



ECOOP'15 / CurryOn Prague Programme
29th edition, July 5–10

ECOOP
Prague 2015

FitO

Contents

1	General Information	1
1.1	Welcome to Prague	1
1.2	Organisation	2
1.2.1	Programme Chair	2
1.2.2	Programme Committee	2
1.2.3	Artifact Evaluation Committee Chairs	2
1.2.4	Artifact Evaluation Committee	2
1.2.5	Organising Committee	3
1.2.6	Official Sponsors of ECOOP 2015	4
1.3	Useful Information	5
1.3.1	Conference Address	5
1.3.2	Registration Desk	5
1.3.3	Badge Policy	5
1.3.4	Internet Access	5
1.3.5	ECOOP Proceedings	6
1.3.6	Dahl-Nygaard Awards	6
1.3.7	ECOOP Distinguished Paper	6
1.3.8	ECOOP Distinguished Artifacts	6
1.3.9	Restaurants	6
1.3.10	Taxi Companies	8
1.3.11	Transportation	8
1.3.12	Safety	8
1.3.13	Currency	9
1.3.14	Power	9
1.3.15	Emergency	9
1.3.16	Laws and Regulations	9
1.3.17	Liability	9
1.3.18	Conduct	9
1.4	Social Programme	9
1.4.1	Breaks	9
1.4.2	Lunches	10
1.4.3	Workshop Reception and Poster session	10
1.4.4	Curry On Party	10
1.4.5	ECOOP Banquet	11
1.4.6	Walks around town	11
1.5	Housing	11

1.5.1	Hotel BB Prague City	11
1.5.2	Novomestsky Hotel	12
1.5.3	Svehlova Dormitory	12
2	Scientific Program	13
2.1	ECOOP Research Track	13
2.2	Curry On	20
2.3	Summer School	26
2.4	7th International Workshop on Context Oriented Programming (COP)	29
2.5	Doctoral Symposium	30
2.6	MASPEGHI 2015: MechAnisms for SPEcialization, Generalization and inHerItance	31
2.7	Pharo 2015: Coding sprint	32
2.8	RIOT 2015: R Implementation, Optimization and Tooling Workshop	33
2.9	Fourth Annual Workshop on Tools for JavaScript Analysis	34
2.10	NetPL 2015: Networking meets Programming Languages	35
2.11	10th Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems Workshop (ICOOOLPS)	36
2.12	STOP 2015: International Workshop on Scripts to Programs	37
2.13	PLAS 2015: ACM SIGPLAN 10th Workshop on Programming Languages and Analysis for Security	38
2.14	DSLDI 2015	39
2.15	FTfJP 2015: 17th Workshop on Formal Techniques for Java-like Programs	40
2.16	ML4PL 2015	41
2.17	PLE 2015 Programming Language Evolution	42
2.18	Truffle 2015	43
2.19	Invited Speakers	44
3	Reference	49
3.1	Schedule Overview	49
3.2	ECOOP / CurryOn Program	50
3.3	Maps	55

1 | General Information

This installment of ECOOP is the 29th in ECOOP's history and the first ECOOP in the Czech Republic. The ECOOP conferences are organised locally and coordinated by AITO, a non-profit organisation dedicated to the advancement of object technology.

1.1 Welcome to Prague

It is with great pleasure that I welcome you to the ECOOP 2015 conference that is held for the first time in Prague.

ECOOP has become over its almost 30 year history the superior event for researchers, scholars, and companies who believe in object-oriented principles of design and programming. These object-oriented principles have had tremendous impact both in computer science research and in IT industry; and ECOOP has become the meeting place for those who advance the knowledge in this field.

CTU in Prague with its more 300 year history belongs to the premium European technical universities and Faculty of Information Technology is its youngest faculty. Since the beginning 5 years ago, object oriented programming has become an important part of our research and curricula portfolio at bachelor, master, and PhD level. Our organizing team of young scholars and students will do its best to make the Prague edition of ECOOP a success.

Pavel Tvrdik
Dean
Faculty of Information
Technology,
Czech Technical University
in Prague

1.2 Organisation

1.2.1 Programme Chair

John Boyland, University of Wisconsin, Milwaukee

1.2.2 Programme Committee

Stephanie Balzer, Carnegie Mellon University
Walter Binder, University of Lugano
Eric Bodden, Fraunhofer SIT and TU Darmstadt
Viviana Bono, University of Torino
Einar Broch Johnsen, University of Oslo
Dave Clarke, Uppsala University and KU Leuven
Werner Dietl, University of Waterloo
Danny Dig, Oregon State University
Irene Finocchi, Sapienza University Rome
Christian Hammer, Saarland University
Martin Hirzel, IBM Research
Marieke Huisman, University of Twente
Xuandong Li, Nanjing University
Francesco Logozzo, Facebook
Yi Lu, Oracle Labs
Todd Millstein, University of California, Los Angeles
Peter Müller, ETH Zurich
Bruno Oliveira, University of Hong Kong
Tamiya Onodera, IBM, Japan
Pavel Parizek, Charles University in Prague
Matthew Parkinson, Microsoft Research, UK
Christoph Reichenbach, Goethe University
Marco Servetto, Victoria University of Wellington
Friedrich Steimann, Fernuniversität
T. Stephen Strickland, Brown University
Mohsen Vakilian, Google
Tom Van Cutsem, Alcatel-Lucent Bell Labs
Harry Xu, University of California, Irvine
Nobuko Yoshida, Imperial College London

1.2.3 Artifact Evaluation Committee Chairs

Camil Demetrescu, Sapienza University of Rome
Matthew Flatt, University of Utah

1.2.4 Artifact Evaluation Committee

Karim Ali, Technical University at Darmstadt
Stefan Blom, University of Twente
Lubomir Bulej, Charles University

Nicolas Cardózo, Trinity College Dublin
Fernando Chirigati, NYU Polytechnic School of Engineering
Emilio Coppa, Sapienza University of Rome
Paolo G. Giarrusso, University of Tübingen
Raghavendra Kagalavadi, Oracle Labs
Du Li, Carnegie Mellon University
Sihan Li, University of Illinois at Urbana-Champaign
Stefan Marr, INRIA
Philip Mayer, Ludwig Maximilians University Munich
Cyrus Omar, Carnegie Mellon University
Daniel Perelman, University of Washington
Cosmin Radoi, University of Illinois
Christophe Scholliers, Vrije Universiteit Brussel
Wei Yang, University of Illinois at Urbana-Champaign

1.2.5 Organising Committee

Organising Chairs

Pavel Kordik, Czech Technical University
Tomas Kalibera, Northeastern University

Workshop Chairs

Stephane Ducasse, Inria, Lille
Nate Nystrom, University of Lugano

Poster Chair

Petr Maj, ReactorLabs

Publicity Chair

Ondrej Lhotak, University of Waterloo

Student Volunteer Chairs

Pavel Parizek, Charles University, Prague
Robert Pergl, Czech Technical University, Prague
Gregor Richards, University of Waterloo

Summer School Chair

James Noble, Victoria University of Wellington

Sponsorship Chair

Heather Miller, Ecole Polytechnique Federale de Lausanne

Housing Chair

Petr Maj, ReactorLabs

Comfy Chair

Jan Vitek, Northeastern University

Webmaster

Darya Kurilova, Carnegie Mellon University

1.2.6 Official Sponsors of ECOOP 2015

Silver Sponsors

Microsoft
Research

Oracle Labs

facebook

Google™

SAMSUNG RESEARCH AMERICA

Bronze Sponsors

R Studio

IBM Research

Goldman
Sachs

JetBrains


mozilla research

Other Partners



Northeastern University
College of Computer and Information Science

UWM
UNIVERSITY OF WISCONSIN
MILWAUKEE

ETH zürich

PiO

1.3 Useful Information

This section gathers information about the conference, and about health and safety.

1.3.1 Conference Address

Main Conference. The main conference venue is the **Marriott** in the historical center of Prague. It will host all events except the Sunday workshops. When using public transportation, look for stops "Masarykovo nadrazi", "Namesti Republiky" or "Mustek".

V Celnici 8

110 00 Prague 1

Googlemaps URL: <https://goo.gl/u0zsmH>

Sunday Workshops. The Faculty of Information Technology of the Czech Technical University will host the Sunday workshops. The venue is walking distance from the Subway stop Dejvicka (Green line - A, direction Nemocnice motol or just Nadrazi Veleslavin or Dejvicka). The entrance to the building is at the concave corner (behind the pillar in the direction in which the photo in Figure 1.2 was taken).

Thakurova 9

160 00 Prague 6

Googlemaps URL: <https://goo.gl/JcDdVT>



Figure 1.1: The Prague Marriott



Figure 1.2: The FIT building.

1.3.2 Registration Desk

Registration is in the Marriott, Sunday to Friday, 8:30 to 16:00. (Sunday morning at FIT)

1.3.3 Badge Policy

Weekdays, the conference venue is a hotel with considerable traffic. Thus, we must enforce a strict badge policy, badges should be worn at all times during the conference, and badges will be checked at the doors.

1.3.4 Internet Access

At the Marriott, the WIFI has a login of ECOOP and a password of 72015.

Sunday workshops' WIFI SSID is FIT-guests with password ecoop2015 (WPA2).

1.3.5 ECOOP Proceedings

Dagstuhl LIPICs proceedings are open access. Proceedings can be downloaded from: drops.dagstuhl.de/opus/volltexte/lipics-complete/lipics-vol37-ecoop2015-complete.pdf. Printed proceedings can be ordered ahead of time.

1.3.6 Dahl–Nygaard Awards

Dahl–Nygaard awards are handed out in conjunction with keynotes on Thursday and Friday. In 2004, AITO established an annual prize in the name of the Ole-Johan Dahl and Kristen Nygaard to honour their pioneering work on object-orientation.

The Senior Prize is awarded to *Bjarne Stroustrup* for the design of C++ and his commitment to the evolution of the C++ standard. The Junior Prize is awarded to *Alex Summers* for contributions to models, languages and tools for reasoning about programs.

The Members of the 2015 Dahl–Nygaard Award Committee were: Markku Sakkinen (Chair), Richard Jones, Gerti Kappel, Awais Rashid, and Tobias Wrigstad.

1.3.7 ECOOP Distinguished Paper

The program includes 31 excellent papers. The ECOOP PC selected *Towards Practical Gradual Typing* for its Distinguished Paper Award.

1.3.8 ECOOP Distinguished Artifacts

The ECOOP Distinguished Artifact Awards go to *A Pattern Calculus for Rule Languages: Expressiveness, Compilation, and Mechanization* and *Scalable and Precise Static Analysis of JavaScript Applications via Loop-Sensitivity*.

1.3.9 Restaurants

There may easily be over a hundred of restaurants within a 15 minutes walk from the Marriott. Many can be found from [TripAdvisor](#) and [Google](#). If you're looking for a non-smoking environment, always check with the waiters when entering the restaurant. Tap water is usually not offered, the cost model is that meals are cheap and drinks, particularly non-alcoholic, are expensive. You have to ask for ice with soft drinks explicitly. Tips are roughly 10%. Usually you can pay with cards but in very small places, better ask in advance. Cheap and average-priced places usually have lunch menus (“menu” in Czech)—this means cheaper food that is prepared faster. Alcoholic beverages tend to be cheap, but soft-drinks expensive—it is not uncommon for beer to be the cheapest beverage. Most restaurants have also non-alcoholic beers.

The list below has restaurants within walking distance from Marriott. The rankings are from TripAdvisor.

Expensive

The main dish including sides may be about 18-28€.

- **Kampa Park** (rank: 51). Na Kampe 8b, Prague 1. International food, sea-food, a nice view of the Charles Bridge. 25 mins walk.
- **Mlync** (rank: 28). Novotneho lavka 9, Prague 1. Next to the Charles Bridge, 18 mins walk.
- **Bellevue** (rank: 7). Smetanovo nabrezi 18, Prague 1. Very close to the Charles Bridge, nice view of the Prague Castle. 19 mins walk.
- **Francouzska Restaurace Art Nouveau** (rank: 523). Namesti republiky 5, Prague 1. French cuisine, very nice interior (Municipal House of Prague). 3 mins walk.
- **U Modre Kachnicky** (rank: 79). Michalska 16, Prague 1. Many variations on duck, quiet place a block away from main tourist routes. 13 mins walk.
- **Sarah Bernhardt** (rank: 49). U obecniho domu 1080/1, Prague 1.

Average Priced

The main dish including sides may be about 8-13€.

- **V Kolkovne** (rank: 584). V Kolkovne 8, Prague 1. Czech food (duck, pork, dumplings). A bit noisy environment. 11 mins walk.
- **Cafe Imperial** (rank: 47). Na Porici 15, Prague 1. Czech and international food. Very nice interior. 4 mins walk.
- **U Pinkasu** (rank: 779). Jungmannovo namesti 15/16, Prague 1. Traditional Czech food (duck, pork, “svickova”), also famous tavern (serves beer). 11 mins walk.
- **Konirna** (rank: 277). Maltezske namesti 10, Prague 1. Traditional Czech food, close to Charles Bridge. 25 mins walk.
- **Cafe Palanda** (rank: 57). Zlatnicka 1122/11, Prague 1. Burgers, sandwiches, a cheaper option in this category. 5 mins walk.
- **Velky Vinograf (wine bar)** (rank: 348). Senovazne namesti 23, Prague 1. The best wine-bar in Prague, also serves food (cheese, salami, etc). 6 mins walk.

Budget

The main dish including sides may be about 3-8€. The cheapest places are usually Pizzerias, local pubs (heavy local food, smoking, beer), or Chinese restaurants.

- **Ambiente Pizza Nuova** (rank: 367). Revolucni 1, Prague 1. A new and still not that cheap pizzeria, good food and decent service, offers all-you-can-eat salad/pasta options. Non-smoking. 4 min walk.
- **Pizzeria Rustica** (rank: 724). Opletalova 36, Prague 1. Pizzeria with decent food and service and very good prices. The smoking and non-smoking part are not well separated, but usually there is not much smoke inside. 7 mins walk.
- **Pizzeria Al Capone** (rank: 374). Na Porici 1933/36, Prague 1. Pizzeria with decent food and bearable service. The smoking and non-smoking part not perfectly separated, but usually there is not too much smoke in the non-smoking part. 6 mins walk.

- **Kolkovna Celnice** (rank: 473). V Celnici 4, Prague 1. Decent Czech food, modern interior, but gets rather noisy. 1 min walk.
- **Lokal Dlouha** (rank: 490). Dlouha 33, Prague 1. Czech food, beer. 7 mins walk.
- **Chinese Restaurant Sechuan**. Sokolovska 90/30, Prague 8. A small and very cheap Chinese restaurant, non-smoking, 12 mins walk from Marriott and 1 min from Hotel B&B Prague City.

1.3.10 Taxi Companies

The following are established taxi companies. Order taxis via phone (in English), stop one on the street may get you higher prices. Uber is available. AAA allows ordering on-line and also getting a quote.

- AAA Taxi, +420 222 333 222 <http://www.aaataxi.cz/en/>
- Affordable Taxi, +420 226 000 226 <http://www.nejlevnejsi-taxi.cz/en/>

1.3.11 Transportation

Plane. Vaclav Havel Airport (PRG) has two terminals: Terminal 1 is for non-Schengen countries and Terminal 2 for Schengen countries. The terminals are walking distance and are served by taxis, the airport shuttle, and public transportation. PRG is 30 minutes drive or 50 minutes by public transport from the conference hotel. If you are staying at the conference hotel, the CEDAZ airport shuttle is convenient. It runs every 30 minutes and costs 150CZK (6€).

Train. Prague Main Railway Station is located in the city center, 12 minutes walk from the Marriott. The public transportation stop is named “Hlavni nadrazi”.

Bus. Florenc Bus Terminal also called “UAN Florenc” or “Ustredni Autobusove Nadrazi Praha—Florenc” is 9 min walk from the Marriott. It serves many international bus connections. The public transport stop is named “Florenc”.

