sitaraman*
15:30 – 16:00	**Coffee break**
16:00 – 17:30	**Session 4** **Discussion Round**

Tutorials

	Monday – June 22	Tuesday – June 23	
09:00 – 12:30	**Modelling and Analysis of Communicating Systems** Room: 2465 Prolog		**Theory and Practice of Runtime Verification** Room: 1416 Smalltalk
14:00 – 17:30		**Abstract Behavioral Specifications** Room: 2452 Pascal	

Table 7: Overview of Tutorials

Modelling and Analysis of Communicating Systems

Monday – June 22

Time: 09:00 – 17:30
Room: 2465 Prolog

Tutors:
Jan Friso Groote (Eindhoven University of Technology)
Mohammad Mousavi (Halmstad University, Sweden)
Tim Willemse (Eindhoven University of Technology)

Brief description: This full day tutorial addresses the issue of formal modeling and verification of distributed and concurrent systems based on the recent book Modeling and Analysis of Communicating Systems that appeared in the summer of 2014. The theory in the book is developed with as major guideline how to model and verify real life behaviour. It rests on two pillars, namely process algebras for behavioural description and modal logic to characterise requirements. As data is indispensable when describing realistic systems, both formalisms are endowed with a whole range of data types, including functions, sets and reals. Timed behaviour can also be expressed in both the algebra and the logic.

An important purpose of the tutorial is to give an overview of the techniques addressed in the book to mathematically verify the correctness of communicating systems. There are two major approaches. The first one is by proving that an implementation behaves the same (in the sense of e.g. branching bisimulation) to the specification. For these normal forms, tau-confluence and especially the Cones and Foci method are indispensible. The second one is to verify that modal formulas are valid for given processes, for which parameterised boolean equation systems are essential.

The theory is implemented in the mCRL2 toolset, which is freely available under the extremely liberal Boost license. This toolset can be used to prove behavioural equivalences, check the validity of modal formulas, visualise behaviour in various ways, and of course also to do normal matters such as simulating behaviour and generating or reducing state spaces. The toolset has been used to design and analyse a plethora of applications, including control systems at CERN. Experience shows that the quality of programs developed using such tools goes up ten-fold.

Abstract Behavioral Specifications

Tuesday – June 23

Time: 14:00 – 17:30
Room: 2452 Pascal

Tutors:
Reiner Hähnle (Technische Universität Darmstadt)
Einar Broch Johnsen (University of Oslo)
Rudolf Schlatte (University of Oslo)

Brief description: ABS (for abstract behavioral specification) is a formal and executable language for modelling feature-rich, concurrent, object-oriented systems at an abstract, yet precise level. ABS has a clear and simple concurrency model that integrates tightly coupled as well as actor-style distributed communication. ABS abstracts away from specific datatype, I/O or scheduler implementations, but is a fully executable language and has code generators for Java, Haskell, Erlang, and Maude.

ABS supports product line-based software development by relating feature specifications to executable models. Thus it allows gapless modelling of product lines from feature analysis down to executable code. ABS also can model properties of its runtime environment in the form of abstract deployment components. These can be related to cost annotations and reflected into cost-sensitive scheduling instructions, which makes ABS ideally suited for computationally precise modelling of applications running in the cloud.

ABS has been designed with formal analysability in mind. This makes it possible, for example, to perform automatic resource estimation, deadlock analysis, and even functional verification of concurrent ABS models. In the tutorial we present the most important language features of ABS and illustrate them by examples. We will also demonstrate parts of the ABS tool chain, including code generation, code animation, deadlock analysis, and resource analysis. Participants will be able to download the ABS tools in the form of an Eclipse plugin and try them out hands-on, if they wish.

Theory and Practice of Runtime Verification

Tuesday – June 23

Time: 09:00 – 17:30
Room: 1416 Smalltalk

Tutors:
Martin Leucker (Universität zu Lübeck)
Daniel Thoma (Universität zu Lübeck)

Brief description: In this tutorial we give an introduction to the field of Runtime Verification. More specifically, we give a comprehensive and coherent assessment to Linear Temporal Logic-based monitor synthesis approaches. We cover both rewriting and automata-based techniques, each from a propositional as well as from a data perspective. Beyond a formal account we present applications, especially in the area of testing. To this end, we give a practical introduction to the tool JUnitRV, which combines traditional unit testing for Java with Runtime Verification techniques.

