FUNCTIONAL
CONF
OCT9-11 '14
BANGALORE
ThoughtWorks®
helpshift
DYALOG
nilenso
F#
ERLANG

Thursday, 9 October

Week 41 of 2014

▬ Functional Conf 2014 Important Dates

October 2014

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	**9**	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

November 2014

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Functional Conference 2014 09/10 to 11/10

8 am

8:30 am Registration

9 am

9:00 am Keynote - The Joy of Functional Programming by Venkat Subramaniam
Track 1 (GBR1)

10 am

10:00 am Welcome Address

10:15 am Coffee/Tea Break

10:30 am Functional Reactive UIs with Elm by Shashi Gowda
Track 1(GBR1)

10:30 am Haskell for Everyday Programmers by Venkat Subra-maniam
Track 2(GBR2)

11 am

Noon

12:00 pm Break

12:15 pm Transforming your C# code to Functional Style by Venkat Subramaniam

12:15 pm Applying functional programming principles to large scale data processing by Kishore

1 pm

1:00 pm Lunch

2 pm

2:00 pm Functional Programming in Java by Premanand Chan-drasekaran
Track 1(GBR1)

2:00 pm Compile your own cloud with Mirage OS v2.0 by Thomas Gazagnaire
Track 2(GBR2)

3 pm

3:00 pm Coffee/Tea Break

3:15 pm Property based testing for functional domain models by Debasish Ghosh

3:15 pm Functional Programming in Ruby by Keith Bennett from
Track 1(GBR1)

4 pm

4:00 pm Break

4:15 pm Clojurescript and Om - Pragmatic functional program-ming in the Javascript Land by Vagmi Mudumbai from

4:15 pm Code Jugalbandi by Dhaval Dalal and Ryan Lemmer from
Track 2(GBR2)

5 pm

5:15 pm Break

5:30 pm Discovering Functional Treasure in Idiomatic Groovy by Naresha K

5:30 pm Learning (from) Haskell - An experience report by Aditya Godbole

6 pm

6:15 pm Break

6:30 pm Fish Bowl
Track 1(GBR1)

7 pm

7:30 pm Dinner & Networking

8 pm

9 pm

10 pm

● All-Day Events

09/10 to 11/10 Functional Conference 2014
Location: http://functionalconf.com

- - - - - - - - - - - - - - - -

● Timed Events

8:30 am to 9:00 am Registration
Location:

9:00 am to 10:00 am Keynote - The Joy of Functional Programming by Venkat Subramaniam
Location: Track 1 (GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/420/keynote-the-joy-of-functional-programming
Notes: It's been around for a long time, but everyone's talking about it all of a sudden. But why and why now? We've been

programming in languages like Java for a while, quite well. Now we're asked to change and the languages themselves

are changing towards this style of programming. In this keynote, a passionate polyglot programmer and author of

"Functional Programming in Java: Harnessing the Power of Java 8 Lambda Expressions" will share the reasons

we need to make the paradigm shift and the pure joy—the benefits—we will reap from it.

10:00 am to 10:15 am Welcome Address
Location: Track 1(GBR1)

10:15 am to 10:30 am Coffee/Tea Break
Location:

10:30 am to 12:00 pm Functional Reactive UIs with Elm by Shashi Gowda
Location: Track 1(GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/329/functional-reactive-uis-with-elm
Notes: Elm is a strongly typed functional reactive programming (FRP) language that compiles to HTML, CSS, and Javascript. In Elm, the Signal type represents a time-varying value--things like mouse position, keys pressed, current time are signals. With Signals, one can write terse code that is isomorphic to a dataflow diagram of the app. The code hence feels natural and is 100% callback free. All this, with powerful type inference.

This talk is an introduction to FRP. It explores functionally composing graphics and UIs, and creating interactions and animations with the Signal type. There will also be an overview of Elm's execution mechanism and the time traveling debugger: a consequence of Elm's purely functional approach.

While instructive, it will be good fun too, in the spirit of Elm.

10:30 am to 12:00 pm Haskell for Everyday Programmers by Venkat Subramaniam
Location: Track 2(GBR2)
URL: http://confengine.com/functional-conf-2014/proposal/394/haskell-for-everyday-programmers
Notes: I learn different languages not to make use of them, but to program in my current languages in a better way. As we adapt functional style of programming in mainstream languages, like Java, C#, and C++, we can learn a great deal from a language that is touted as a purely functional language.