Parking. Parking in the city center is possible in dedicated car parks but expensive. Some hotels provide parking for prices lower or comparable to public car parks (Marriott about 22 €/day). A list of car parks is at <http://bezpecneparkovani.cz/>. You can also park for free in the street, but not in the center—finding out exactly where it is allowed is a challenge.

Public Transportation. Google maps helps finding public transport stops and schedules, but once you know the names of the stops, you may check with a local connection finder at <http://www.dpp.cz/en/journey-planner/> to learn about detours/constructions. The fares structure is complicated, the simplest is to use one-way transfer tickets (32 CZK for 90 minutes). Time-stamp your ticket before entering a subway station or after entering a bus.

1.3.12 Safety

Generally Prague is safe but there are several well known risks to travelers. Pick-pocketing is common in the city center on the streets, in shops, in public transport, close to ATMs or exchange offices—organized groups target mostly tourists; the highest risk is in crowded areas and sometimes the crowds are created artificially by well organized criminals. Simply keep your wallet well hidden and under control. A very high risk of significant ripoff exists in currency exchange offices. It is not just that their exchange rate may not be a bargain for the customer, but they often intentionally mislead or outright lie about what

their rate is and refuse to abort the transaction once the customer gets the money. A decent exchange office is EXCHANGE at Kaprova 14/13, Prague 1. Please do not ever leave your belongings unattended.

1.3.13 Currency

The local currency is Czech crown (CZK).

1.3.14 Power

The mains voltage is 230V (all of European Union) and the socket type is E (like continental Europe).

1.3.15 Emergency

The general emergency number is 112. The operators speak English. Czech emergency numbers are: 155 (ambulance), 158 (police). In case of accident, you'll be treated in a hospital whether insured or not.

1.3.16 Laws and Regulations

Everyone is required by law to carry an ID card at all times . The police may ask anyone for their ID card at their own discretion. There is zero tolerance to drinking and driving. Drugs are illegal.

1.3.17 Liability

The organisers are not liable for damages and/or losses of any kind which may be incurred by the conference delegates or by any other individuals accompanying them, both during the official activities as well as going to/from the conference. Delegates are responsible for their own safety and belongings.

1.3.18 Conduct

Please show respect for those around you. This applies to both in-person and online behavior. The central goal of a conference is to encourage the open exchange of ideas through free expression. This requires an environment that enables all to participate without fear of harassment. ECOOP is dedicated to providing a harassment-free conference experience for everyone, regardless of gender, gender identity and expression, sexual orientation, disability, physical appearance, body size, race, age or religion (or lack thereof). All communication should be appropriate for a technical audience, including people of many different backgrounds. Sexual language, innuendo, and imagery is not appropriate for any conference venue, including talks. If you are being harassed, notice that someone else is being harassed, or have any other concerns, please contact a member of conference staff immediately. Conference staff will be happy to help participants contact hotel/venue security or local law enforcement, provide escorts, or otherwise assist those experiencing harassment to feel safe for the duration of the conference. We value your attendance.

1.4 Social Programme

The Workshop reception is free, while the Curry On Party and the ECOOP Banquet both require tickets.

1.4.1 Breaks

Food and drinks are served during breaks, Monday to Friday, at the Marriott for all attendees.

1.4.2 Lunches

Lunch is served at the **Plzencka Restaurace**, Monday to Friday at 12–13:30, 250 meters from the hotel.

Náměstí Republiky 5,
110 00, Praha 1

Googlemaps: <https://goo.gl/maps/yMuB7>

The Marriott will have food stations next to the meeting rooms (cheap but not included in registration).

The menu is: (Monday) Beef Soup with home made noodles, vegetable and beef meat, Roast Pork Larded served with sauerkraut and dumplings. (Tuesday) Potato Soup with wild mushrooms, Roasted Chicken Breast steamed vegetables with butter and honey, parsley potatoes. (Wednesday) Goulash, Penne with basil pesto and sun-dried organic tomatoes. (Thursday) Split pea soup, Beef Goulash Prague Style roasted onion garnish, dumplings. (Friday) Cabbage soup, Pork Schnitzel with mashed potatoes.

While we strive to provide Vegetarian options, the Czech Republic is still a meat-and-potato-based economy.

1.4.3 Workshop Reception and Poster session

A reception for all attendees will be held on Tuesday July 7th at 18:00 in the Marriott together with poster presentations. Light food and beverages will be served.

1.4.4 Curry On Party

The Curry On Party takes place on Monday July 6th from 19.30 at **Velka Klasterni Restaurace**. Drinks and a buffet dinner will be served.

Strahovske nadvori 302, 118 00 Prague 1
www.klasternirestaurace.cz

Googlemaps URL: <https://goo.gl/maps/YQppg>

The restaurant is on top of the hill where the Prague castle and the Strahov Monastery are located. It can be reached in 50 minutes on foot, and somewhat faster by tram. We recommend taking the tram to get there and walking to return to your hotel.

From Marriott through the historic center of Prague (Foot): start on the Royal Way (Kralovska cesta) to Old Town Square (Staromestske namesti), across Charles Bridge, and then through Mala Strana via Nerudova street, passing the entrance to the castle. Then one can walk up on Loretanska street to Pohorelec, then take stairs up through the building no 8 on the left, entering Strahovske nadvori.

From Marriott through the castle (Foot): take smaller streets of the Old Town (Stare mesto) to get to the Manes Bridge (Manesuv most), cross the bridge, walk right diagonally through a small park, cross the street (Klarov), walk along it to the right, all the way to stairs to the castle (Stare zamecke schody). You can exit the castle through the main entrance on Loretanska street (and then follow instructions above).

From Marriott with the tram: tram number 5 from Namesti Republiky to Malostranska (the final stop), it is just three stops and it will take you close to where the stairs up to the castle start (Stare zamecke schody), so you can then walk through the castle like the previous option given.

If you wanted to avoid walking up the hill, from Malostranska take tram 22 direction Bila Hora, exit at Pohorelec (it is the closest tram stop to the restaurant), or exit already at Prazsky hrad and you can still

walk partially through the castle, but avoiding the exercise.

1.4.5 ECOOP Banquet

The ECOOP Banquet takes place on Thursday July 9th from 19.30 at **Hergertova Cihelna**. Drinks and dinner will be served.

Cihelna 2b, 118 00 Prague 1

Googlemaps URL: <https://goo.gl/maps/Xln8L>

The restaurant is next to Franz Kafka Museum. The best way to get there from the Marriott is on foot, through the historic centre of Prague. One can take the usual route of most visitors to Prague starting on the Royal Way (Kralovska cesta) to Old Town Square (Staromestske namesti), across Charles Bridge, and then through Mala Strana. One can also avoid this main tourist route and take the smaller streets of the Old Town to Manes Bridge (Manesuv most) and cross this bridge instead. Both ways take about 25-30mins, but there is a lot to see on the way.

1.4.6 Walks around town

Our student volunteers will offer walking tours around town. Information will be posted by the registration desk.

1.5 Housing

1.5.1 Hotel BB Prague City

Address: Prvniho Pluku 29, Prague 8. Nearest metro station: Florenc (B - yellow). From the airport: take bus 119 to Nadrazi Veleslavin (final stop), change to metro A (green, direction Depo Hostivar), change at Mustek to metro B (yellow, direction Cerny Most), get off at Florenc. Ticket: 32 CZK.

Getting to the Marriott: Walking: 14 min (1.4km). Public Transport: Walk to Florenc, take tram 3 to Masarykovo Nadrazi, then walk to Marriott. Ticket: 24 CZK. (See Figure 1.3.)

Getting to the FIT: Walk to Florenc metro station, take train direction Zlicin and change at Mustek to green line A, direction Nemocnice Motol (or just Dejvicka or Nadrazi Veleslavin), get off at Dejvicka. Ticket: 24 CZK.

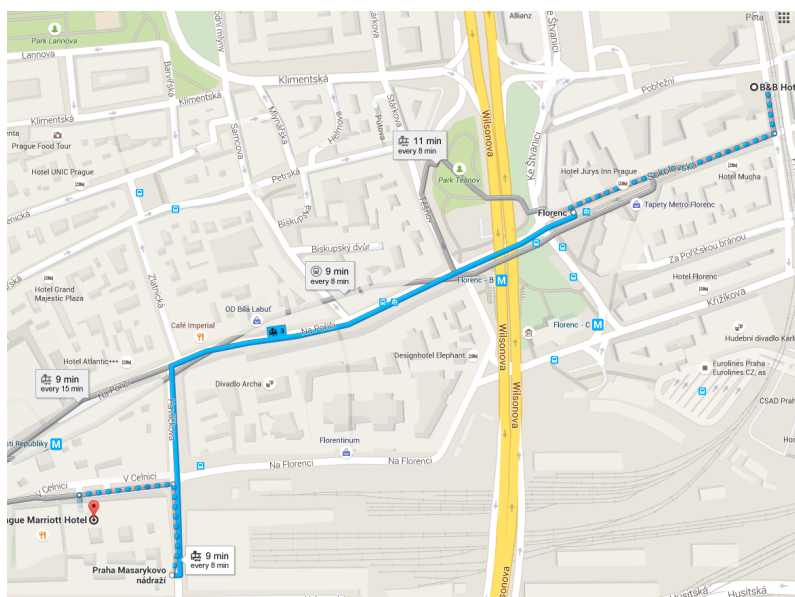


Figure 1.3: Getting to the Marriott from Hotel BB.

1.5.2 Novomestsky Hotel

Address: Reznicka 1890/4, Prague 1.

Nearest metro: Namesti Republiky (B - yellow). From the airport: take bus 119 to Nadrazi Veleslavin (final stop), change to metro A (green, direction Depo Hostivar), change at Mustek to yellow line B, direction Zlicin, get off at Narodni Trida. Ticket: 32 CZK.

Getting to the Mariott: Walking: 19 min (1.6km). Public Transport: walk to Narodni Trida metro station, take train direction Cerny Most, get off at Namesti Republiky, walk to the Mariott hotel. Ticket 24 CZK.

Getting to the FIT: Walk to Narodni Trida, take train in direction to Cerny Most, change at Mustek to green line direction Nemocnice Motol (or just Dejvicka or Nadrazi Veleslavin), get off at Dejvicka. Ticket: 24 CZK.

1.5.3 Svehlova Dormitory

Address: Slavikova 1499/22, Prague 3.

Nearest metro : Jiriho z Podebrad (A - green). From the airport: take bus 119 to Nadrazi Veleslavin (final stop), change to metro A (green, direction Depo Hostivar), get off at Jiriho z Podebrad. Ticket price is 32 CZK.

Checkin: noon to midnight, Checkout: till 10:00 am. Students requiring extended checkin/checkout times should contact housing chair with their details ASAP. Once you get to the dormitory, please give your name at the reception and you will be shown to your room. Be advised that English capabilities of the receptionists at the dorm might be limited, but your name should be all you need. Internet is provided for free, each room has an ethernet socket, but no wifi. In order to use it, bring your own ethernet cable. We recommend to bring your own towel and toiletries with you.

Getting to the Mariott: Walking: 18 min

(1.7km) Figure 1.4 Public Transport: Either take the metro (19 mins), or walk to Husinecka Tram station (750m), then take tram 5 or 26 and get off at Masarykovo Nadrazi station. Walk to Mariott from there. Ticket: 24 CZK.

Getting to the FIT: Public Transport: Take the metro from Jiriho z Podebrad direction Nemocnice Motol (or just Dejvicka or Nadrazi Veleslavin), get off at Dejvicka station. Ticket price 24 CZK.

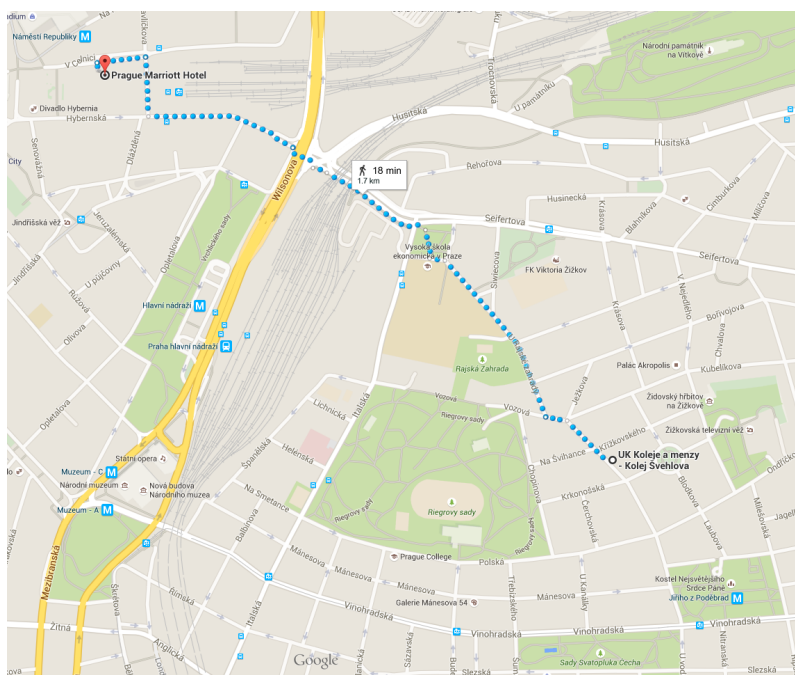


Figure 1.4: Getting to the Mariott from Svehlova – walking.

2 | Scientific Program

2.1 ECOOP Research Track

Session I – Gradual Typing

Wednesday, July 8, 10:30-12:00, Bohemia

Towards Practical Gradual Typing

Asumu Takikawa *Northeastern University*
Daniel Feltey *Northeastern University*
Earl Dean *Indiana University*
Robby Findler *Northeastern University*
Matthew Flatt *University of Utah*
Sam Tobin-Hochstadt *Indiana University*
Matthias Felleisen *Northeastern University*



Over the past 20 years, programmers have embraced dynamically-typed programming languages. By now, they have also come to realize that programs in these languages lack reliable type information for software engineering purposes. This paper presents an implementation of a gradual type system for a full-featured class-based language as well as a novel performance evaluation framework for gradual typing.

TreatJS: Higher-Order Contracts for JavaScripts

Matthias Keil *University of Freiburg*
Peter Thiemann *University of Freiburg*



TreatJS is a language embedded, higher-order contract system for JavaScript which enforces contracts by runtime monitoring. Beyond providing the standard abstractions for building higher-order contracts, TreatJS's novel contributions are its guarantee of non-interfering contract execution, its systematic approach to blame assignment, its support for contracts in the style of union and intersection types, and its notion of a parameterized contract scope.

Trust, but Verify: Two-Phase Typing for Dynamic Languages

Panagiotis Vekris *University of California, San Diego*
Benjamin Cosman *University of California, San Diego*
Ranjit Jhala *University of California, San Diego*

A key challenge when statically typing so-called dynamic languages is the ubiquity of value-based overloading, where a given function can dynamically reflect upon and behave according to the types of its arguments. Thus, to establish basic types, the analysis must reason precisely about values, but in the presence of higher-order functions and

polymorphism, this reasoning itself can require basic types. We address this chicken-and-egg problem by introducing the framework of two-phased typing.

Session II – Implementation

Wednesday, July 8, 13:30-15:00, Bohemia

Concrete Types for TypeScript

Gregor Richards *University of Waterloo*

Francesco Zappa Nardelli *Inria*

Jan Vitek *Northeastern University*

TypeScript extends the JavaScript programming language with a set of optional type annotations that are, by design, unsound and, that the TypeScript compiler discards as it emits plain JavaScript code. This paper details an alternative design for TypeScript, one where it is possible to support the same degree of dynamism, but also where types can be strengthened to provide runtime guarantees.

Simple and Effective Type Check Removal through Lazy Basic Block Versioning

Maxime Chevalier-Boisvert *DIRO, University of Montréal*

Marc Feeley *DIRO, University of Montréal*

In order to maximize performance, dynamic language VM implementations must attempt to eliminate redundant dynamic type checks. However, type inference analyses are often costly and involve tradeoffs between compilation time and resulting precision. This paper introduces lazy basic block versioning, a simple JIT compilation architecture which effectively removes redundant type checks from critical code paths.