The tutorial is tailored towards researches in theoretical foundations as well as towards users of formal methods.

Maps

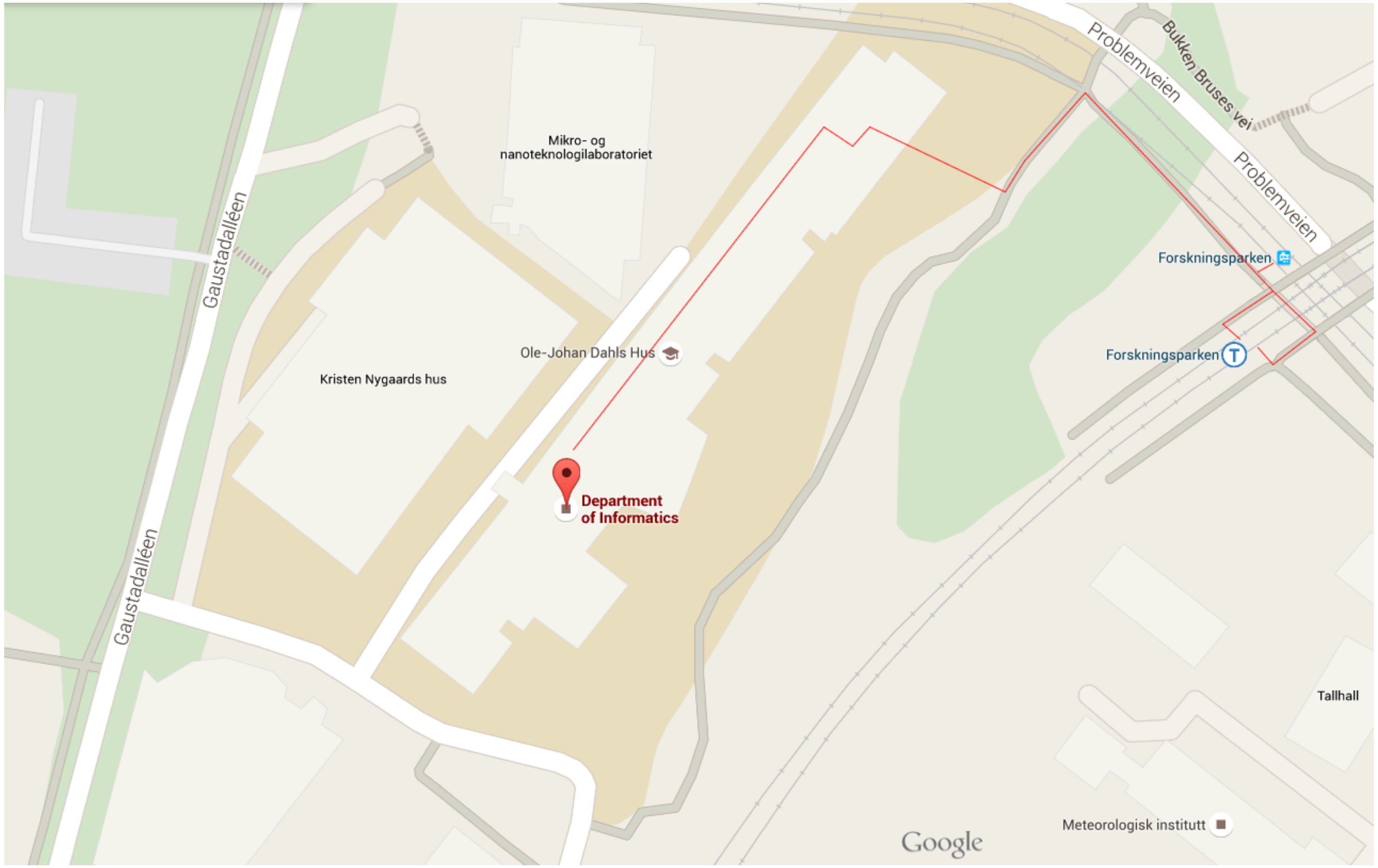


Figure 2: Conference venue – Ole-Johan Dahl's House