Haskell is statically typed, but not in a way like Java, C#, or C++. Its static typing does not get in the way of productivity. Haskell quietly does lazy evaluation and enforces functional purity for greater good. Everyday programmers, like your humble speaker, who predominantly code in mainstream languages, can greatly benefit from learning the idioms and style of this elegant language. The next time we sit down to crank out some code in just about any language, we can make use of some of those styles, within the confines of the languages, and move towards a better, functional style.

Thursday, 9 October

Week 41 of 2014

- **12:00 pm to 12:15 pm Break**
  Location:

- **12:15 pm to 1:00 pm Transforming your C# code to Functional Style by Venkat Subramaniam**
  Location: Track 2(GBR2)
  URL: http://confengine.com/functional-conf-2014/proposal/670/transforming-your-c-code-to-functional-style
  Notes: Since the introduction of lambda expressions in C#, we have had two different style of programming. Yet, programmers used to the habitual style often find it easy to fall back on those old practices. In this presentation we will take a number of common tasks we code in C#, discuss the downsides of the habitual style, transform it into functional style, and discuss the benefits. We will also discuss some techniques that can help make this transformation easier on everyday projects.

- **12:15 pm to 1:00 pm Applying functional programming principles to large scale data processing by Kishore Nallan**
  Location: Track 1(GBR1)
  URL: http://confengine.com/functional-conf-2014/proposal/354/applying-functional-programming-principles-to-large-scale-data-processing
  Notes: At Indix, we deal with a stream of unstructured and constantly changing data. This data is processed through a series of systems before being fed as structured input to our analytics system. In this talk, I will walk through our experience of building a large scale data processing system using Hadoop that's focused on immutability, composition and other functional programming principles.

- **1:00 pm to 2:00 pm Lunch**
  Location:

- **2:00 pm to 3:00 pm Functional Programming in Java by Premanand Chandrasekaran**
  Location: Track 1(GBR1)
  URL: http://confengine.com/functional-conf-2014/proposal/321/functional-programming-in-java
  Notes: Functional programming has started (re)gaining prominence in recent years, and with good reason too. Functional programs lend an elegant solution to the concurrency problem, result in more modular systems, are more concise and are easier to test. While modern languages like Scala and Clojure have embraced the functional style whole-heartedly, Java has lagged a bit behind in its treatment of functions as first-class citizens. With the advent of Java 8 and its support for lambdas, however, Java programmers can finally start reaping the power of functional programs as well. Even without Java 8, it is possible to adopt a functional style with the aid of excellent libraries such as Guava.

  This talk will explore how to apply functional concepts using the Java programming language and demonstrate how it can result in simpler, more elegant designs. We will conduct this in a hands-on workshop style with attendants being encouraged to code-along. So bring your favorite Java 8 aware IDE, an open mind and prepare to have a lot of fun.

- **2:00 pm to 3:00 pm Compile your own cloud with Mirage OS v2.0 by Thomas Gazagnaire**
  Location: Track 2(GBR2)
  URL: http://confengine.com/functional-conf-2014/proposal/476/compile-your-own-cloud-with-mirage-os-v20
  Notes: Most applications running in the cloud are not optimized to do so. They make assumptions about the underlying operating system, resulting in larger footprints with increased costs and risks. The open source Mirage OS represents a new approach where the application code is combined with the specific components of the operating system it needs into a single-purpose unikernel appliance. With Mirage OS, developers can create lean and efficient unikernels for secure, cost-effective and high-performance network applications. Mirage OS unikernels run directly on the Xen Project hypervisor, which allows them to be quickly deployed to many leading cloud platforms.

  Mirage OS is fully written in OCaml, from the device drivers and network stack to higher-level synchronisation protocols and databases. In this presentation I will explain how we developed Mirage OS and why we choose to do so in a strongly typed functional language with a powerful module langage. I will then present some of the new features of Mirage OS v2.0 such as: support for ARM devices, Irmin: a Git-like distributed database and OCaml-TLS: a comprehensive implementation of the TLS protocol in pure OCaml.