Loop tiling in the presence of exceptions

Abhilash Bhandari *IIT Madras*

V Krishna Nandivada *IIT Madras*

Exceptions in OO languages provide a convenient mechanism to deal with anomalous situations. However, many of the loop optimization techniques cannot be applied in the presence of conditional ‘throw’ statements in the body of the loop, owing to possible cross iteration control dependences. In this paper, we present a generalized scheme to do exception-safe loop optimizations and present a scheme of optimized exception-safe loop tiling (oESLT), as a specialization thereof.

Session III – Objects

Wednesday, July 8, 15:30-17:00, Bohemia

A Theory of Tagged Objects

Joseph Lee *Carnegie Mellon University*

Jonathan Aldrich *Carnegie Mellon University*

Troy Shaw *Victoria University of Wellington*

Alex Potanin *Victoria University of Wellington*



Foundational models of object-oriented constructs typically model objects as records with a structural type. In this paper, we explore how to model statically-typed object-oriented languages that support dynamic class creation using foundational constructs of type theory.

Brand Objects for Nominal Typing

Timothy Jones *Victoria University of Wellington*
 Michael Homer *Victoria University of Wellington*
 James Noble *Victoria University of Wellington*



Combinations of structural and nominal object typing have focused on extending existing nominal, class-based systems with structural subtyping. The typical rules of nominal typing do not lend themselves to such an extension. We have implemented brand objects to explicitly type objects, using existing features of the structurally typed language Grace, along with a static type checker which treats the brands as nominal types.

Transparent Object Proxies in JavaScript

Matthias Keil *University of Freiburg*
 Peter Thiemann *University of Freiburg*



Proxies are the swiss army knives of object adaptation. One important question in the design of a proxy API is whether a proxy object has its own identity. We examine the issue with transparency in detail, consider various use cases for proxies, discuss different approaches to obtain transparency, and propose two designs that cannot be bypassed by the programmer but that require modest modifications in the JavaScript engine.

Session IV – Analysis I

Wednesday, July 8, 17:30-18:30, Bohemia

Access-rights Analysis in the Presence of Subjects

Paolina Centonze *Iona College*
 Marco Pistoia *IBM Research*
 Omer Tripp *IBM Research*

This paper introduces Subject Access Rights Analysis (SARA), a novel analysis algorithm for statically computing the permissions required by subjects at run time. We have applied SARA to 348 libraries in a commercial enterprise application server written in Java that consists of >2 million lines of code and is required to support the Java permission- and subject-based security model.

Variability Abstractions: Trading Precision for Speed in Family-Based Analyses

Aleksandar S. Dimovski *IT University of Copenhagen, Denmark*
 Claus Brabrand *IT University of Copenhagen, Denmark*
 Andrzej Wasowski *IT University of Copenhagen, Denmark*

Family-based (lifted) data-flow analysis for Software Product Lines (SPLs) is capable of analyzing all valid products without generating any of them explicitly. In this paper, we introduce variability abstractions defined as Galois connections and use abstract interpretation as a formal method for the calculational-based derivation of approximate lifted analyses of SPL programs.

Session V – Developer Assistance

Thursday, July 9, 10:30-12:00, Bohemia

Optimization Coaching for JavaScript

Vincent St-Amour *Northeastern University*
 Shu-yu Guo *Mozilla Corporation*



The performance of dynamic object-oriented programming languages such as JavaScript depends heavily on highly optimizing just-in-time compilers. We show how to solve the problem of silent optimization failures and explain how to create a so-called optimization coach for an object-oriented just-in-time-compiled programming language.

PerfBlower: Quickly Detecting Memory-Related Performance Problems via Amplification

Lu Fang *University of Science and Technology, China*

Liang Dou *East China Normal University*

Harry Xu *University of California, Irvine*

Performance problems in managed languages are extremely difficult to find. Despite many efforts to find those problems. We describe PerfBlower, a general performance testing framework that allows developers to quickly test Java programs to find memory-related performance problems.

Hybrid DOM-Sensitive Change Impact Analysis for JavaScript

Saba Alimadadi *University of British Columbia*

Ali Mesbah *University of British Columbia*

Karthik Pattabiraman *University of British Columbia*

Performing change impact analysis on JavaScript applications is challenging due to features such as the interplay with the DOM, event-driven and dynamic function calls, etc. The results of our empirical study show that the DOM-related and dynamic features of JavaScript need to be taken into consideration in the analysis since they affect change impact propagation. We propose a DOM-sensitive hybrid change impact analysis technique through a combination of static and dynamic analysis.

Session VI – Type Systems

Thursday, July 9, 13:30-15:00, Bohemia

Intensional Effect Polymorphism

Yuheng Long *Iowa State University*

Yu David Liu *State University of New York (SUNY) Binghamton*

Hriday Rajan *Iowa State University*

Type-and-effect systems are a powerful tool for program construction and verification. We describe intensional effect polymorphism, a new foundation for effect systems that integrates static and dynamic effect checking.

Type Inference for Place-Oblivious Objects

Riyaz Haque *University of California, Los Angeles*

Jens Palsberg *University of California, Los Angeles*

Place-oblivious objects safely access other objects without knowledge of their place. In response, we present the first type system for place-oblivious objects along with an efficient inference algorithm.

Asynchronous Liquid Separation Types

Johannes Kloos *Max Planck Institute for Software Systems*

Rupak Majumdar *Max Planck Institute for Software Systems*

Viktor Vafeiadis *Max Planck Institute for Software Systems*

We present a refinement type system for reasoning about asynchronous programs manipulating shared mutable state. Our type system guarantees the absence of races and the preservation of user-specified invariants using a combination of two ideas: refinement types and concurrent separation logic.

Session VII – Parallelism

Thursday, July 9, 15:30-17:00, Bohemia

The Eureka Programming Model for Speculative Task ParallelismShams Imam *Rice University*Vivek Sarkar *Rice University*

We describe the Eureka Programming Model (EuPM) that simplifies the expression of speculative parallel tasks, and is especially well suited for parallel search and optimization applications. We provide a clean semantics for such “eureka-style” computations (EuSCs) in general structured task parallel programming models.

Cooking the Books: Formalizing JMM Implementation RecipesGustavo Petri *Purdue University*Jan Vitek *Northeastern University*Suresh Jagannathan *DARPA*

The Java Memory Model (JMM) is intended to characterize the meaning of concurrent Java programs. Because of the model’s complexity, however, its definition cannot be easily transplanted within an optimizing Java compiler. In response, the JSR-133 Cookbook, an informal guide to realizing the principles underlying the JMM was developed. We formalize the cookbook, and show that the rules for compiling Java onto Power are inconsistent with the JMM.

Defining Correctness Conditions for Concurrent Objects in Multicore ArchitecturesBrijesh Dongol *Brunel University*John Derrick *University of Sheffield*Lindsay Groves *Victoria University of Wellington*Graeme Smith *The University of Queensland*

We present a formal framework for defining correctness conditions for multicore architectures, covering both standard conditions for totally ordered memory and newer conditions for relaxed memory, which allows them to be expressed in uniform manner, simplifying comparison.

Session VIII – Empirical Studies

Thursday, July 9, 17:30-18:30, Bohemia

The Good, the Bad, and the Ugly: An Empirical Study of Implicit Type Conversions in JavaScriptMichael Pradel *Technische Universität Darmstadt*Koushik Sen *University of California, Berkeley*

Most programming languages support situations where a value of one type is converted into a value of another type without any explicit cast. We dynamically analyze hundreds of JS programs and find that coercions are widely used and that most coercions are likely to be harmless.

The Love/Hate Relationship with the C Preprocessor: An Interview StudyFlavio Medeiros *Federal University of Campina Grande*Christian Kästner *Carnegie Mellon University*Márcio Ribeiro *Federal University of Alagoas*Sarah Nadi *Technische Universität Darmstadt*Rohit Gheyi *Federal University of Campina Grande*

The C preprocessor has received strong criticism in academia, among others regarding separation of concerns, error proneness, and code obfuscation, but is widely used in practice. Many alternatives to the preprocessor exist, but have not been adopted in practice. Since developers continue to use the preprocessor despite all criticism and research, we ask how practitioners perceive the C preprocessor.

Session IX – Abstraction

Friday, July 10, 10:30-12:00, Bohemia

A Pattern Calculus for Rule Languages: Expressiveness, Compilation, and Mechanization

Avraham Shinnar *IBM Research*

Jerome Simeon *IBM Research*

Martin Hirzel *IBM Research*



This paper introduces a core calculus for pattern-matching in production rule languages: the Calculus for Aggregating Matching Patterns (CAMP). CAMP is expressive enough to capture modern rule languages such as JRules, including extensions for aggregation. We show how CAMP can be compiled into a nested-relational algebra (NRA), with only minimal extension. This paves the way for applying relational techniques to running rules over large stores.

Global Sequence Protocol: A Robust Abstraction for Replicated Shared State

Sebastian Burckhardt *Microsoft Research*

Daan Leijen *Microsoft Research*

Jonathan Protzenko *Microsoft Research*

Manuel Fähndrich *Google*

In the age of cloud-connected mobile devices, users want responsive apps that read and write shared data everywhere, even if network connections are slow or unavailable. The solution is to replicate data and propagate updates asynchronously. Unfortunately, such mechanisms are notoriously difficult. We present GSP (global sequence protocol), an operational model for replicated shared data. GSP is simple and abstract enough to serve as a mental reference model, and offers fine control over the asynchronous update propagation.

Streams a la carte: Extensible Pipelines with Object Algebras

Aggelos Biboudis *University of Athens*

Nick Palladinos *Nessos Information Technologies, SA*

George Fourtounis *University of Athens*

Yannis Smaragdakis *University of Athens*



Streaming libraries have become ubiquitous in object-oriented languages. All such libraries, however, suffer in terms of extensibility: there is no way to change the semantics of a streaming pipeline without changes to the library code. Furthermore, in some languages it is not even possible to add new operators without changing the library. We address such extensibility shortcomings with a new design.

Session X – Verification

Friday, July 10, 15:00-16:30, Bohemia

Lightweight Support for Magic Wands in an Automatic Verifier

Malte Schwerhoff *ETH Zurich*

Alexander J. Summers *ETH Zurich*



Permission-based verification logics such as separation logic have led to the development of many practical verification

tools over the last decade. We show how to integrate support for the magic wand into an automatic verifier, requiring low specification overhead from the tool user, due to a novel approach for choosing footprints for magic wand formulas automatically.

Modular Verification of Finite Blocking in Non-terminating Programs

Pontus Boström *Abo Akademi University*

Peter Müller *ETH Zurich*

We present a verification technique for finite blocking in non-terminating programs. The key idea is to track explicitly whether a thread has an obligation to perform an operation that unblocks another thread.

Modular Termination Verification

Bart Jacobs *iMinds – DISTRINET, KU Leuven*

Dragan Bosnacki *Eindhoven University of Technology*

Ruurd Kuiper *Eindhoven University of Technology*

We propose an approach for the modular specification and verification of total correctness properties of object-oriented programs. We start from an existing program logic for partial correctness based on separation logic and abstract predicate families. We extend the approach to a concurrent setting, by incorporating an existing approach for verifying deadlock-freedom of channels and locks.

Session XI – Analysis II

Friday, July 10, 17:00-18:30, Bohemia

Framework for Static Analysis of PHP Applications

David Hauzar *Charles University in Prague*

Jan Kofroň *Charles University in Prague*



We present static analysis framework for PHP, automatically resolving features common to dynamic languages and thus reducing the complexity of defining new static analyses.

Adaptive Context-sensitive Analysis for JavaScript

Shiyi Wei *Virginia Tech*

Barbara Ryder *Virginia Tech*

Our empirical study suggests that there is no single context-sensitive analysis that always produces precise results for JavaScript applications. This observation motivated us to design an adaptive analysis, selecting a context-sensitive analysis from multiple choices for each function.

Scalable and Precise Static Analysis of JavaScript Applications via Loop-Sensitivity

Changhee Park *Korea Advanced Institute of Science and Technology*

Sukyoung Ryu *Korea Advanced Institute of Science and Technology*



In this paper, we present Loop-Sensitive Analysis (LSA) that improves the analysis scalability by enhancing the analysis precision in loops. LSA distinguishes loop iterations as many as needed by automatically choosing loop unrolling numbers during analysis. We formalize LSA in the abstract interpretation framework and prove its soundness and precision.

2.2 Curry On

Session Keynote I – Room 1

Monday, July 6, 9:00–9:45, Bohemia

Programs Wanted: Dead or Alive

Gilad Bracha *Google*

For many years, work on programming tools and environments has been regarded with disdain by many in the academic programming language community. The recent flood of interest in live programming is changing this situation for the better, albeit belatedly. Rather than thinking about programming languages in isolation, we need to think of programming language systems with a holistic view of the programming experience.

Session I – Room 1

Monday, July 6, 10:05–12:25, Bohemia

Kotlin: Challenges in Language Design

Andrey Breslav *JetBrains*

Programming language design in its modern sense is not limited to syntax nor to execution semantics: the presence of IDEs, build systems and other tools opens up new dimensions in the design space. This talk is about interesting problems we are facing while working on the Kotlin – a statically typed language compiled to the JVM and JavaScript.

Structured Synchronous Programming

Francisco Sant’Anna *PUC-Rio, Brazil*

Functional Reactive Programming (FRP) has modernized the dataflow style, inspiring a number of languages and libraries, such as Flapjax, Rx, React, and Elm. In this talk, we present a contemporary outlook of synchronous imperative programming through the programming language Céu. Céu extends classical structured programming with lexical compositions of lines of execution that can await.

Encapsulating replication, high concurrency and consistency with CRDTs

Marc Shapiro *INRIA/LIP6*

We propose a theoretically sound approach, Conflict-Free Replicated Data Types (CRDTs). A CRDT encapsulates replication and concurrency, and safe and convergent by construction, thanks to some simple mathematical properties of conflict resolution. CRDTs remain responsive, available and scalable despite high network latency, faults, or disconnection.

Session I – Room 2

Monday, July 6, 10:05–12:25, Bohemia

Crystal: a programming language for humans and computers

Ary Borenszweig *Manas*

Juan Wajnerman *Manas*

Crystal is a new programming language that focuses on developer productivity, type safety and execution performance. It is statically checked and compiles to native code. It combines a global type inference algorithm, compile-time

macros, and compile-time type introspection. It provides a Garbage Collector, uses LLVM as its backend and doesn't run on a Virtual Machine.

Akka Typed: Between Session Types and the Actor Model

Roland Kuhn *Typesafe*

We introduce the Akka Typed project, a large jump in the platform's evolution that includes not only typed Actor references but also a principled approach to behavioral composition of individual Actors. We discuss how far this can be pushed in the direction of protocols and session types.

Pull > Push: Please stop polluting our imperative languages with pure concepts

Ron Pressler *Parallel Universe*

The past few years have seen an influx of ideas from functional programming languages to mainstream languages. Not only lambdas and higher-order functions have found their way into Java and other languages, but even more advanced concepts like monads. We'll explore the validity and applicability of those concepts to imperative languages (or, rather, lack thereof), and the motivation for their inclusion.

Session II – Room 1

Monday, July 6, 13:50–15:20, Bohemia

Empowering Spreadsheet Users with Probabilistic Programs

Andy Gordon *Microsoft Research*

Probabilistic programming is a declarative form of machine learning: the user writes a probabilistic model of their data as a short piece of code, while the compiler turns the code into an efficient inference routine to learn and predict properties of the data.

It Probably Works

Tyler McMullen *Fastly*

Probabilistic algorithms are an awesome and underused tool for engineers building very large scale systems. Making calculations or consistency guarantees “with high probability” instead of “exactly” is often easier to scale and surprisingly reliable. This talk presents two widely useful probabilistic algorithms which have seen successful deployments at massive scale.