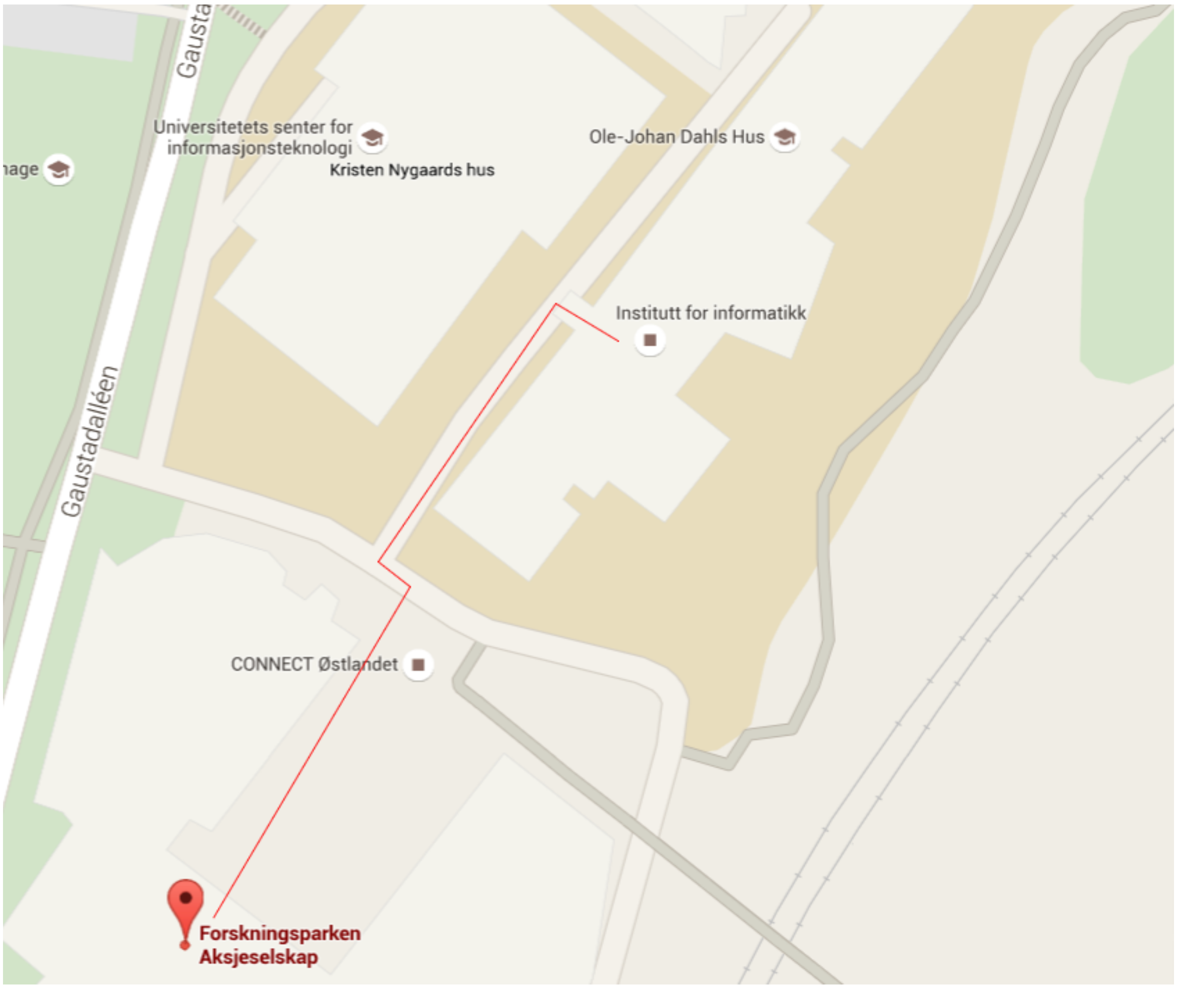

Gausta
Universitetets senter for informasjonsteknologi
Ole-Johan Dahls Hus
nage
Kristen Nygaards hus
Institutt for informatikk
Gaustadalléen
CONNECT Østlandet
Forskningsparken Aksjeselskap