- **3:00 pm to 3:15 pm Coffee/Tea Break**
  Location:

- **3:15 pm to 4:00 pm Property based testing for functional domain models by Debasish Ghosh**
  Location: Track 2 (GBR2)
  URL: http://confengine.com/functional-conf-2014/proposal/386/property-based-testing-for-functional-domain-models
  Notes: Manual testing is something that's error prone, incomplete and impossible to replicate on a large scale. We have instead been using xUnit style of testing for quite some time now. This approach has a number of drawbacks like (a) We need to write test cases by hand which again doesn't scale for large systems (b) We may miss out some of the edge cases (c) Safeguarding missing cases with coverage metrics doesn't help, since metrics are mostly based on heuristics (d) maintaining test cases and test data is a real pain.

  In property based testing we write properties and not low level test cases. And let the system generate test cases which validate such properties. There are 2 main advantages with this approach:

  1. You think in terms of properties (or specifications) of the domain model which is the right granularity to think about
  2. You don't need to manage test cases, which is completely done by the system that generates a large collection of test data

  This approach is ideal for the functional programming paradigm, which focuses on pure functions. Using functional programming it's easier to reason about your model - hence it's easier to test functional programs using properties. In this talk I will take some real world examples of property validation and verification using scalacheck (the property based testing library for Scala) and a real world domain model.

- **3:15 pm to 4:00 pm Functional Programming in Ruby by Keith Bennett from**
  Location: Track 1(GBR1)
  URL: http://confengine.com/functional-conf-2014/proposal/388/functional-programming-in-ruby
  Notes: Although Ruby is not known as a functional language, it does support higher order functions in the form of lambdas and procs. Ruby's support for both object oriented and functional approaches, along with its conciseness, clarity, and expressiveness, make it an excellent choice as a general purpose programming language.

  This session, geared toward the functional novice, shows how to implement functional approaches in Ruby, and shows why you would want to.

  Topics covered will include:

  - in testing, using lambdas to verify that certain behaviors do or do not not raise errors
  - lambdas as predicates
  - deferred execution
  - composite functions
  - nested functions
  - using lambdas to hide variables
  - functions that return functions (partial application, currying)
  - lightweight event handling
  - chaining behavior with lambda arrays
  - how lambdas differ from conventional Ruby methods

- **4:00 pm to 4:15 pm Break**
  Location:

- **4:15 pm to 5:15 pm Clojurescript and Om - Pragmatic functional programming in the Javascript Land by Vagmi Mudumbai from**
  Location: Track 1(GBR1)
  URL: http://confengine.com/functional-conf-2014/proposal/326/clojurescript-and-om-pragmatic-functional-programming-in-the-javascript-land
  Notes: Javascript programmers have had a lot of choices when it comes to programming. There were days of mootools, scriptaculous and jQuery and then there are now days of Angular, Ember, Knockout and the like. As a javascript programmer myself, I find that Clojurescript/React as Om offers a fresh perspective into building performant Javascript UIs that are easy to write.

  The talk will introduced concepts of React, immutable datastructures in Clojure and live code an application that demonstrates the concepts.

- **4:15 pm to 5:15 pm Code Jugalbandi by Dhaval Dalal and Ryan Lemmer from**
  Location: Track 2(GBR2)
  URL: http://confengine.com/functional-conf-2014/proposal/385/code-jugalbandi

Thursday, 9 October

Week 41 of 2014

Notes: In Indian classical music, we have Jugalbandi, where two lead musicians or vocalist engage in a playful competition. There is jugalbandi between Flutist and a Percussionist (say using Tabla as the instrument). Compositions rendered by flutist will be heard by the percussionist and will replay the same notes, but now on Tabla and vice-versa is also possible.

In a similar way, we will perform Code Jugalbandi to see how the solution looks using different programming languages and paradigms.

During the session, Dhaval and Ryan will take turns at coding the same problem using different languages and paradigms. There would be multiple such rounds during the Jugalbandi.

5:15 pm to 5:30 pm Break
Location:

5:30 pm to 6:15 pm Discovering Functional Treasure in Idiomatic Groovy by Naresha K
Location: Track 1(GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/328/discovering-functional-treasure-in-idiomatic-groovy
Notes: Groovy is a dynamic language on the JVM. Groovy supports programming in multiple paradigms - imperative, object oriented and even functional programming.

When I started using Groovy with Java background, the code used to be mostly imperative. As I explored the language in detail, I realized the power of idiomatic code. While the attempt to write idiomatic Groovy code helped me to realise the benefits of functional approach, thinking functionally resulted in better code too.

In this talk, I will demonstrate functional programming constructs in Groovy and show how to use them effectively. I will provide plently of examples to help the audience realize the benefits.