Session II – Room 2

Monday, July 6, 13:50–15:20, Bohemia

Speed at a Price: The Evolution of V8 and the Challenges of Research in a Billion User VM

Ben Titzer *Google*

Google Chrome catapulted JavaScript performance from a toy scripting language to a fast and smooth platform for rich applications, igniting a browser war that's seen at least four vendors competing neck-and-neck with increasingly performant JSVMs. This talk will outline in broad strokes the initial design pressures in V8 and catalog some turning points along its history as it evolved to support new applications.

Coccinelle for the Working Programmer

Julia Lawall *INRIA/LIP6/UPMC/Sorbonne University*

Coccinelle is a program matching and transformation tool for C code, targeting bug finding and fixing, as well as

the automation of pervasive source code evolutions. It can also be used for applications such as code understanding and calculation of software metrics, making it relevant to both software developers and software researchers.

Session III – Room 1

Monday, July 6, 16:00–17:30, Bohemia

Java @ Twitter—a bird’s eye view

Tony Printezis *Twitter*

Twitter’s infrastructure consists of a swarm of services onto of a bevy of managed runtimes, starting with Ruby but moving to Java and Scala. Targeting the JVM allows developers to quickly develop and deploy reliable code.

Post-FRP Frontend Programming

Bodil Stokke

Functional Reactive Programming was a thing that happened in the 90s. We’ve been rehashing these ideas, usually not even getting them right, for the past 20 years, and very little has happened to advance the state of the art. Very little, but that’s still something. We’re going to explore some of the advances we’ve made recently, with a particular focus on PureScript and the Halogen library.

Session III – Room 2

Monday, July 6, 16:00–17:30, Bohemia

Servant: a type-level DSL for web APIs

Julian Arni *Zalora*

Servant is a new, type-level Haskell EDSL for describing web APIs. It allows for arbitrarily sophisticated descriptions of a web server’s API. We describe what it’s like to have data at the type-level, and in particular how that design is an outgrowth of the literature on what is known as the “expression problem”.

Making Embedded Domain Specific Languages a Practical Reality

Jurriaan Hage *Utrecht University*

A showstopper for embedded domain specific languages in a strongly typed setting is that when the program contains a type error, type error diagnosis is communicated to the domain programmer in terms of the host language, not the domain.

Session Keynote II – Room 1

Monday, July 6, 17:40–18:30, Bohemia

Julia

Jeff Bezanson *MIT*

Stefan Karpinski *MIT*

Session Keynote III – Room 1

Tuesday, July 7, 9:00–9:45, Bohemia

JS @ 20

Brendan Eich

Session IV – Room 1

Tuesday, July 7, 10:05–12:25, Bohemia

Bits of Advice for VM WritersCliff Click *H₂O*

This is a talk about the choices one makes when building a Virtual Machine. Many of these choices aren't even obviously being made when you first get the machine running—it's not until years later when you look at your limitations that you even realize there was a choice.

Pyro—Typing Python in PythonAlex Muscar *Lyst*

Pyro, our in-house Python 2 type checker written in Python adopts a structural typing discipline, and it uses row polymorphism to represent objects.

How to be a good host: miniKanren as a case studyDan Friedman *Indiana University*Jason Hemann *Indiana University*

The new criterion for choosing a language may be not what it can do, but what kinds of languages it can support. Using miniKanren as an example, we survey the more than 40 hosts that have “Kanren” implementations, and demonstrate what separates an adequate host from a great one.

Session IV – Room 2

Tuesday, July 7, 10:05–12:25, Bohemia

Some of the Things That Macros DoZach Tellman *Factual*

It is a truth universally acknowledged that a language in possession of homoiconic syntax must be in want of macros. And yet, most industry languages have eschewed anything resembling homoiconicity, leaving syntax transformations either entirely out of reach, or the realm of a brave few. We'll explore the use of macros in the Clojure community, characterizing what's been done, what's been avoided, and what may lie ahead.

QuickCheck: from invention to productThomas Arts *Quviq*

QuickCheck is a new way of software testing by automatically generating test cases from properties. We'll show how QuickCheck grew from invention to product and what obstacles one finds on the way when doing so.

Data Manipulation using Programming By Examples and Natural LanguageSumit Gulwani *Microsoft Research*

The paradigms of programming by examples (PBE) and programming by natural language (PBNL) have the potential to make data wrangling a delightful experience for the masses.

Session V – Room 1

Tuesday, July 7, 13:50–15:20, Bohemia

What—if anything—have we learned from C++?Bjarne Stroustrup *Morgan Stanley***Tern: Practical Type Inference for JavaScript Editing**

Marijn Haverbeke

Tern implements a type inference engine for JavaScript. It can be plugged into various editors to provide language-aware integration. To be able to do this in near real-time, it uses an optimized form of data-flow analysis. The analysis is far from sound, but immensely helpful regardless.

Session V – Room 2

Tuesday, July 7, 13:50–15:20, Bohemia

Coding for Types: The Universe Pattern in IdrisDavid Christiansen *IT Univerity of Copenhagen*

A dependently typed language allows programmers to write functions that compute types from ordinary data. We say that the data is a code for the resulting types, and that the collection of types selected by the codes is a closed universe.

Everything old is new again: Quoted domain specific languagesPhilip Wadler *University of Edinburgh*

Fashions come, go, return. We describes a new approach to domain specific languages, called QDSL, that resurrects two old ideas: quoted terms for domain specific languages, from McCarthy’s Lisp of 1960, and the subformula property, from Gentzen’s natural deduction of 1935.

Session VI – Room 1

Tuesday, July 7, 16:00–17:30, Bohemia

LLVM for JavaScriptFilip Pizio *Apple*

Can we reuse existing C compiler infrastructures to generate good code for a dynamic language? I’ll dive into the engineering of the WebKit Fourth-Tier JIT compiler, a production-strength compiler for JavaScript that uses the LLVM C compiler optimization pipeline and backend.

Rust: A Type System You Didn’t Know You WantedFelix Klock *Mozilla*

Rust is a new system programming language that provides memory safety and data-race freedom while offering

efficiency and low-level control comparable to that of C and C++. Rust allows for safe systems programming, including use of concurrent threads with shared data.

Pony: making it easy to write efficient, concurrent, data-race free programs

Sylvan Clebsch, Sebastian Blessing, Sophia Drossopoulou *Imperial College*

Pony is an object oriented, imperative, actor based programming language which makes it easy to write super-fast programs.

Session VI – Room 2

Tuesday, July 7, 16:10–18:20, Bohemia

GS Collections: Echoes of Smalltalk’s Past

Alex Iliev *Goldman Sachs*

We present the history of a Java collections library developed at Goldman Sachs. GS Collections is now one of the top open source libraries of its kind, and is influencing the evolving design of the Java core libraries.

Java everywhere again—with DukeScript!

Anton Epple *NetBeans*

In this session you’ll learn how to manage development costs and cost of ownership for cross-platform single-source applications that run on Desktop, Apple iOS, Google Android and embedded devices using NetBeans HTML for Java APIs.

Scaling SQL Databases Beyond Limits with PostgreSQL

Valentin Gogichashvili *Zalando*

We’ll cover the main aspects of how Zalando’s agile engineering teams use PostgreSQL. I’ll discuss the techniques and best practices that we have developed over time to scale SQL-based databases without the need to give up on the power of SQL, and introduce the tools that we have open-sourced to make the work of our developers and database administrators easier.

Session Keynote IV – Room 1

Tuesday, July 7, 18:30–19:20, Bohemia

Let’s be mainstream! User-focused design in Elm

Evan Czaplicki *Prezi*

So one of my primary goals is for Elm to be extraordinarily easy to learn and use productively. I want a new user to start playing with real code in minutes. I want programmers to feel confident in Elm in a few days. This talk will demonstrate some concrete cases where Elm uses language design, library design, terminology, and culture to accelerate learning and ultimately help people make delightful products.

2.3 Summer School

Session I –

Wednesday, July 8, 10:30-12:00, Moravia

Verifying Correctness of Statefull Networks

Mooly Sagiv *Tel Aviv University*

Modern computer networks use states to store temporary status of the network. This allows to enforce complicated forwarding policies and to enhance networks in a modular way. However, this can lead to subtle errors and makes network verification hard. I will survey techniques for dealing with statefull networks and show a system for verifying networks using SMT solvers.

Session II –

Wednesday, July 8, 13:30-15:00, Moravia

The Evolving Design of Pyret

Shriram Krishnamurthi *Brown University*

Pyret is a new programming language built to support a family of curricula pioneered by “How to Design Programs” (Felleisen, ea). These curricula emphasize type-driven program design, testing, and specification. We will discuss this style of programming and provide a snapshot of the resulting language. We will emphasize how enabling testing drives some of our decisions.

Session III –

Wednesday, July 8, 15:30-17:00, Moravia

Nimble yet Systematic Trading

Robert Grimm *Goldman Sachs*

Running a successful trading business requires being nimble and systematic at the same time. Being nimble means continually monitoring one’s risk exposure, and swiftly adjusting to changing market as well as regulatory conditions. Yet being systematic means also accounting for every trade through its lifecycle, from order to settlement, while following applicable rules and regulations at every step. I discuss how SecDB, Goldman Sachs’ trade and risk management system, directly helps meet the two requirements. First, SecDB integrates an in-house scripting language, called Slang, with a set of globally replicated object databases. Second, the issue tracking, regression testing, and code review systems are implemented in Slang and let developers easily follow a common workflow while also subjecting code to a rigorous vetting process.

Session IV –

Wednesday, July 8, 17:00-18:30, Moravia

Modern Concurrent Separation Logics

Philippa Gardner *Imperial College London*

Consider increment program $\text{inc}(x) \parallel \text{inc}(x)$, where $\text{inc}(x)$ increments x atomically. It is difficult to show that, with pre-condition $x = 0$, the post-condition is $x = 2$. RGSep combines separation logic with rely-guarantee to reason about more fine-grained concurrent algorithms and CAP reasoning added abstraction. This talk provides an

introduction to modern CAP reasoning, giving an overview of the strengths of current approaches and highlighting places where there is room for improvement.

Session V –

Thursday, July 9, 10:30-12:00, Moravia

Declarative Probabilistic Programming

Molham Aref *LogicBlox*

I will summarize our work on a declarative programming language that offers native language support for expressing predictive (e.g. machine learning) and prescriptive (e.g. combinatorial optimization) analytics. The presentation gives an overview of the platform and the language. In particular, it focuses on the important role of integrity constraints, which are used not only for maintaining data integrity, but also, for example, for the specification of complex optimization problems and probabilistic programming.

Session VI –

Thursday, July 9, 13:30-15:00, Moravia

Synthesis and Verification for Everyone

Emina Torlak *University of Washington*

Synthesis and verification are revolutionizing the way we program. They are helping us create programs that run on energy-efficient hardware, that grade student work and provide feedback, and that automate end-user programming by example. But these tools are hard to build — all of them are made by highly skilled researchers with expertise in many fields, from formal methods to programming languages to software engineering. I present a new approach to constructing tools for synthesis and verification that requires little beyond modern programming skills.

Session VII – Thursday, July 9, 15:30-17:00

Moravia,

Graceful Programming — Teaching Introductory Programming

Kim Bruce *Pomona College*

What if you could teach novices programming with a language that was explicitly designed to help students understand how to do object-oriented programming without saddling them with obscure and complicated syntax and convoluted constructs? We've constructed such a language that can be used to teach novices using the features of modern programming languages, but without all the overhead of existing industrial strength languages. It is object-based, supports first-class closures, is gradually typed, and is designed to be simple, both syntactically and semantically. Dialects can be used to introduce new features or to restrict the language. We taught an introductory course using Grace this fall and found it to be both effective and a pleasure to teach. We will give a quick overview of the language, explain how we taught it, discuss student feedback on the use of Grace.

Session VIII –

Thursday, July 9, 17:00-18:30, Moravia

Noise-based Testing of Concurrent Software

Tomas Vojnar *Brno University of Technology*

Testing of concurrent software written in programming languages like Java and C/C++ is a challenging task due to

the many possible interactions among threads. A simple, cheap, and effective approach that addresses this challenge is testing with noise injection which influences the scheduling so that different interleavings of concurrent actions are witnessed. We will review the principles of testing based on noise injection as well as various advances achieved in the area.

Session IX –

Friday, July 10, 10:30-12:00, Moravia

Taming Uncertainty, Scale, and Change: A Programming Language Perspective

Suresh Jagannathan *DARPA*

The modern-day software ecosystem is a messy and chaotic one. Among other things, it includes an intricate stack of sophisticated services and components, susceptible to frequent (and often incompatible) upgrades and patches; emerging applications that operate over large, unstructured, and noisy data; and, an ever growing code base replete with latent defects and redundancies. Devising novel techniques to tame this complexity, and improve software resilience, trustworthiness, and expressivity in the process, is a common theme actively being explored by ongoing DARPA programs. I'll present three such efforts - PPAML (Probabilistic Programming Advancing Machine Learning), MUSE (Mining and Understanding Software Enclaves), and BRASS (Building Resource Adaptive Software Systems).

Session X –

Friday, July 10, 15:00-16:30, Moravia

Programming with Millions of Examples

Eran Yahav *Technion*

The vast amount of code available on the web is increasing on a daily basis. Open-source hosting sites such as GitHub contain billions of lines of code. Community question-answering sites provide millions of code snippets with corresponding text and metadata. The amount of code available in executable binaries is even greater. I will cover recent research trends on leveraging such “big code” for program analysis, program synthesis and reverse engineering. Along the way, we will consider a range of semantic program representations based on symbolic automata, tracelets and numerical abstractions as well as notions of code similarity based on these representations.

Session XI –

Friday, July 10, 17:00-18:30, Moravia

Herdin Weak Memory Cats

Jade Alglave *University College London*

There is a joke where a physicist and a mathematician are asked to herd cats. The physicist starts with an infinitely large pen which he reduces until it is of reasonable diameter yet contains all the cats. The mathematician builds a fence around himself and declares the outside to be the inside. Defining memory models is akin to herding cats: both the physicist's or mathematician's attitudes are tempting, but we cannot rely on one more than on the other. I will show ways of defining formal models for weak memory, within a generic framework in which one can represent Sequential Consistency, x86, Power, ARM and C++.

2.4 7th International Workshop on Context Oriented Programming (COP)

Context information plays an increasingly important role in our information-centric world. Software systems must adapt to changing contexts over time, and must change even while they are running. Unfortunately, mainstream programming languages and development environments do not support this kind of dynamic change very well, leading developers to implement complex designs to anticipate various dimensions of variability. Starting from this observation, Context-Oriented Programming (COP) has emerged as a solution to directly support variability depending on a wide range of dynamic attributes, making it possible to dispatch run-time behaviour on any property of the execution context.