Figure 3: Lunches – Forskningsparken

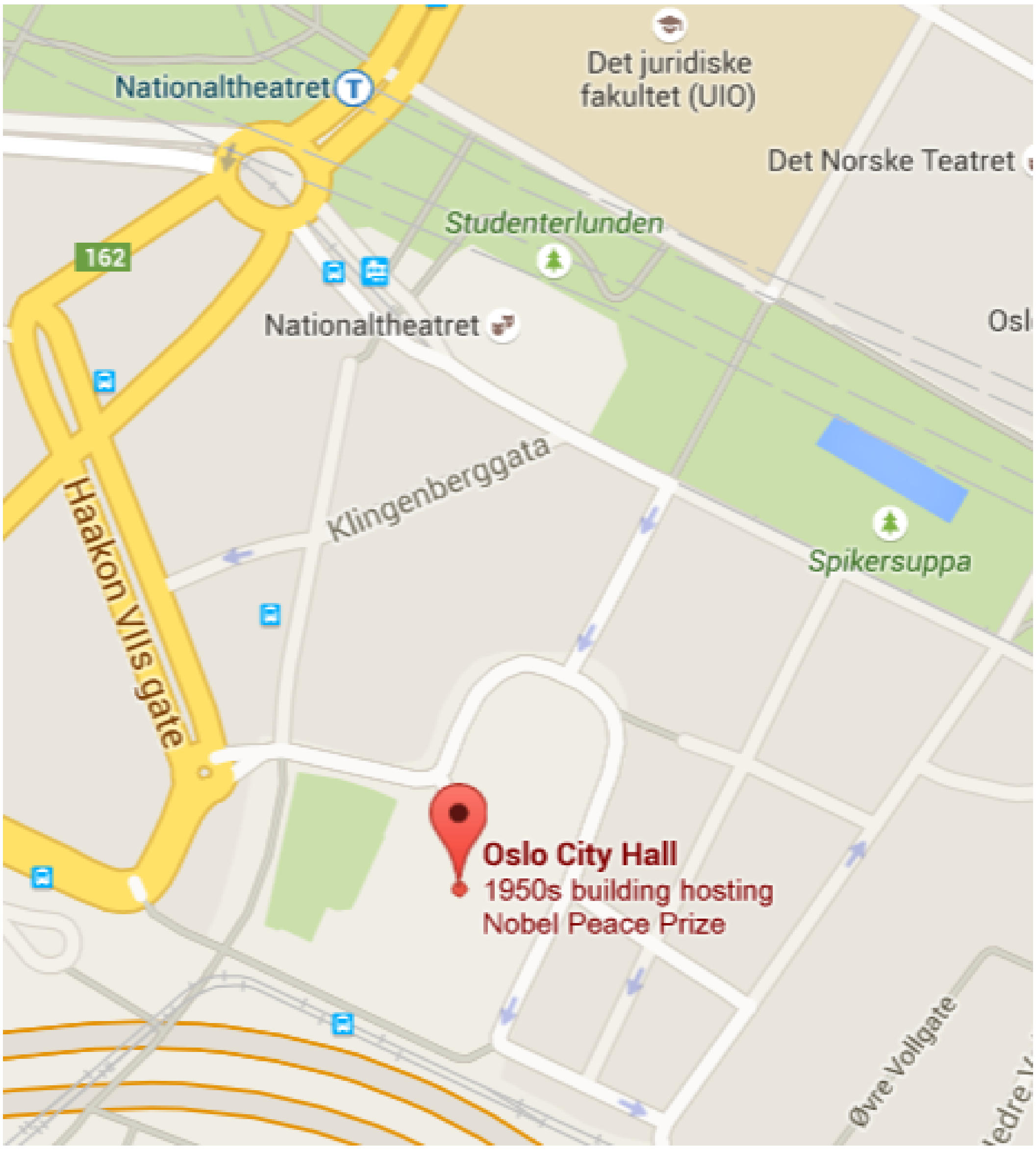


Figure 4: Reception – City Hall

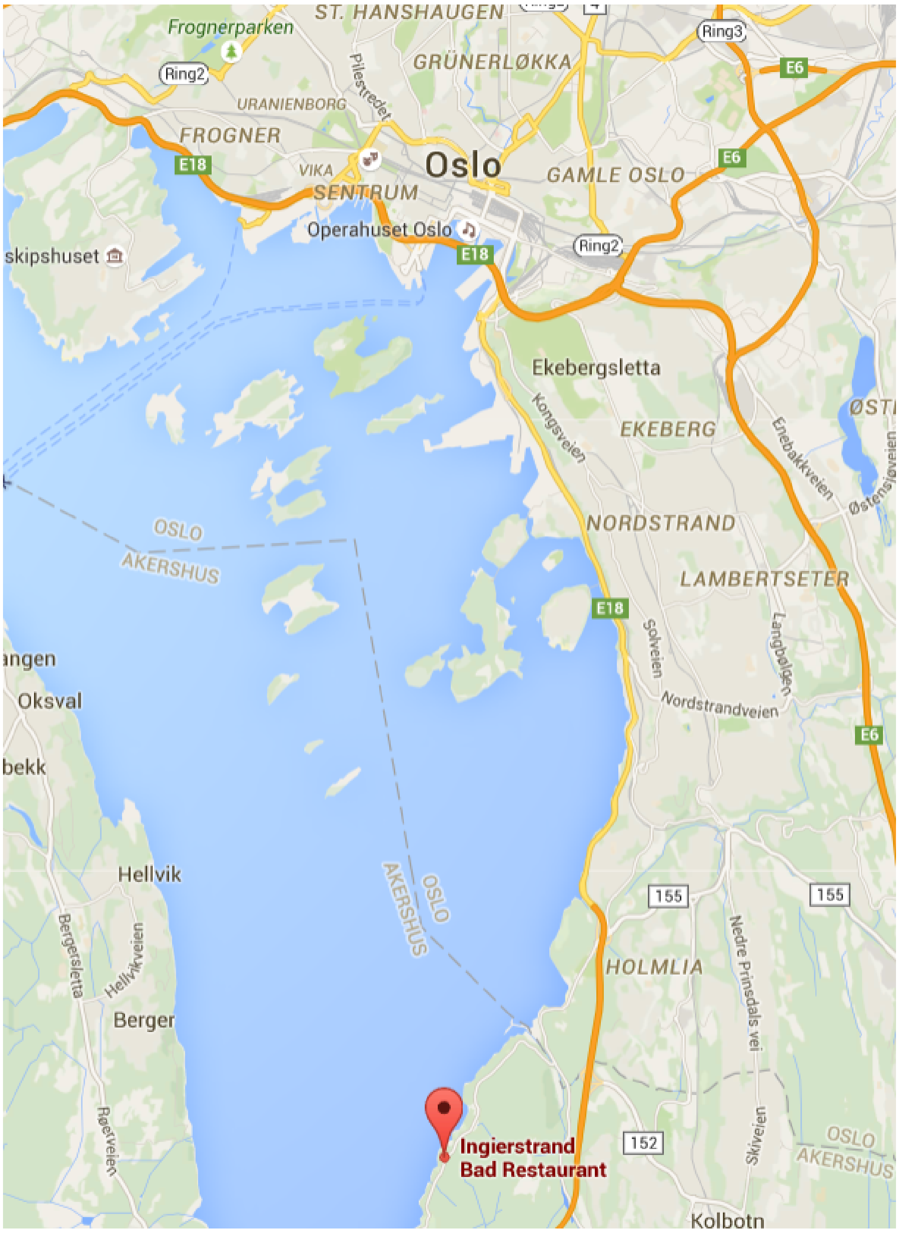


Figure 5: Banquet – Ingierstrand Bad

Sponsors of FM 2015

The Research Council of Norway

City of Oslo

Certus
Centre for Software Verification & Validation

CWI
Centrum Wiskunde & Informatica

CRNA
Centre for Resilient Networks & Applications

DNV·GL

Microsoft Research

FM
2015
CONFERENCE
HANDBOOK