5:30 pm to 6:15 pm Learning (from) Haskell - An experience report by Aditya Godbole
Location: Track 2(GBR2)
URL: http://www.confengine.com/functional-conf-2014/proposal/320/learning-from-haskell-an-experience-report
Notes: Functional programming as a programming style and discipline is useful even in languages which are not pure functional languages. By practising programming in a pure functional language like Haskell, programmers can drastically improve the quality of code when coding in other languages as well.

The talk is based on first hand experience of using Haskell in internal courses in our organisation to improve code quality.

This talk will cover Gofer (one of the earliest variants of Haskell) as a teaching tool, including the choice of the language, the features from Haskell that should (and shouldn't) be covered and the obstacles and benefits of the exercise.

6:15 pm to 6:30 pm Break
Location:

6:30 pm to 7:30 pm Fish Bowl
Location: Track 1(GBR1)
URL: http://en.wikipedia.org/wiki/Fishbowl_(conversation)

7:30 pm to 10:00 pm Dinner & Networking
Location:

Friday, 10 October

Week 41 of 2014

■ Functional Conf 2014 Important Dates

October 2014

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	**10**	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

November 2014

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Functional Conference 2014 09/10 to 11/10

8 am

9 am

9:00 am Keynote – Fear: The Role of Fear in Language Adoption by Bruce Tate
Track 1(GBR1)

10 am

10:00 am Important Announcements

10:15 am Coffee/Tea Break

10:30 am Thinking in LINQ by Sudipta Mukherjee
Track 1(GBR1)

10:30 am Pragmatic Functional Programming using Dyalog by Morten Kromberg
Track 2(GBR2)

11 am

11:30 am Break

11:45 am Writing and improving tail recursive functions by Bhasker Kode
Track 1(GBR1)

11:45 am Monads you already use (without knowing it) by Tejas Dinkar
Track 2(GBR2)

Noon

12:30 pm Lunch Break

1 pm

1:30 pm Distributed computing: dealing with Time and Failure in the wild by Ryan Lemmer
Track 2(GBR2)

1:30 pm Purely functional data structures demystified by Mohit Thatte
Track 1(GBR1)

2 pm

2:15 pm Break

2:30 pm Demystify the Reactive Jargons by Mushtaq Ahmed
Track 2(GBR2)

2:30 pm An introduction to Continuation Passing Style (CPS) by Ramakrishnan Muthukrishnan
Track 1(GBR1)

3 pm

3:30 pm Coffee/Tea Break

3:45 pm Elixir : a round-up on state of Elixir and it's ecosystem by Akash Manohar
Track 1(GBR1)

3:45 pm Object-functional programming: Beautiful unification or a kitchen sink? by Rahul Goma Phulore
Track 2(GBR2)

4 pm

4:45 pm Break

5 pm

5:00 pm Keynote – Methodologies, Mathematics, and the Metalinguistic Implications of Swift by Daniel Steinberg
Track 1(GBR1)

6 pm

6:00 pm Closing Talk

7 pm

● All-Day Events

09/10 to 11/10 Functional Conference 2014
Location: http://functionalconf.com

● Timed Events

9:00 am to 10:00 am Keynote – Fear: The Role of Fear in Language Adoption by Bruce Tate
Location: Track 1(GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/711/fear-the-role-of-fear-in-language-adoption
Notes: Old languages emerge and new languages are born when big things happen, and big things are happening now. In this keynote, we'll look at some of the biggest challenges facing programming evolution, and their likely impacts on programming.

Functional Programming Languages have a prominent role, but also there are interesting things happening in the browser and a pendulum shift toward better type models.

10:00 am to 10:15 am Important Announcements
Location: Track 1(GBR1)

10:15 am to 10:30 am Coffee/Tea Break
Location:

10:30 am to 11:30 am Thinking in LINQ by Sudipta Mukherjee
Location: Track 1(GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/690/thinking-in-linq
Notes: LINQ draws on principles of functional programming and represents a paradigm shift for developers used to an imperative/object oriented programming style. Thinking in LINQ explains the benefits of functional programming built into LINQ, allowing developers to use these techniques write more efficient and concise data-intensive applications.

While other books on the subject merely scratch the surface in terms of problem solving using LINQ, Thinking in LINQ shows readers how use functional programming techniques to solve common every-day problems as well as more complex problems using LINQ's features.