Committee

Carl Friedrich Bolz, <i>King's College</i> (Chair)	Tetsuo Kamina, <i>Ritsumeikan Uni</i>
Tomoyuki Aotani, <i>Tokyo IT</i>	Jens Lincke, <i>Hasso Plattner Institute</i>
Nicolas Cardózo, <i>Trinity College</i>	Somayeh Malakuti, <i>TU Dresden</i>
Coen De Roover, <i>Vrije Uni Brussel</i>	Hidehiko Masuhara, <i>Tokyo IT</i>
David H. Lorenz, <i>Open Uni / Technion IIT</i>	Harold Ossher, <i>IBM</i>
Robert Hirschfeld, <i>Hasso Plattner Institute</i>	Guido Salvaneschi, <i>TU Darmstadt</i>
Atsushi Igarashi, <i>Kyoto Uni</i>	Friedrich Steimann, <i>Fernuniversität</i>
	Naoyasu Ubayashi, <i>Kyushu Uni</i>

Program: Sunday 5th of July

09:30–10:30	Keynote: ContextJS and Lively Kernel: Safely Evolving a Self-supporting Development Environment with COP. <i>Jens Lincke</i>
11:00–11:30	Context-Oriented Image Processing. <i>Didier Verna, François Ripault</i>
11:30–12:00	A Study of Context-Oriented Programming for Applying to Robot Development. <i>Harumi Watanabe, Midori Sugaya, Ikuta Tanigawa, Nobuhiko Ogura, Kenji Hisazumi</i>
12:00–12:30	Towards a Decoupled Context-Oriented Programming Language for the Internet of Things. <i>Baptiste Maingret, Frédéric Le Mouël, Julien Ponge, Nicolas Stouls, Jian Cao, Yannick Loiseau</i>
13:30–14:00	Connecting Object Constraints with Context-oriented Programming: Scoping Constraints with Layers and Activating Layers with Constraints. <i>Stefan Lehmann, Tim Felgentreff, Robert Hirschfeld</i>
14:00–14:30	Method Safety Mechanism for Asynchronous Layer Deactivation. <i>Tetsuo Kamina, Tomoyuki Aotani, Hidehiko Masuhara, Atsushi Igarashi</i>
14:30–15:00	Context Slices: A lightweight discovery module for adaptations. <i>Nicolas Cardózo, Siobhán Clarke</i>
15:30–16:00	Efficient Layered Method Execution in ContextAmber. <i>Matthias Springer, Jens Lincke, Robert Hirschfeld</i>
16:00–16:30	Type-Safe Layer-Introduced Base Functions with Imperative Layer Activation. <i>Tomoyuki Aotani, Tetsuo Kamina, Hidehiko Masuhara</i>
16:30–17:00	Discussion

2.5 Doctoral Symposium

This is a full-day event of interactive presentations where the morning session will be dedicated to student talks and the afternoon session will be dedicated to invited talks from those in academia and the industry on a variety of topics related to PhD studies and doing research. Besides the formal presentations and discussions in sessions, there will be plenty of opportunities for informal interactions during breaks and lunch.

Academic Panel

Jonathan Aldrich, *Carnegie Mellon Uni*
Shriram Krishnamurthi, *Brown Uni*

Yu David Liu, *State Uni of New York*

Committee

Timothy Jones, *Victoria Uni of Wellington* (Chair)
Oliver Bracevac, *TU Darmstadt*
Dmitri Makarov, *Uni of Lugano*

Amanj Sherwany, *Uni della Svizzera italiana*
Shiyi Wei, *Virginia Tech*

Program: Sunday 5th of July

Session 1: Student Talks	
09:30–09:40	Welcome
09:40–10:05	A virtual machine for testing compilation/recompilation protocols in multiple inheritance. <i>Julien Pagès</i>
10:05–10:30	Domain Behavior Analysis of Software Using Code Relative Detection. <i>Fang-Hsiang Su</i>
Session 2: Student Talks	
11:00–11:25	Transformation language for expressing automated API Migration. <i>Krishna Narasimhan</i>
11:25–11:50	Analyzing Nonconformances in Contract-Based Programs. <i>Alysson Milanez</i>
11:50–12:30	Fast and Lean Immutable Data Structures. <i>Michael Steindorfer</i>
Session 3: Invited Talks	
13:30–14:00	The story of Arjun Guha, or: The arc of a research project. <i>Shriram Krishnamurthi</i>
14:00–15:00	So you want to be an industrial researcher? <i>Mario Wolczko</i>
Session 4: Invited Talks	
15:30–16:00	Happy Graduate School Years. <i>Yu David Liu</i>
16:00–16:30	Tales from Dissertationland and the Job Hunt. <i>Jonathan Aldrich</i>
16:30–16:40	Formation of next year's committee

2.6 MASPEGHI 2015: MechAnisms for SPEcialization, Generalization and in-HerItance

MASPEGHI 2015 will continue the discussion about mechanisms for managing and manipulating specialization and generalization hierarchies: single and multiple inheritance of interfaces or implementation, support for specialization/generalization, mixins, traits, delegation, virtual classes, etc. We are concerned with both the design of inheritance mechanisms, and the difficulties of implementing and using it.

These concerns are reflected differently by disciplines such as databases, knowledge discovery and representation, modeling and design methods, and object-based programming languages. The emphasis is sometimes on modeling the problem domain, and sometimes on organizing the computational artifacts that simulate the domain.

Despite the wide use of specialization hierarchies, there is no standard methodology for constructing and maintaining them independently from the domains that they represent and the artifacts that they organize. This workshop will provide a forum for researchers from a variety of domains to learn from each other and work together to develop such a methodology.

Committee

Andrew Black,	<i>Portland State Uni</i> (Chair)	Gerti Kappel,	<i>Technical Uni of Vienna</i>
Gabriela Arévalo,	<i>DCyT</i>	Stein Krogdahl,	<i>Uni of Oslo</i>
Kim Bruce,	<i>Pomona College</i>	Manuel Oriol,	<i>ABB Corporate Research</i>
Erik Ernst,	<i>Google</i>	Markku Sakkinen,	<i>Uni of Jyväskylä</i>
Robert Godin,	<i>Uni Québec à Montréal</i>	Bernhard Thalheim,	<i>Christian-Albrechts-Uni</i>
Martin Hitz,	<i>Alpen-Adria-Universität Klagenfurt</i>	Roberto Zicari,	<i>Goethe University Frankfurt</i>
Marianne Huchard,	<i>LIRMM</i>		

Program: Sunday 5th of July

09:30–09:50	Introduction
09:50–10:30	Position paper 1: jUCM: Universal Class Morphing. <i>Aggelos Biboudis, George Fourtounis, and Yannis Smaragdakis</i>
11:00–11:45	Analyzing inheritance hierarchies through Formal Concept Analysis: A 22-years walk in a landscape of conceptual structures. <i>Marianne Huchard</i>
11:45–12:30	The Expression Problem, Gracefully. <i>Andrew Black</i>
13:30–14:30	Invited talk: Combining Inheritance and Modularity. <i>Gilad Bracha</i>
14:30–15:00	Position paper 2: On Feature Protection in C++, Java and Eiffel. <i>Markku Sakkinen</i>
15:30–16:40	Discussion on approaches to multiple inheritance
16:40–17:00	Workshop retrospective

2.7 Pharo 2015: Coding sprint

Pharo (<http://pharo.org/community>) is used by more than 20 Universities worldwide, more than 10 research groups and 50 companies. Pharo has an industrial consortium to sustain the project development (<http://consortium.pharo.org>).

More important Pharo is Yours in the sense that you can really contribute to it and learn a lot on your coding skills. Join, have fun and learn with us. At ECOOP 2015, we will not run a workshop on Pharo but a coding sprint! Participants to the coding sprint will help improving Pharo.

Committee

Guillermo Polito, *INRIA, Lille* (Presenter)

Program: Sunday 5th of July

09:00–10:00	Welcome + Presentation
10:00–10:30	Coding Example
11:00–12:30	Coding Session 1
13:30–15:00	Coding Session 2
15:30–17:00	Coding Session 3

2.8 RIOT 2015: R Implementation, Optimization and Tooling Workshop

RIOT 2015 is a one-day workshop dedicated to exploring future directions for development of R language implementations and tools. The goals of the workshop include, but are not limited to, sharing experiences of developing different R language implementations and tools and evaluate their status, exploring possibilities for increasing involvement of the R users community in the efforts of constructing different R implementations, identifying R language development and tooling opportunities enabled by the emerging implementations, and discussing future directions for the R language. The workshop will consist of a number of short talks and discussions and will bring together developers of R language implementations and tools.

Committee

Adam Welc,	<i>Oracle Labs</i> (Chair)	Karl Millar,	<i>Google</i>
Jan Vitek,	<i>Northeastern Uni</i>	David Padua,	<i>UIUC</i>
JJ Alaire,	<i>R Studio</i>	Sue Ranney,	<i>Microsoft</i>
Alex Bertram,	<i>Bedatadriven</i>	Indrajit Roy,	<i>HP Labs</i>
Lee Edlefsen,	<i>Microsoft</i>	Michael Sannella,	<i>TIBCO</i>
Michael Haupt,	<i>Oracle Labs</i>	Lukas Stadler,	<i>Oracle Labs</i>
Richard Jones,	<i>Uni of Kent</i>	Duncan Temple Lang,	<i>UC Davis</i>
Mick Jordan,	<i>Oracle Labs</i>	Luke Tierney,	<i>Uni of Iowa</i>
Tomas Kalibera,	<i>Northeastern Uni</i>	Hadley Wickham,	<i>R Studio</i>
Helena Kotthaus,	<i>TU Dortmund</i>	Mario Wolczko,	<i>Oracle Labs</i>
Petr Maj,	<i>ReactorLabs</i>	Peng Wu,	<i>Huawei America Lab</i>

Program: Sunday 5th of July

08:45–09:00	Introduction. <i>Adam Welc and Jan Vitek</i>
	Session 1: Implementation
09:00–09:30	The CXXR project: Status and Future. <i>Karl Millar</i>
09:30–10:00	The FastR Runtime—R as a Citizen of the Truffle Family of Languages. <i>Lukas Stadler</i>
10:00–10:30	Supporting the "Rapi" C-language API in an R-compatible engine. <i>Michael Sannella</i>
	Session 2: Analysis
11:00–11:30	Detecting Memory Protection Errors in GNU-R using Static Checking. <i>Tomas Kalibera</i>
11:30–12:00	Distributed Performance Analysis for R. <i>Helena Kotthaus</i>
12:00–12:30	Feature Specific Profiling in the R Language. <i>Leif Andersen</i>
	Session 3: Optimization
14:00–14:30	R as a Query Language. <i>Alex Bertram</i>
14:30–15:00	Vectorization of Apply Operations for the Exploitation of the Efficient Interpretation of R. <i>Haichuan Wang</i>
15:30–17:30	Open Discussion at RIOT

2.9 Fourth Annual Workshop on Tools for JavaScript Analysis

JSTools will bring together participants from academia and industry working on analysis of JavaScript and its dialects to share ideas and problems, with a focus on presentations of shareable infrastructure created by the participants. We also aim to involve developers working on JavaScript dialects such as TypeScript to share their perspective.

Committee

Julian Dolby, *IBM Research*
Shu-yu Guo, *Mozilla Corp*

Christian Hammer, *Saarland Uni*
Michael Pradel, *TU Darmstadt*

Program: Monday 6th of July

10:10–10:15	Opening Remarks.
	Session: Theory
10:15–10:45	ES5→IVL, Principled Translation using Operational Semantics. <i>Daiva Naudziuniene</i>
	Session: Invited Talks at STOP
10:50–11:35	Experiments with Strengthening JavaScript. <i>Andreas Rossberg</i>
11:35–12:20	Flow: a static type checker for JavaScript. <i>Avik Chaudhuri</i>
	Session: Invited Talks at STOP II
13:50–14:35	JavaScript in the Small. <i>Satish Chandra</i>
	Session: Practical Analysis
14:40–15:10	HybriDroid: Analysis Framework for Android Hybrid Applications. <i>Sukyoungh Ryu</i>
15:10–15:40	Jalangi: A Dynamic Analyses Framework for JavaScript. <i>Koushik Sen</i>
	Session: Practical Analysis II
16:00–16:30	EventRacer: Scalable Analysis for Event-Driven Systems. <i>Martin Vechev</i>
16:30–17:00	Stateless Model Checking for JavaScript. <i>Anders Møller</i>
17:00–17:30	MemInsight: Platform-Independent Memory Profiling for JavaScript. <i>Manu Sridharan</i>
17:30–18:00	Visualizing the interactions of client and server JS code. <i>Saba Alimadadi</i>
18:00–18:30	Closing Remarks

2.10 NetPL 2015: Networking meets Programming Languages

The NetPL workshop will provide a forum to bring together researchers and practitioners from the fields of programming languages, formal methods, and networking. Recent technological trends, such as software-defined networking and network functions virtualization, have created an opportunity for researchers in these traditionally separate communities to collaborate, applying their diverse perspectives towards the development of novel networking applications. The program consists of invited talks, with an emphasis on encouraging engaging technical discussions amongst the participants.

Committee

Marco Canini, *Université catholique de Louvain*
 Robert Soulé, *Università della Svizzera Italiana*

Program: Monday 6th of July

9:00–9:10	Welcome. <i>Marco Canini and Robert Soulé</i>
9:10–9:50	Enabling End Host Network Functions. <i>Hitesh Ballani</i>
10:10–11:10	Tierless Abstractions for Programming Devices. <i>Shriram Krishnamurthi</i>
11:10–11:50	Immutable Distributed Infrastructure with Unikernels. <i>Anil Madhavapeddy</i>
11:50–12:30	Program Synthesis for Network Updates. <i>Pavol Černý</i>
14:00–15:00	The State of (Open Source) SDN and Programming Languages Opportunities. <i>Colin Dixon</i>
15:00–15:40	From Dumb to Smarter Switches in Software Defined Networks: Towards a Stateful Data Plane. <i>Antonio Capone</i>
16:00–17:00	Managing Dynamic Networks: Distributed or Centralized Control?. <i>Roger Wattenhofer</i>
17:00–17:30	NEMO-An Intent Oriented Network Programming Language. <i>Xia Yinben</i>
17:30–18:00	Policy-Compliant Path Diversity and Bisection Bandwidth. <i>Xenofontas Dimitropoulos</i>
18:00–18:40	Beerageous opinion

2.11 10th Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems Workshop (ICOOOLPS)

The ICOOOLPS workshop series brings together researchers and practitioners working in the field of OO languages implementation and optimization. ICOOOLPS key goal is to identify current and emerging issues relating to the efficient implementation, compilation and optimization of such languages, and outlining future challenges and research directions.

Committee

Floréal Morandat, <i>Enseirb-Matmeca</i> (Chair)	Gaël Thomas, <i>Telecom SudParis</i>
Carl Friedrich Bolz, <i>King's College London</i>	Laurence Tratt, <i>King's College London</i>
Eric Jul, <i>Alcatel-Lucent Bell Labs</i>	Jan Vitek, <i>Northeastern Uni</i>
Tobias Pape, <i>Hasso Plattner Institute</i>	Mario Wolczko, <i>Oracle Labs</i>
Jean Privat, <i>Uni du Québec à Montréal</i>	Olivier Zendra, <i>INRIA</i>
Jeremy Singer, <i>Uni of Glasgow</i>	

Program: Monday 6th of July

09:00–09:45	Invited Speaker: Data-centric Metaprogramming in Object-Oriented Languages. <i>Vlad Ureche</i>
	Session 1: Debugging
10:00–10:30	Position Paper: Building Debuggers and Other Tools: We Can “Have it All”. <i>Michael Van De Vanter</i>
10:30–11:00	Trace-based Debugging for Advanced-Dispatching Programming Languages. <i>Christoph Bockisch, Marnix van't Riet, Haihan Yin, Mehmet Aksit, Ziyi Lin, Yuting Chen, and Jianjun Zhao</i>
	Session 2: Method Call
11:00–11:30	Preexistence revisited. <i>Roland Ducournau, Julien Pagès, Jean Privat, and Colin Vidal</i>
11:30–12:00	Short Paper: Call-target-specific Method Arguments. <i>Fabio Niephaus, Matthias Springer, Tim Felgentreff, Tobias Pape, and Robert Hirschfeld</i>
	Session 3: Garbage Collection
13:30–14:00	Ownership and Reference Counting based Garbage Collection in the Actor World. <i>Sylvan Clebsch, Sebastian Blessing, Juliana Franco, and Sophia Drossopoulou</i>
14:00–14:30	METIS: A Smart Memory Allocator Using Historical Reclamation Information. <i>Shijie Xu, Qi Guo, Gerhard Dueck, David Bremner, and Yang Wang</i>
	Session 4: Inside the Virtual Machine
14:30–15:00	Towards Reducing the Need for Algorithmic Primitives in Dynamic Language VMs Through a Tracing JIT. <i>Tim Felgentreff, Tobias Pape, Lars Wassermann, Robert Hirschfeld, and Carl Friedrich Bolz</i>
15:00–15:30	Techniques and Applications for Guest-Language Safepoints. <i>Benoit Dalozé, Chris Seaton, Daniele Bonetta, and Hanspeter Mössenböck</i>
	Session 5: Wrap-up
16:00–16:30	Open discussion
16:30–17:00	Wrap-up and future editions

2.12 STOP 2015: International Workshop on Scripts to Programs

Scripting languages are lightweight, dynamic programming languages designed to maximize productivity. The rising popularity of scripting languages has many underlying causes: they permit interactive experimentation, support reflection and metaprogramming, and admit quick interim solutions that can later be revised. The concept of gradual typing has been proposed, in which the programmer controls which portions of the program are dynamically typed and which portions are statically typed. Over the last decade there has been significant progress on the theory and practice of gradual typing, but there are still many open questions and unexplored points in the design space.

Committee

Jeremy G. Siek, *Indiana Uni* (Chair)
 Amal Ahmed, *Northeastern Uni*
 Avik Chaudhuri, *Facebook*
 Ravi Chugh, *Uni of Chicago*

Ronald Garcia, *Uni of British Columbia*
 James Noble, *Victoria Uni of Wellington*
 Ilya Sergey, *IMDEA Software Institute*
 Eric Tanter, *Uni of Chile*
 Tobias Wrigstad, *Uppsala Uni*

Program: Monday 6th of July

10:05–10:50	MyPy, an Optionally Typed Python. <i>Jukka Lehtosalo</i>
10:50–11:35	Experiments with Strengthening JavaScript. <i>Andreas Rossberg</i>
11:35–12:20	Flow: a static type checker for JavaScript. <i>Avik Chaudhuri</i>
13:50–14:35	JavaScript in the Small. <i>Satish Chandra</i>
Session: Contracts	
14:40–15:00	Statically Checking Effect Contracts in JavaScript. <i>Christopher Schuster and Cormac Flanagan</i>
15:00–15:20	Adding Practical Dependent Types to Typed Racket. <i>Andrew Kent and Sam Tobin-Hochstadt</i>
15:20–15:40	Contracts for Async Patterns in JavaScript. <i>Tim Disney and Cormac Flanagan</i>
Session: Gradual Typing	
16:10–16:30	Position Paper: Performance Evaluation for Gradual Typing. <i>Asumu Takikawa, Daniel Feltey, Ben Greenman, Max New, Jan Vitek, Matthias Felleisen</i>
16:30–16:50	Towards Absolutely Efficient Gradually Typed Languages. <i>Andre Kuhlenschmidt, Deyaaeldeen Almahallawi, and Jeremy G. Siek</i>
16:50–17:10	Towards a Static Type Checker for Python. <i>Francisco Ortin, Baltasar García Perez-Schofield, and José Manuel Redondo</i>
Session: Felleisen v Bracha	
17:10–19:00	Debate: Types for an Untyped World. <i>Matthias Felleisen, Gilad Bracha, and Jan Vitek</i>

2.13 PLAS 2015: ACM SIGPLAN 10th Workshop on Programming Languages and Analysis for Security

PLAS aims to provide a forum for exploring and evaluating ideas on the use of programming language and program analysis techniques to improve the security of software systems. Strongly encouraged are proposals of new, speculative ideas, evaluations of new or known techniques in practical settings, and discussions of emerging threats and important problems.

For details of previous workshops, visit <http://www.cs.cornell.edu/conferences/plas2015/>.

Committee

Limin Jia,	<i>CMU (Chair)</i>	John C. Mitchell,	<i>Stanford</i>
Michael Clarkson,	<i>Cornell Uni (Chair)</i>	Toby Murray,	<i>NICTA and UNSW</i>
Stephen Chong,	<i>Harvard</i>	Benjamin C. Pierce,	<i>Uni of Pennsylvania</i>
Christian Hammer,	<i>Saarland Uni</i>	Frank Piessens,	<i>KU Leuven</i>
Matthew A. Hammer,	<i>Uni Maryland, College Park</i>	Marco Pistoia,	<i>IBM Research</i>
Matteo Maffei,	<i>Saarland Uni</i>	Tamara Rezk,	<i>INRIA</i>
Stephen McCamant,	<i>Uni of Minnesota</i>	Tachio Terauchi,	<i>JAIST</i>

Program: Monday 6th of July

10:05–10:10	Welcome. <i>Michael Clarkson and Limin Jia</i>
	Session 1: Invited Talk I
10:10–11:10	Micro-Policies: Formally Verified, Tag-Based Security Monitors. <i>Cătălin Hrițcu (INRIA)</i>
	Session 2: Technical Talks I
11:10–11:40	Swapsies on the Internet—Towards Reasoning about Risk and Trust in the Open World. <i>Sophia Drossopoulou, James Noble, Mark S. Miller</i>
11:40–12:10	Memory-safe Execution of C on a Java VM. <i>Matthias Grimmer, Roland Schatz, Chris Seaton, Thomas Würthinger, Hanspeter Mössenböck</i>
12:10–12:40	Dynamic Enforcement of Dynamic Policies. <i>Pablo Buiras, Bart van Delft</i>
	Session 3: Invited Talk II
13:50–14:50	Tierless Programming and Reasoning for Networks. <i>Shriram Krishnamurthi</i>
	Session 4: Technical Talks II
14:50–15:10	Short Paper: On High-Assurance Information-Flow-Secure Programming Languages. <i>Toby Murray</i>
15:10–15:30	Short Paper: The Meaning of Attack-Resistant Systems. <i>Vijay Ganesh, Sebastian Banescu, Martín Ochoa</i>
16:10–16:30	Discussion: The future of PLAS
16:30–17:00	Informal Social Gathering

2.14 DSLDI 2015

The goal of the DSLDI workshop is to bring together researchers and practitioners interested in sharing ideas on how DSLs should be designed, implemented, supported by tools, and applied in realistic application contexts. We are both interested in discovering how already known domains such as graph processing or machine learning can be best supported by DSLs, but also in exploring new domains that could be targeted by DSLs. More generally, we are interested in building a community that can drive forward the development of modern DSLs.

Committee

Tijs van der Storm,	<i>CWI</i> (Chair)	Bruno Oliveira,	<i>Uni Hong Kong</i>
Sebastian Erdweg,	<i>TU Darmstadt</i> (Chair)	Cyrus Omar,	<i>CMU</i>
Emilie Balland,	<i>INRIA</i>	Richard Paige,	<i>Uni of York</i>
Martin Bravenboer,	<i>LogicBlox</i>	Tony Sloane,	<i>Macquarie Uni</i>
Hassan Chafi,	<i>Oracle Labs</i>	Emma Söderberg,	<i>Google</i>
William Cook,	<i>UT Austin</i>	Emma Tosch,	<i>UMass Amherst</i>
Shriram Krishnamurthi,	<i>Brown Uni</i>	Jurgen Vinju,	<i>CWI</i>
Heather Miller,	<i>EPFL</i>		

Program: Tuesday 7th of July

10:05–10:20	Introduction. <i>Sebastian Erdweg and Tijs van der Storm</i>
	Session 1
10:20–10:50	SCROLL—A Scala-based library for Roles at Runtime. <i>Max Leuthäuser</i>
10:50–11:20	A case for Rebel, a DSL for product specifications. <i>Jouke Stoel</i>
	Session 2
11:30–12:00	Flick: A DSL for middleboxes. <i>Nik Sultana</i>
12:00–12:30	Towards a Next-Generation Parallel Particle-Mesh Language. <i>Sven Karol, Pietro Incardona, Yaser Afshar, Ivo Sbalzarini, and Jeronimo Castrillon</i>
	Session 3
13:30–14:00	DSLs for Graph Algorithms and Graph Pattern Matching. <i>Oskar van Rest, Sungpack Hong, and Hassan Chafi</i>
14:00–14:30	DSLs of Mathematics, Theorems and Translations. <i>Cezar Ionescu and Patrik Jansson</i>
	Session 4
14:40–15:10	Check Syntax: An Out-of-the-Box Tool for Macro-Based DSLs. <i>Spencer Florence, Ryan Culpepper, Matthew Flatt, and Robby Findler</i>
15:10–15:40	Dynamic Compilation of DSLs. <i>Vojin Jovanovic</i>
	Session 5
16:10–16:40	A practical theory of language-integrated query—and—Everything old is new again. <i>Philip Wadler</i>
16:40–17:40	Discussion

2.15 FTfJP 2015: 17th Workshop on Formal Techniques for Java-like Programs

Work on formal techniques and tools for programs and work on the formal underpinnings of programming languages themselves naturally complement each other. This workshop aims to bring together people working in both these fields, on topics such as language semantics; specification techniques and languages; verification of program properties; verification logics; dynamic program analysis; static program analysis; type systems; challenge problems and solutions; and security.

For details of previous workshops, visit <http://www.cs.ru.nl/ftfjp/>.

Committee

Rosemary Monahan,	<i>Maynooth Uni (Chair)</i>	Gary T. Leavens,	<i>Uni of Central Florida</i>
Dave Clarke,	<i>Uppsala Uni and KU Leuven</i>	K. Rustan M. Leino,	<i>Microsoft Research</i>
Werner Dietl,	<i>Uni of Waterloo</i>	Peter Müller,	<i>ETH Zurich</i>
Sophia Drossopoulou,	<i>Imperial College London</i>	Ruzica Piskac,	<i>Yale</i>
Jean-Christophe Filliatre,	<i>CNRS</i>	Nadia Polikarpova,	<i>MIT CSAIL</i>
Reiner Hähnle,	<i>TU Darmstadt</i>	Erik Poll,	<i>Radboud Uni Nijmegen</i>
Truong Anh Hoang,	<i>Vietnam National Uni</i>	Alexander J. Summers,	<i>ETH Zurich</i>
Marieke Huisman,	<i>Uni of Twente</i>	Elena Zucca,	<i>Uni of Genova</i>
Atsushi Igarashi,	<i>Kyoto Uni</i>		

Program: Tuesday 7th of July

10:05—10:15	Opening. <i>Rosemary Monahan</i>
	Session 1
10:15—10:45	A three-valued type system for true positives detection in Java-like languages. <i>Davide Ancona and Federico Frassetto</i>
10:45—11:15	Imperative Objects with Dependent Types. <i>Joana Campos and Vasco Vasconcelos</i>
11:15—11:45	Foo: A minimal Modern OO Calculus. <i>Prodromos Gerakios, George Fourtounis and Yannis Smaragdakis</i>
11:45—12:15	Delegation vs Inheritance for Typestate Analysis. <i>Du Li, Alex Potanin and Jonathan Aldrich</i>
12:15—12:35	Tool Demo: Why3. <i>Léon Gondelman</i>
	Session 2
13:45—14:15	Provably Live Exception Handling. <i>Bart Jacobs</i>
14:15—14:45	Run-time Assertion Checking of JML Annotations in Multithreaded Applications with e-OpenJML. <i>Jorne Kandziora, Marieke Huisman, Christoph Bockisch and Marina Zaharieva-Stojanovski</i>
14:45—15:35	Invited Talk: Verification of Concurrent Software: Java and OpenCL. <i>Marieke Huisman</i>
	Session 3
16:00—16:30	Automatic Verification of Dafny Programs with Traits. <i>Reza Ahmadi, K. Rustan M. Leino and Jyrki Nummenmaa</i>
16:30—17:00	Conditional Effects in Fine-grained Region Logic. <i>Yuyan Bao, Gary Leavens and Gidon Ernst</i>
17:00—17:30	Regression Verification for Java Using a Secure Information Flow Calculus . <i>Bernhard Beckert, Vladimir Klebanov and Mattias Ulbrich</i>
17:30—17:50	Tool Demo: Viper (Verification Infrastructure for Permission-based Reasoning). <i>Malte Schwerhoff</i>
17:50—18:00	Closing

2.16 ML4PL 2015

Welcome to ML4PL, the first workshop on machine learning techniques applied to programming language-related applications. This workshop puts an emphasis on identifying open problem rather than presenting solution, and encourages discussion amongst the participants. Attendance will be limited to ensure that meeting retains an interactive character.

Committee

Prasad Kulkarni,	<i>Uni Kansas</i> (Chair)	Suresh Jagannathan,	<i>DARPA</i>
Jan Vitek,	<i>Northeastern Uni</i> (Chair)	Pavel Kordik,	<i>Czech Techn. Uni in Prague</i>
Molham Aref,	<i>LogicBlox</i>	Aditya Nori,	<i>MSR, UK</i>
Emery Berger,	<i>UMass Amherst</i>	Martin Vechev,	<i>ETH Zurich</i>
Andrew D. Gordon,	<i>MSR and Uni Edinburgh</i>	Zheng Wang,	<i>Lancaster Uni</i>
Kathleen Fisher,	<i>Tufts Uni</i>	Eran Yahav,	<i>Technion</i>
Björn Franke,	<i>Uni Edinburgh</i>		

Program: Tuesday 7th of July

10:00–10:15	Welcome and introductions
10:15–11:00	Machine Learning for Programming. <i>Martin Vechev</i>
	Session 1
11:00–11:30	Problems and opportunities—ML for program verification. <i>Aditya Nori</i>
11:30–12:00	Machine Learning for Type-Based Inference and Verification. <i>Suresh Jagannathan</i>
12:00–12:30	Learning to Decipher the Heap. <i>Marc Brockschmidt, Byron Cook, Pushmeet Kohli, and Daniel Tarlow</i>
	Session 2
13:45–14:15	Problems and opportunities—Program similarity. <i>Eran Yahav</i>
14:15–14:45	Inferring Coding Conventions with Machine Learning. <i>Miltiadis Allamanis, Earl Barr, Christian Bird, and Charles Sutton</i>
14:45–15:15	Using topic models to understand programming languages literature. <i>Kathleen Fisher</i>
15:15–15:45	Scaling Program Synthesis by Exploiting Existing Code. <i>James Bornholt and Emina Torlak</i>
	Session 3
16:10–16:40	Problems and opportunities—Statistical modeling in (declarative) PLs. <i>Molham Aref</i>
16:40–17:10	Bimodal Modelling of Source Code and Natural Language. <i>Andrew D. Gordon</i>
17:10–17:40	Machine learning for predictive modeling and recommender systems automation. <i>Pavel Kordik</i>
17:40–18:10	Man vs. Machine: Challenges of Integrating Programming Languages and People. <i>Emery Berger</i>

2.17 PLE 2015 Programming Language Evolution

Programming languages tend to evolve in response to user needs, hardware advances, and research developments. Evolving programming languages is however challenging at various levels. For example, impact on developers can be negative, evaluating a proposed language change is difficult, and new language features may interact badly with existing features. This workshop brings together researchers and developers to tackle the important challenges faced by programming language evolution, to share new ideas and insights, to reflect on past experience, and to advance programming language design.

Committee

Alan Mycroft,	<i>Uni of Cambridge</i> (Chair)	Sarah Mount,	<i>Uni of Wolverhampton</i>
Dominic Orchard,	<i>Imperial College London</i> (Chair)	Jeffrey Overbey,	<i>Auburn Uni</i>
Raoul-Gabriel Urma,	<i>Uni of Cambridge</i> (Chair)	Max Schaefer,	<i>Semmler</i>
Heather Miller,	<i>EPFL</i>		

Program: Tuesday 7th of July

10:05–10:20	Introduction. <i>Dominic Orchard</i>
	Session 1
10:20–10:45	On the origin of the block concept. <i>Gauthier van den Hove</i>
10:45–11:10	Retrofitting static typing to Python. <i>Jukka Lehtosalo</i>
11:10–11:40	Discussion
	Session 2
11:40–12:05	Fine grained language composition. <i>Laurence Tratt</i>
12:05–12:30	The Good, the Bad, and the Ugly: An Empirical Study of Implicit Type Conversions in JavaScript. <i>Michael Pradel</i>
	Session 3
13:50–14:40	Keynote: What—if anything—have we learned from C++? (<i>joint with CurryOn</i>). <i>Bjarne Stroustrup</i>

2.18 Truffle 2015

The goal of this full day workshop is to attract programming language developers interested in using Truffle and Graal for creating programming language implementations and tools, as well as, more broadly, developers interested in discussing language implementation approaches heavily relying on dynamic profiling feedback and specialization. The workshop is meant to be a forum where language developers can learn about Truffle and Graal, share their experience using the toolkit, identify potential limitations and discuss methods of rectifying them, as well as propose future directions for the development of Truffle languages tooling support and of the toolkit itself. We are especially interested in attracting participation of language developers that are not yet familiar with Truffle or Graal but are interested in exploring how they can simplify development of their own current or future projects.

