The Makam Metalanguage

A day in the life of Paul the PL researcher

Antonis Stampoulis Adam Chlipala

Programming Languages and Verification group

PL/SE Offsite 2014


SGT. PEPPER'S
LONELY HEARTS
CLUB BAND
THE BEATLES

(notes from previous slide)

In 1967, Beatles released the record that changed rock music for ever – the Sergeant Peppers Lonely Hearts Club Band. The last track was perhaps the most experimental, most beautiful and most influential. It was called ‘A Day in the Life’ and it was mostly about mundane everyday life, about dreaming and daydreaming, and the space between these. It was also about other things. There were two parts; one was by John, another one by Paul. I’ll focus on Paul’s part for this talk.

Paul’s part

Paul's part

- Woke up

Paul's part

- Woke up
- Fell out of bed

Paul's part

- Woke up
- Fell out of bed
- Dragged a comb across my head

Paul's part

- Woke up
- Fell out of bed
- Dragged a comb across my head
- Found my way downstairs and drank a cup

Paul's part

- Woke up
- Fell out of bed
- Dragged a comb across my head
- Found my way downstairs and drank a cup
- And looking up I noticed I was late

Paul’s part (continued)

Paul’s part (continued)

- Found my coat and grabbed my bag

Paul's part (continued)

- Found my coat and grabbed my bag
- Made the T in seconds flat

Paul's part (continued)

- Found my coat and grabbed my bag
- Made the T in seconds flat
- Found my way inside Stata; there was a talk

Paul's part (continued)

- Found my coat and grabbed my bag
- Made the T in seconds flat
- Found my way inside Stata; there was a talk
- And somebody spoke about a type system with dependent types (and some other fancy features)

Paul's part (continued)

- Found my coat and grabbed my bag
- Made the T in seconds flat
- Found my way inside Stata; there was a talk
- And somebody spoke about a type system with dependent types (and some other fancy features)
- And I went into a dream where I had other new PL and type system design ideas, like a typed tactic language for proof assistants like Coq which I could call VeriML, or a secure web services language that I could perhaps call Ur/Web

(notes from previous slide)

Now the most surprising thing both in 1967 but still to this day, is not the fact that Paul was able to find his way inside Stata, or that they could magically make the penultimate line rhyme – it is the last line that Paul sings, which is the following.

Paul's confounding last line

Paul's confounding last line

"I went back to my office and implemented some of these ideas before I went home to cook dinner"

Paul's confounding last line

"I went back to my office and implemented some of these ideas before I went home to cook dinner"

?!?

(notes from previous slide)

How is this possible?!? Usually implementations like these take months! Was this just the substances that they were taking at the time or should we take it at face value?

(notes from previous slide)

Lately I have figured out this mystery. It is well-known that the Beatles had studied the Raga system of classical Indian music. I have come to discover that they had also studied the Makam system of traditional Arabic and Turkish music.

In 1967 the Beatles knew about Makam

In 1967 the Beatles knew about Makam

- a set of scales, rules and patterns for melodic improvisation

  (14th century and to this day, developed in the Near East, Central Asia and North Africa)

In 1967 the Beatles knew about Makam

- a set of scales, rules and patterns for melodic improvisation

  (14th century and to this day, developed in the Near East, Central Asia and North Africa)

- a metalanguage and a set of libraries that make programming language prototyping quick and pleasant

  (2013 - 2014, developed at MIT)

(notes from previous slide)

The Makam system, is a set of scales, rules and patterns for melodic improvisation. It also has a large subpart that is not as well-documented: an expressive metalanguage which is surprisingly rich though it is relatively small; I am designing and implementing it with Adam during my post-doc, starting in 2013. The Beatles were indeed quite ahead of their time.

declarative and executable specifications

Prolog

built-in, sophisticated support for binding

Higher-order abstract syntax + Higher-order unification

reusable data structures like lists

Polymorphism

reusable binding structures like multiple binding and linearity

Unrestricted higher-order predicates

generic operations and traversals like structural recursion

Dynamic typing baked into unification

extensions to the meta-language definable within it

Staging and reflection

OCaml type system	550 lines
Type classes	100 lines
Higher-order logic	250 lines
VeriML constructs	150 lines
System F to TAL	850 lines
PEG parser gen	350 lines
LF	350 lines

(notes from previous slide)

The reusable library that all of these depend on is around 600 lines. All of these examples are quite readable and pleasant to code; sometimes nicer to write than the LaTeX code for the typeset version of the type system rules.

Summary

Summary

Because of Makam, the song has the catchy title
‘A day in the life’, not ‘Six months in the life’

Summary

Because of Makam, the song has the catchy title ‘A day in the life’, not ‘Six months in the life’

(notes from previous slide)

We are as excited as the Beatles are for Makam; catch us afterwards and we will be glad to tell you more about it!