LINQ lets you write code that resembles natural language and is easier to debug compared to traditional loops and branching statements. The purpose of a well written LINQ Query will be immediately evident unlike the looping construct so commonly used in traditional programming. LINQ operators can be used in unison to orchestrate a solution for complex real world problems.

What viewers will learn

- Text Processing using LINQ
- Refactoring using LINQ
- Monitoring code health using LINQ
- Creating DSL using LINQ

10:30 am to 11:30 am Pragmatic Functional Programming using Dyalog by Morten Kromberg
Location: Track 2(GBR2)
URL: http://confengine.com/functional-conf-2014/proposal/436/pragmatic-functional-programming-using-dyalog
Notes: APL is a member of the family of languages that are approaching middle age (Ken Iverson's book titled "A Programming Language" was published in 1962). APL was very influential in the 60's and 70's, and widely used to deliver "end user computing" – but although the REPL, dynamic scope and lack of a type system endeared APL to domain experts, it also drew fire from computer scientists, most famously when Edsger Dijkstra declared that "APL is a mistake, carried through to perfection. It is the language of the future for the programming techniques of the past it creates a new generation of coding bums."

Dyalog is a modern, array-first, multi-paradigm programming language, which supports functional, object-oriented and imperative programming based on an APL language kernel. Dyalog allows people with good ideas – from bright high school students to PhDs – to contribute directly to the software development process using a notation which fits comfortably with those used in their own domains. Subject matter experts can write prototypes or, with suitable training and/or support, highly efficient, parallel and robust code that can be embedded in high-performance production applications.

11:30 am to 11:45 am Break
Location:

11:45 am to 12:30 pm Writing and improving tail recursive functions by Bhasker Kode
Location: Track 1(GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/322/writing-and-improving-tail-recursive-functions
Notes: Brief history of recursion

Snippets from a few languages

What is tail recursion?

Design choices around recursion

The importance of tail recursion in erlang

How do you profile such improvements?

11:45 am to 12:30 pm Monads you already use (without knowing it) by Tejas Dinkar
Location: Track 2(GBR2)
URL: http://confengine.com/functional-conf-2014/proposal/437/monads-you-already-use-without-knowing-it
Notes: Monads are a little bit like Quantum Physics: If you think you understand quantum mechanics, you don't understand quantum mechanics.

Monads are very useful for chaining computation together, in a simple way. The best explanation I've heard for them so far is that they are `programmable semicolons'.

In this session, I'll describe a few patterns that are solved by monads in some FP languages, and how you are already using them.

Some monads I plan to cover:

* Maybe Monad (being the easiest to explain)

* List monad, and how it is used to model non-determinism

* The state monad

* The IO monad

And maybe a few others

12:30 pm to 1:30 pm Lunch Break
Location:

1:30 pm to 2:15 pm Distributed computing: dealing with Time and Failure in the wild by Ryan Lemmer
Location: Track 2(GBR2)
URL: http://confengine.com/functional-conf-2014/proposal/387/realtime-distributed-computing-dealing-with-time-and-failure-in-the-wild
Notes: There is a growing need for scalable, realtime business systems that are continuously running and highly-available. Two very different frameworks/approaches you could use to build such systems are Storm and Akka.

Systems created with Storm or Akka are distributed, at runtime, on as many machines as you choose. The inherent concurrency implied by this brings the issues of State, Time and Failure into sharp focus. Functional programming has much to say about dealing with state and time; not surprisingly, both Storm and Akka have strong roots in functional languages (for Storm it is Clojure, and for Akka, Scala).

In this talk we'll explore the core concepts and challenges of distributed computation; the role of functional programming in concurrent distributed computing; we'll take a look at Storm and Akka, by example, and see that as different as these 2 approaches are, the underlying difficulties of distributed computation remains evident in both: dealing with time, and dealing with failure.

1:30 pm to 2:15 pm Purely functional data structures demystified by Mohit Thatte
Location: Track 1(GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/325/purely-functional-data-structures-demystified
Notes: Immutable, persistent data structures form a big part of the value proposition of most functional programming languages.

It is important to understand why these data structures are useful and how they make it easier to reason about your program.

It is also instructive to see how these data structures are implemented to get a greater appreciation for the inherent tradeoffs between performance and immutability.

In this talk I will do a walkthrough of some of these data structures drawing from the work of Chris Okasaki[1], and attempt to explain the essential ideas in a simple way.