Committee

Adam Welc, *Oracle Labs*
Mario Wolczko, *Oracle Labs*

Thomas Wuerthinger, *Oracle Labs*

Program: Tuesday 7th of July

10:05–10:20	Introduction. <i>Thomas Wuerthinger</i>
10:20–10:45	One VM to Rule Them All And With Interoperability Bind Them. <i>Matthias Grimmer</i>
10:45–11:10	Substrate VM. <i>Paul Wögerer</i>
11:10–11:35	Integrating Truffle into an Existing Language Implementation. <i>Chris Seaton</i>
11:35–12:00	Developer Tools for Truffle-implemented Languages. <i>Michael Van De Vanter</i>
12:00–12:25	Truffle Trade-offs, Concepts and Experiences. <i>Christian Humer</i>
12:25–12:35	Additional discussion
13:50–14:15	Complementary Directions for Truffle Languages and Liballocs. <i>Stephen Kell</i>
14:15–14:40	Accurate Bytecode-level Profiling of Dynamically Optimized Code with Graal. <i>Yudi Zheng</i>
14:40–15:05	Product Lines of Interpreters Using Truffle with Object Algebras. <i>Yanlin Wang</i>
15:05–15:30	Heterogenous Data Structures for the Masses. <i>Michael Steindorfer</i>
15:30–15:40	Additional discussion
16:10–16:35	Which Meta-Compilation Approach is Better for Self-Optimizing Interpreters: Tracing or Partial Evaluation?. <i>Stefan Marr</i>
16:35–17:00	Enabling Heterogenous Computing in Java with Graal. <i>Juan Fumero</i>
17:00–17:25	Project Beehive: A Hardware/Software Co-designed Stack for Runtime and Architectural Research. <i>Christos Kotselidis</i>
17:25–17:50	Truffle-LMS: Combining Self-modifying Interpreters and Generative Meta-programming. <i>Tiark Rompf</i>
17:50–18:15	SQPyte: JITing a DBMS. <i>Carl Friedrich Bolz, Darya Kurilova, and Laurence Tratt</i>
18:15–18:30	Additional discussion

2.19 Invited Speakers

Jade Alglave (University College London) Lecturer at University College London. Her area of expertise is weak memory models and the interface between hardware and programming languages. She actually understands what C++ programs do when run on a PPC!

Herding Weak Memory Cats in Moravia, Friday, July 10th, 17:00.

Molham Aref (LogicBlox) Founder of LogicBlox and Predictix. He has over 23 years of experience leading teams that deliver high-value predictive and prescriptive analytics solutions to some of the world's largest enterprises.

Declarative Probabilistic Programming in Moravia, Thursday, July 9th, 10:30.

Hitesh Ballani (Microsoft Research, UK) Researcher in the Systems and Networking group in Cambridge. He graduated from Cornell University where he indulged in follies like Scalable Internet Routing and Network Management. His current research centers around systems and networking issues in data centers.

Enabling End Host Network Functions in Hluboka II, Monday, July 6th, 09:10.

Jeff Bezanson (Julia Computing) Before the Julia effort began, Jeff worked as a software engineer at Interactive Supercomputing, which developed the Star-P parallel extension to MATLAB. Jeff was a principal developer of M#, an implementation of the MATLAB language on .NET.

Julia in Curry I, Monday, July 6th, 17:50.

Ary Borenszweig (Manas) Software craftsman and guru in all code-related problems. Always teaching how to develop the best code with simplicity in mind, he makes everything seem easy no matter how difficult, to the point that nothing seems impossible with him on the team.

Crystal: a programming language for humans and computers in Curry II, Monday, July 6th, 11:45.

Gilad Bracha (Google) Creator of the Newspeak language and a software engineer at Google where he works on Dart. He is co-author of the Java Language Specification, and a researcher in the area of object-oriented programming languages. Prior to joining Sun, he worked on Strongtalk, the Animorphic Smalltalk System.

Combining Inheritance and Modularity in MASPEGHI, Sunday, July 5th, 13:30.

Programs Wanted: Dead or Alive in Curry I, Monday, July 6th, 09:00.

Types for an Untyped World in Moravia III, Monday, July 6th, 17:10.

Andrey Breslav (JetBrains) Lead language designer for Kotlin, Andrey began his career at Borland, where he worked on language implementations for MDA support. He represented JetBrains in the JCP expert group for JSR 335, "Project Lambda".

Kotlin: Challenges in language design in Curry I, Monday, July 6th, 10:05.

Kim Bruce (Pomona College) Professor at Pomona College. His research focus evolved from models of the polymorphic lambda calculus to the study of semantics and type theory. This led to his work in the design of object-oriented languages.

Graceful Programming — Teaching Introductory Programming in Moravia, Thursday, July 9th, 15:30.

Antonio Capone (Politecnico di Milano) Professor at Polimi and co-founder and CTO of MobiMESH. His expertise is on networking, protocol design and performance evaluation of wireless access and multi-hop networks, traffic management and quality of service issues in IP networks, and network planning and optimization.

From Dumb to Smarter Switches in Software Defined Networks: Towards a Stateful Data Plane in Hluboka II, Monday, July 6th, 15:00.

Pavol Černý (University of Colorado Boulder) He works on computer-aided verification and programming languages, program synthesis, and algorithmic and logical foundations for reliable software.

Program Synthesis for Network Updates in Hluboka II, Monday, July 6th, 11:50.

Jong-Deok Choi (Samsung Research America) Executive Vice President at Samsung Electronics managing its Software R&D Center. He led the development of the Tizen Open Source Platform for smart devices. He is a Co-Chair of the Tizen's Technical Steering Group.

Programming in the Large for the Internet of Things in Bohemia, Wednesday, July 8th, 09:00.

Cliff Click (0xdata) CTO and Co-Founder of H₂O, the opensource math and machine learning engine for Big Data. Cliff wrote his first compiler when he was 15 (Pascal to TRS Z-80!), although his most famous compiler is the HotSpot Server Compiler (the Sea of Nodes IR). Cliff helped Azul Systems build an 864 core pure-Java mainframe that keeps GC pauses on 500Gb heaps to under 10ms.

Bits of Advice for VM Writers in Curry I, Tuesday, July 7th, 10:05.

Xenofontas Dimitropoulos (University of Crete and FORTH) Leads the Internet Research Group doing research on software defined networks and Internet measurements.

Policy-Compliant Path Diversity and Bisection Bandwidth in Hluboka II, Monday, July 6th, 17:30.

Colin Dixon (Brocade) Chair of OpenDaylight's Technical Steering Committee and a Principal Engineer at Brocade. He focuses on leveraging open source software to build innovative networking solutions. He is also a researcher specializing in distributed systems, networking, operating systems and security.

The State of (Open Source) SDN and Programming Languages Opportunities in Hluboka II, Monday, July 6th, 14:00.

Anton Epple (Netbeans) Consultant worldwide for a wide variety of companies, ranging from startups to Fortune 500 companies. His main interest is Client side development, and he has authored books and numerous articles on this topic. He is a member of the NetBeans Dream Team and an Oracle Java Champion.

Java everywhere again - with DukeScript! in Curry II, Tuesday, July 7th, 17:00.

Philippa Gardner (Imperial College London) Professor at Imperial. Her research focuses on program verification: in particular, reasoning about web programs and reasoning about concurrent programs.

Modern Concurrent Separation Logics in Moravia, Wednesday, July 8th, 17:00.

Andrew Gordon (Microsoft Research and University of Edinburgh) Principal Researcher at Microsoft Research. My current research is on probabilistic programming for machine learning.

Empowering Spreadsheet Users with Probabilistic Programs in Curry I, Monday, July 6th, 13:50.

Bimodal Modelling of Source Code and Natural Language in ML4PL, Tuesday, July 7th, 16:40.

Robert Grimm (Goldman Sachs) My research explores how to leverage programming language technologies to make complex systems easier to build, maintain, and extend.

Nimble yet systematic trading in Moravia, Wednesday, July 8th, 15:30.

Sumit Gulwani (Microsoft Research) Principal researcher at Microsoft Research. His research interests lie in the cross-disciplinary areas of automating end-user programming and building intelligent tutoring systems.

Data Manipulation using Programming By Examples and Natural Language in Curry I, Tuesday, July 7th, 10:55.

Cătălin Hrițcu (INRIA Paris) Researcher at INRIA where he develops rigorous formal techniques for solving security problems. He is particularly interested in: formal methods for computer and network security, programming languages, and the design and verification of security-critical systems.

Micro-Policies: Formally Verified, Tag-Based Security Monitors in PLAS, Monday, July 6th, 10:10.

Marieke Huisman (University of Twente, Netherlands) Adjoint Professor at the University of Twente. In 2010 she received a personal ERC Starting Grant for the VerCors project on Verification of Concurrent Data Structures. She has also been involved in several EU projects, such as CARP, Mobius, VerifiCard.

Verification of Concurrent Software: Java and OpenCL in FTfJP, Tuesday, July 7th, 14:45.

Suresh Jagannathan (DARPA) At DARPA since 2013, Suresh's research interests include programming languages, compilers, program verification, and concurrent and distributed systems.

Machine Learning for Type-Based Inference and Verification in ML4PL, July 7th, 11:30.

Taming Uncertainty, Scale, and Change: A Programming Language Perspective in Moravia, Friday, July 10th, 10:30.

Stefan Karpinski (Julia Computing) I'm a data scientist and applied mathematician. I've previously worked at Akamai, Citrix Online, and Etsy. Currently, I'm working on a next-generation programming language for numerical and scientific computing, called Julia.

Julia in Curry I, Monday, July 6th, 17:50.

Shriram Krishnamurthi (Brown University) Professor of Computer Science at Brown University. Hacker, scientist, teacher.

The story of Arjun Guha, or: The arc of a research project in Doctoral Symposium, July 5th, 13:30.

Tierless Abstractions for Programming Devices in Hluboka II, July 6th, 10:10.

Tierless Programming and Reasoning for Networks in PLAS, Monday July 6th, 13:50.

The evolving design of Pyrets in Moravia, Wednesday, July 8th, 13:30.

Julia Lawall (INRIA LIP6) Senior Researcher at Inria. Her research interests are in the area of improving the quality of infrastructure software, using a variety of approaches including program analysis, program transformation, and the design of domain-specific languages.

Coccinelle for the Working Programmer in Curry I, Monday, July 6th, 10:55.

Anil Madhavapeddy (University of Cambridge) Lecturer at the University of Cambridge. He was on the original team that developed the Xen hypervisor, and helped develop an industry-leading cloud management toolstack written in OCaml. He founded and directs the OCaml Labs group and leads the MirageOS unikernel project.

Immutable Distributed Infrastructure with Unikernels in Hluboka II, Monday, July 6th, 11:10.

Tyler McMullen (Fastly) CTO of Fastly, where he's responsible for the system architecture and leads the company's technology vision. A self-described technology curmudgeon, he has experience in everything from web design to kernel development, and loathes all of it. Especially distributed systems.

It Probably Works in Curry I, Monday, July 6th, 14:40.

Bertrand Meyer (ETH Zurich, Innopolis University and Eiffel Software) Professor of Software Engineering at ETH Zurich and at ITMO, Saint Petersburg. He is the initial designer of the Eiffel method and language and has continued to participate in its evolution.

The Prague List: Open Problems in Object-Oriented Programming in Bohemia, Friday, July 10th, 13:30.

Filip Pizlo (Apple) Works on compilers and language runtime systems at Apple. These days most of his work involves WebKit's excellent JavaScript engine mostly on the optimizing JIT compiler and its dynamic type inference engine, but also the interpreter, object model, and garbage collector. He also sometimes commits things to LLVM.

LLVM for JavaScript in Curry I, Tuesday, July 7th, 14:40.

Tony Printezis (Twitter) Specialties: 15+ years of virtual machine implementation experience with special focus on memory management / garbage collection. Close to 20 years of C/C++ experience. 15+ years of Java experience. Expert in concurrent/parallel programming.

Java @ Twitter - a bird's eye view in Curry I, Monday, July 6th, 16:10.

Mooly Sagiv (Tel Aviv University) Member of the Advisory Board at Panaya Inc and Senior Member of staff at Tel-Aviv University. His fields of interests include Programming Languages, Compilers, Abstract interpretation, Profiling, Pointer Analysis, Shape Analysis, Inter-procedural dataflow analysis, Program Slicing, Language-based programming environments.

Verifying Correctness of Statefull Networks in Moravia, Wednesday, July 8th, 10:30.

Marc Shapiro (Inria & LIP6) Researches distributed computer systems, data replication and consistency algorithms, and distributed garbage collection. He invented the proxy concept, which is now universal on the Internet.

Encapsulating replication, high concurrency and consistency with CRDTs in Curry I, Monday, July 6th, 11:45.

Bjarne Stroustrup (Morgan Stanley) Designer and implementer of C++.

What, if anything, have we learned from C++? in Moravia, Tuesday, July 7th, 14:30.

Object-Oriented Programming without Inheritance in Bohemia, Wednesday, July 8th, 10:30.

Emina Torlak (University of Washington) Assistant Professor at the University of Washington, working at the intersection of software engineering, programming languages, and formal methods. Her research focuses on developing automated tools that help people build better software more easily.

Synthesis and Verification for Everyone in Moravia, Thursday, July 9th, 13:30.

Tomas Vojnar (Brno University of Technology) Worked on efficient ways of dealing with various kinds of non-deterministic automata and logics, with applications in the above mentioned verification approaches. As for concurrent programs, he mainly works in the area of their noise-based testing and dynamic analysis, self-healing, as well as applications of search techniques in noise-based testing.

Noise-based testing of Concurrent Software in Moravia, Thursday, July 9th, 17:00.

Philip Wadler (University of Edinburgh) ACM Fellow and a Fellow of the Royal Society, past chair of SIGPLAN, and a winner of the POPL Most Influential Paper. He contributed to the designs of Haskell, Java, and XQuery.

Everything old is new again: Quoted domain specific languages in Curry II, Tuesday, July 7th, 14:40.

A practical theory of language-integrated query -and- Everything old is new again in Karlstejn, Tuesday, July 7th, 16:10.

Juan Wajnerman (Manas) Juan is the one you turn to when there is a question that no one else can answer: whether Internet is down or the architecture for a massive system needs to be designed, Juan has the solution. Wise and humble, he applies his experience with patience and pragmatism.

Crystal: a programming language for humans and computers in Curry II, Monday, July 6th, 11:45.

Roger Wattenhofer (ETHZ) Professor at the ETHZ. His research interests are a variety of algorithmic and systems aspects in computer science and information technology, currently in particular wireless networks, wide area networks, mobile systems, social networks, and physical algorithms. He publishes in different communities: distributed computing, networking, or theory.

Managing Dynamic Networks: Distributed or Centralized Control? in Hluboka II, Monday, July 6th, 16:00.

Eran Yahav (Technion) Professor at the Technion, Eran works on program analysis, program synthesis and program verification.

Program similarity in Moravia III, Tuesday July 7, 13:45.

Programming with Millions of Examples in Moravia, Friday, July 10th, 15:00.

Xia Yinben (Huawei) Senior staff researcher within Huawei's network research department from 2008. He dedicated his research effort in SDN and network programming language (NEMO Project).

NEMO - An Intent Oriented Network Programming Language in Hluboka II, Monday, July 6th, 17:00.

3 | Reference

3.1 Schedule Overview

Sunday (July 5)

Event	Room	Explanation
COP	FIT-107	Context-Oriented Programming
PhD Symposium	FIT-302	ECOOP 2015 Doctoral Symposium
MASPEGHI	FIT-364	Mechanisms for Specialization, Generalization and Inheritance
Pharo	FIT-301	Pharo Coding Sprint
RIOT	FIT-111	R Implementation, Optimization and Tooling

Monday (July 6)

Event	Room	Explanation
Curry I	CurryI	Curry On Track I
Curry II	CurryII	Curry On Track II
ICOOOLPS	Bohemia III	Implem., Compil., Optim. of OO Languages, Programs & Systems
JSTools	Hluboka I	Tools for JavaScript Analysis
NetPL	Hluboka II	Networking meets Programming Languages
PLAS	Karlstejn	Programming Languages and Analysis for Security
STOP	Moravia III	Scripts to Programs

Tuesday (July 7)

Event	Room	Explanation
Curry I	CurryI	Curry On Track I
Curry II	CurryII	Curry On Track II
DSLDI	Karlstejn	Domain-Specific Language Design and Implementation
FTfJP	Hluboka I	Formal Techniques for Java-like Programs
ML4PL	Moravia III	Machine Learning for Programming Languages
PLE	Hluboka II	Programming Language Evolution
Truffle	Bohemia III	Truffle/Graal Languages

Wednesday (July 8)–Friday (July 10)

Event	Room	Explanation
Research Track	Bohemia	ECOOP Research Track
Summer School	Moravia	ECOOP Summer School

3.2 ECOOP / CurryOn Program

Program: Monday, July 6

8:45–9:00	Curry On opening remarks. <i>Jan Vitek and Heather Miller</i>
9:00–9:45	Keynote: Gilad Bracha
9:45–10:05	Coffee break
10:55–11:35	Kotlin: Challenges in language design. <i>Andrey Breslav, JetBrains</i>
(Room 2)	Crystal: a programming language for humans and computers. <i>Ary Borenszweig & Juan Wajnerman, Manas</i>
10:55–11:35	Structured Synchronous Programming. <i>Francisco Sant’Anna, PUC-Rio, Brazil</i>
(Room 2)	Akka Typed: Between Session Types and the Actor Model. <i>Roland Kuhn, Typesafe</i>
11:45–12:25	Encapsulating replication, high concurrency and consistency with CRDTs. <i>Marc Shapiro INRIA/LIP6</i>
(Room 2)	Pull > Push: Please stop polluting our imperative languages with pure concepts. <i>Ron Pressler, Parallel Universe</i>
12:35–13:50	Lunch
13:50–14:30	Empowering Spreadsheet Users with Probabilistic Programs. <i>Andy Gordon, Microsoft Research</i>
(Room 2)	Speed at a Price: The Evolution of V8 and the Challenges of Research in a Billion User VM. <i>Ben Titzer, Google</i>
14:40–15:20	It Probably Works. <i>Tyler McMullen, Fastly</i>
(Room 2)	Coccinelle for the Working Programmer. <i>Julia Lawall, Inria/LIP6/UPMC/Sorbonne University</i>
15:30–16:00	Coffee break
16:00–16:40	Java@Twitter—a bird’s eye view. <i>Tony Printezis, Twitter</i>
(Room 2)	Servant: a type-level DSL for web APIs. <i>Julian Arni, Zalora</i>
16:50–17:30	Post-FRP Frontend Programming. <i>Bodil Stokke</i>
(Room 2)	Making Embedded Domain Specific Languages a Practical Reality. <i>Jurriaan Hage, Utrecht University</i>
17:40–18:30	Keynote: Jeff Bezanson & Stefan Karpinski
19:00–22:00	Curry On Party! <i>Velka Klasterni Restaurace</i>

Program: Tuesday, July 7

9:00–9:45	Keynote: Brendan Eich
9:45–10:05	Coffee break
10:05–10:45	Bits of Advice for VM Writers. <i>Cliff Click, H₂O</i>
(Room 2)	Some of the Things That Macros Do. <i>Zach Tellman, Factual</i>
10:55–11:35	Pyro—Typing Python in Python. <i>Alex Muscar, Lyst</i>
(Room 2)	QuickCheck: from invention to product. <i>Thomas Arts, Quviq</i>
11:45–12:25	How to be a good host: miniKanren as a case study. <i>Dan Friedman & Jason Hemann, Indiana University</i>
(Room 2)	Data Manipulation using Programming By Examples and Natural Language. <i>Sumit Gulwani, Microsoft Research</i>
12:35–13:50	Lunch
13:50–14:30	What—if anything—have we learned from C++?. <i>Bjarne Stroustrup</i>
(Room 2)	Coding for Types: The Universe Pattern in Idris. <i>David Christiansen, IT University of Copenhagen</i>
14:40–15:20	Tern: Practical Type Inference for JavaScript Editing. <i>Marijn Haverbeke</i>
(Room 2)	Everything old is new again: Quoted domain specific languages. <i>Philip Wadler, University of Edinburgh</i>
15:30–16:00	Coffee break
16:00–16:40	LLVM for JavaScript. <i>Filip Pizlo, Apple</i>
(Room 2)	Everything old is new again: Quoted domain specific languages. <i>Philip Wadler, University of Edinburgh</i>
(Room 2)	GS Collections: Echoes of Smalltalk’s Past. <i>Alex Iliev, Goldman Sachs</i>
16:50–17:30	Rust: A Type System You Didn’t Know You Wanted. <i>Felix Klock, Mozilla</i>
(Room 2)	Java everywhere again—with DukeScript!. <i>Anton Epple, NetBeans</i>
17:40–18:20	Pony: making it easy to write efficient, concurrent, data-race free programs. <i>Sylvan Clebsch, Sebastian Blessing, Sophia Drossopoulou, Imperial College</i>
(Room 2)	Scaling SQL Databases Beyond Limits with PostgreSQL. <i>Valentin Gogichashvili, Zalando</i>
18:30–19:20	Keynote: Evan Czaplicki

Program: Wednesday, July 8

09:00–10:00	Keynote: Jong-Deok Choi
10:00–10:30	Coffee
	Session 1: Gradual Typing
10:30–11:00	Towards Practical Gradual Typing. <i>Asumu Takikawa, Daniel Feltey, Earl Dean, Robby Findler, Matthew Flatt, Sam Tobin-Hochstadt, and Matthias Felleisen</i>
11:00–11:30	TreatJS: Higher-Order Contracts for JavaScripts. <i>Matthias Keil and Peter Thiemann</i>
11:30–12:00	Trust, but Verify: Two-Phase Typing for Dynamic Languages. <i>Panagiotis Vekris, Benjamin Cosman, and Ranjit Jhala</i>
12:00–13:30	Lunch
	Session 2: Implementation
13:30–14:00	Concrete Types for TypeScript. <i>Gregor Richards, Francesco Zappa Nardelli, and Jan Vitek</i>
14:00–14:30	Simple and Effective Type Check Removal through Lazy Basic Block Versioning. <i>Maxime Chevalier-Boisvert and Marc Feeley</i>
14:30–15:00	Loop tiling in the presence of exceptions. <i>Abhilash Bhandari and V Krishna Nandivada</i>
15:00–15:30	Coffee
	Session 3: Objects
15:30–16:00	A Theory of Tagged Objects. <i>Joseph Lee, Jonathan Aldrich, Troy Shaw, and Alex Potanin</i>
16:00–16:30	Brand Objects for Nominal Typing. <i>Timothy Jones, Michael Homer, and James Noble</i>
16:30–17:00	Transparent Object Proxies in JavaScript. <i>Matthias Keil and Peter Thiemann</i>
17:00–17:30	Coffee
	Session 4: Analysis I
17:30–18:00	Access-rights Analysis in the Presence of Subjects. <i>Paolina Centonze, Marco Pistoia, and Omer Tripp</i>
18:00–18:30	Variability Abstractions: Trading Precision for Speed in Family-Based Analyses. <i>Aleksandar S. Dimovski, Claus Brabrand, and Andrzej Wasowski</i>

Program: Thursday, July 9

09:00–10:00	Keynote: DN Senior , Bjarne Stroustrup
10:00–10:30	Coffee
	Session 5: Developer Assistance
10:30–11:00	Optimization Coaching for JavaScript. <i>Vincent St-Amour, Shu-yu Guo</i>
11:00–11:30	PerfBlower: Quickly Detecting Memory-Related Performance Problems via Amplification. <i>Lu Fang, Liang Dou, and Harry Xu</i>
11:30–12:00	Hybrid DOM-Sensitive Change Impact Analysis for JavaScript. <i>Saba Alimadadi, Ali Mesbah, and Karthik Pattabiraman</i>
12:00–13:30	Lunch
	Session 6: Type Systems
13:30–14:00	Intensional Effect Polymorphism. <i>Yuheng Long, Yu David Liu, and Hridayesh Rajan</i>
14:00–14:30	Type Inference for Place-Oblivious Objects. <i>Riyaz Haque and Jens Palsberg</i>
14:30–15:00	Asynchronous Liquid Separation Types. <i>Johannes Kloos, Rupak Majumdar, and Viktor Vafeiadis</i>
10:00–10:30	Coffee
	Session 7: Parallelism
15:30–16:00	The Eureka Programming Model for Speculative Task Parallelism. <i>Shams Imam and Vivek Sarkar</i>
16:00–16:30	Cooking the Books: Formalizing JMM Implementation Recipes. <i>Gustavo Petri, Jan Vitek, and Suresh Jagannathan</i>
16:30–17:00	Defining Correctness Conditions for Concurrent Objects in Multicore Architectures. <i>Brijesh Dongol and John Derrick, Lindsay Groves, Graeme Smith</i>
	Session 8: Empirical Studies
17:00–17:30	The Good, the Bad, and the Ugly: An Empirical Study of Implicit Type Conversions in JavaScript. <i>Michael Pradel and Koushik Sen</i>
17:30–18:00	The Love/Hate Relationship with the C Preprocessor: An Interview Study. <i>Flavio Medeiros, Christian Kästner, Márcio Ribeiro, Sarah Nadi, and Rohit Gheyi</i>

Program: Friday, July 10

09:00–10:00	Keynote: DN Junior , Alexander J. Summers
10:00–10:30	Coffee
	Session 9: Abstraction
10:30–11:00	A Pattern Calculus for Rule Languages: Expressiveness, Compilation, and Mechanization. <i>Avraham Shinnar, Jerome Simeon, and Martin Hirzel</i>
11:00–11:30	Global Sequence Protocol: A Robust Abstraction for Replicated Shared State. <i>Sebastian Burckhardt, Daan Leijen, Jonathan Protzenko, and Manuel Fähndrich</i>
11:30–12:00	Streams a la carte: Extensible Pipelines with Object Algebras. <i>Aggelos Biboudis, Nick Palladinos, George Fourtounis, and Yannis Smaragdakis</i>
12:00–13:30	Lunch
13:30–14:30	Keynote: Bertrand Meyer
14:30–15:00	Coffee
	Session 10: Verification
15:00–15:30	Lightweight Support for Magic Wands in an Automatic Verifier. <i>Malte Schwerhoff and Alexander J. Summers</i>
15:30–16:00	Modular Verification of Finite Blocking in Non-terminating Programs. <i>Pontus Boström and Peter Müller</i>
16:00–16:30	Modular Termination Verification. <i>Bart Jacobs, Dragan Bosnacki, and Ruurd Kuiper</i>
16:30–17:00	Coffee
	Session 11: Analysis II
17:00–17:30	Framework for Static Analysis of PHP Applications. <i>David Hauzar and Jan Kofroň</i>
17:30–18:00	Adaptive Context-sensitive Analysis for JavaScript. <i>Shiyi Wei and Barbara Ryder</i>
18:00–18:30	Scalable and Precise Static Analysis of JavaScript Applications via Loop-Sensitivity. <i>Changhee Park and Sukyoung Ryu</i>

3.3 Maps

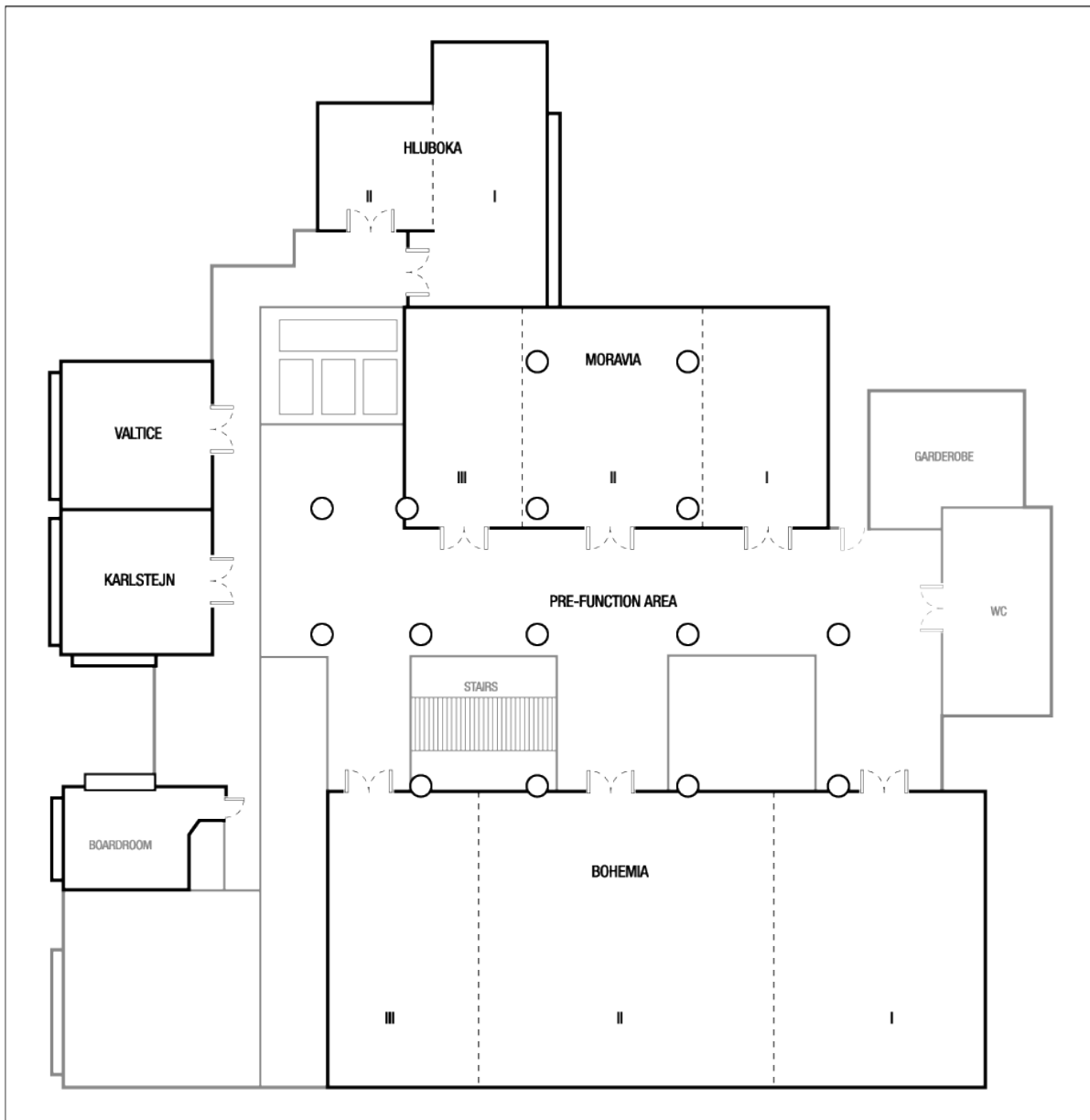


Figure 3.1: Main Conference (Mariott) map



Figure 3.2: Sunday workshops are at FIT, all other events at the Marriott