

BEAM Language Ecosystem

Past, Present and Future



Mariano Guerra - Robert Virding • 07.11.2020
Code BEAM Brasil



Co-founder @ Instadeq
Efene, Interfix, EFE

Language Interop @ ErIEF



Co-creator of Erlang
LFE, Luerl, Erlog

Language Interop @ ErIEF

State of the BEAM Language Ecosystem

LISP

- LFE
- Clojerl
- Joxa

Perl / Python / Ruby / Perl / PHP

- Reia
- Elixir
- Efene
- EPHP

State of the BEAM Language Ecosystem

Rust / OCaml / SML / F#

- Alpaca
- Fez
- Gleam
- Caramel

Haskell / Elm

- PureI
- Hamler



First Wave

1. Reia: reia-lang.org
2. Elixir
3. LFE
4. Efene

New Wave



1. Alpaca
2. Purerl
3. Gleam
4. Hamler
5. Whatsapp's Typed Erlang?

Middle Road

Success / Progressive Typing

1. Type Specs
2. Dialyzer
3. Whatsapp's Typed Erlang?

Porting Existing Languages

Mechanical Sympathy

“You don’t have to be an engineer to be a racing driver, but you do have to have Mechanical Sympathy.”

– Jackie Stewart, racing driver

1. Functional
 2. Immutable
 3. Eager Evaluation
-

Impedance Mismatch

Impedance matching problem exists when transferring sound energy from one medium to another. If the acoustic impedance of the two media are very different most energy will be reflected (or absorbed), rather than transferred across

1. “Object Oriented”
 2. Mutable
 3. Lazy Evaluation
-

*“OOP to me means only messaging,
local retention and protection and
hiding of state-process, and extreme
late-binding of all things”*

-- Alan Kay

Languages inspired by existing ones
but playing to the BEAM strengths
tend to work better

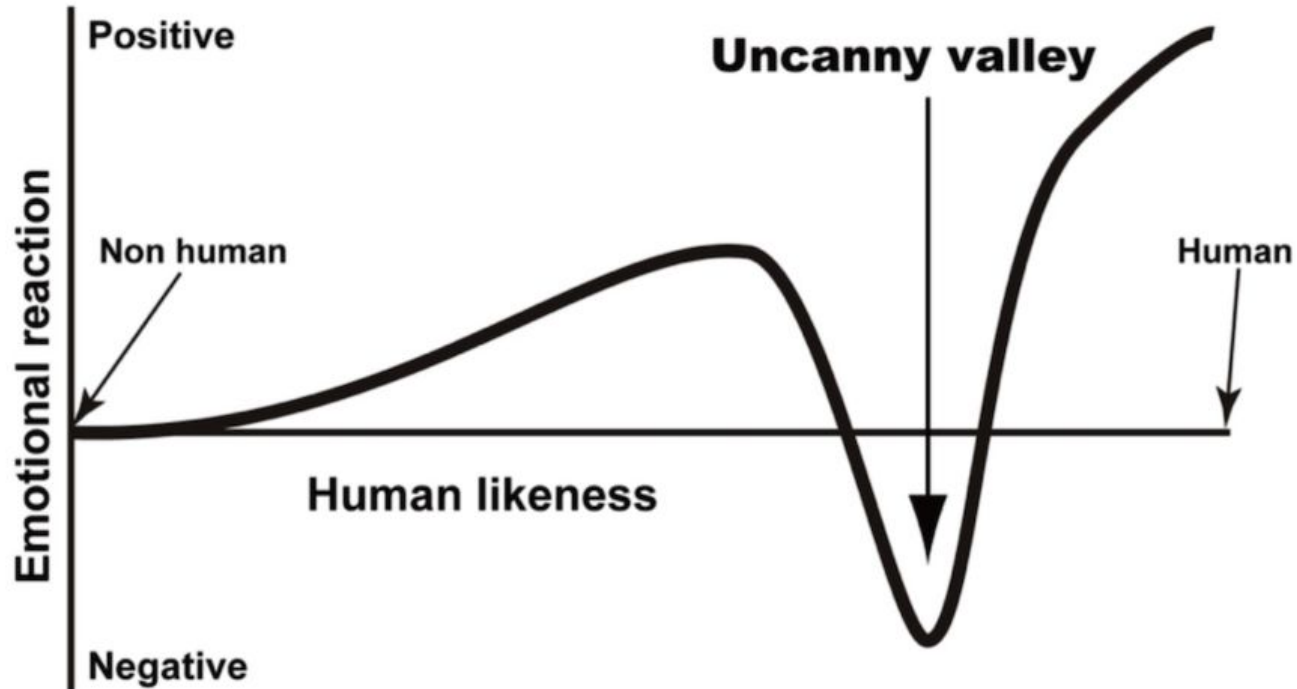
Happens on other platforms too

JVM: Scala, Clojure*, Kotlin, Groovy

.NET: F#, Clojure*

Javascript: Typescript, Clojurescript*

The Uncanny Valley of Compatibility



Language Design / Implementation Tradeoffs

Case: Elixir



Designing a Common Lisp inspired programming language



Porting a mutable dynamically typed language



Porting a logical programming language

Erlog

○

How much is shared across languages?

- **Lexer**
- **Pre Processor: includes, macros**

Streams

- **Parser: absform / AST**
- **Core Erlang**
- **Kernel Erlang**

Trees

- **SSA**

Graph

- **Codegen**

Stream

Elixir Flavoured Erlang

github.com/marianoguerra/efe

github.com/marianoguerra/otp.ex

```

354 def sort([[x, y] | 1]) do
355   case 1 do
356     [] ->
357       [y, x]
358
359     [z] when x <= z ->
360       [[y, x] | 1]
361
362     [z] when y <= z ->
363       [y, z, x]
364
365     [z] ->
366       [z, y, x]
367
368     _ ->
369       split_2(x, y, 1, [], [])
370   end
371 end

```

lists:sort

```

80 def cast({:global, name}, request) do
81   try do
82     :global.send(name, cast_msg(request))
83   catch
84     :error, e -> {:EXIT, {e, __STACKTRACE__}}
85     :exit, e -> {:EXIT, e}
86     e -> e
87   end
88
89   :ok
90 end
91
92 def cast({:via, mod, name}, request) do
93   try do
94     mod.send(name, cast_msg(request))
95   catch
96     :error, e -> {:EXIT, {e, __STACKTRACE__}}
97     :exit, e -> {:EXIT, e}
98     e -> e
99   end
100
101   :ok
102 end

```

gen_server:cast

Interoperability Challenges

“There are only two kinds of languages: the ones people complain about and the ones nobody uses.”

— Bjarne Stroustrup, C++ Creator

1. Records / Structs
 2. Sum Types
 3. Namespaces / Modules
 4. Naming Conventions
 5. Statically Typed Languages calling Dynamically Typed Code
 6. Code Dependencies across Languages
-

The Future

"The best way to predict the future is to create it."

1. Polyglot Projects
 2. Minimal Interop Friction
 3. BEAM as the obvious choice for languages / systems research
-



 @warianoguerra



 @rvirding

Typing Message Passing

Interoperability and Backward Compatibility

Why not add some mutability to the BEAM