2:15 pm to 2:30 pm Break
Location:

2:30 pm to 3:30 pm Demystify the Reactive Jargons by Mushtaq Ahmed
Location: Track 2(GBR2)
URL: http://confengine.com/functional-conf-2014/proposal/383/demystify-the-reactive-jargons
Notes: Sync, Async, Blocking, Non-Blocking, Streaming are the buzzwords in the reactive programming world. This talk will attempt to attach some meaning to them. It will also demo the performance and resource consumption patterns for blocking-io, Scala Futures and RxJava Observables for comparable programs. Finally, a command line application that consumes twitter streams API will demo what is possible using the new reactive abstractions.

2:30 pm to 3:30 pm An introduction to Continuation Passing Style (CPS) by Ramakrishnan Muthukrishnan
Location: Track 1(GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/359/-an-introduction-to-continuation-passing-style-cps
Notes: Traditionally functions return some value. Someone is waiting for that value and does some computation with it. This "someone" is called the continuation of this value. In a normal functional call, the continuation is "implicit". In the "continuation passing style" (hence forth called with the short form, CPS), we make the continuations explicit. In this style, function definitions take an extra argument called "continuation" and it never return. The "return value" of the function 'continues' by passing this value as an argument to the continuation. Continuations are sometimes called "gotos with arguments".

CPS is used as an intermediate stage while compiling a program since it makes the control structure of the program explicit and hence can be converted easily to machine code. Another feature of a CPS-transformed function is that it is tail-recursive even if the original function was not written in a tail-recursive style.

Continuations enable a programmer to build new control operators (if the language's built-in operators does not already provide the control operators the programmer need).

3:30 pm to 3:45 pm Coffee/Tea Break
Location:

3:45 pm to 4:45 pm Elixir : a round-up on state of Elixir and it's ecosystem by Akash Manohar
Location: Track 1(GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/452/elixir-today-a-round-up-on-state-of-elixir-and-its-ecosystem
Notes: Elixir is a functional and dynamic language built on top of the Erlang VM. The development of the language is happening at a fast pace. People in the community have participated actively to write tools and libraries, required to write real-world apps in Eilxir.

In this talk, I will attempt to skim through all the new features in Elixir, a few important libraries with short examples and some learning resources.

Each tool showcased in this talk, will have a 3-step tryout, with the simplest example.

3:45 pm to 4:45 pm Object-functional programming: Beautiful unification or a kitchen sink? by Rahul Goma Phulore
Location: Track 2(GBR2)
URL: http://confengine.com/functional-conf-2014/proposal/412/

object-functional-programming-beautiful-unification-or-a-kitchen-sink
Notes: Scala began its life as an experiment to “unify” object-oriented programming and functional programming. Martin Odersky believed that the differences between FP and OO are more cultural than technical, and that there was a room for beautifully unify various ideas from the two into one simple core.

How successful has Scala been in its goals? Is it the like “the grand unified theory of universe” or like the infamous “vegetarian ham”? [1]

In this talk, we will see just how Scala unifies various ideas - such as type-classes, algebraic data types, first-class modules, functions under one simple core comprising of traits, objects, implicits, and open recursion. We will how this unification unintendedly subsumes many concepts that require seprate features in other languages, such as functional dependencies, type families, GADTs in Haskell. We will see how this has given a rise to a new “implicit calculus”, which could lay a foundation for next generation of generic programming techniques.

We will see that this unification comes at a certain cost, wherein it leads to some compromises on both sides. However many of these trade-offs are particular to Scala (largely due to the JVM imposed restrictions). The goal of unification is still noble, and we need not throw the baby out with the bathwater.

[1]: https://twitter.com/bos31337/status/425524860345778176

4:45 pm to 5:00 pm Break
Location:

5:00 pm to 6:00 pm Keynote - Methodologies, Mathematics, and the Metalinguistic Implications of Swift by Daniel Steinberg
Location: Track 1(GBR1)
URL: http://confengine.com/functional-conf-2014/proposal/712/methodologies-mathematics-and-the-metalinguistic-implications-of-swift
Notes: The rules we agree on define the games we play. We see this in methodologies we adopt for software development, in the mathematics we were forced to learn in high school, and in the syntax and grammar of the languages we choose to use. During this talk we'll explore cases in which the implication of axioms are clear and cases in which they are far from clear. There will be a quiz.

6:00 pm to 6:15 pm Closing Talk
Location: