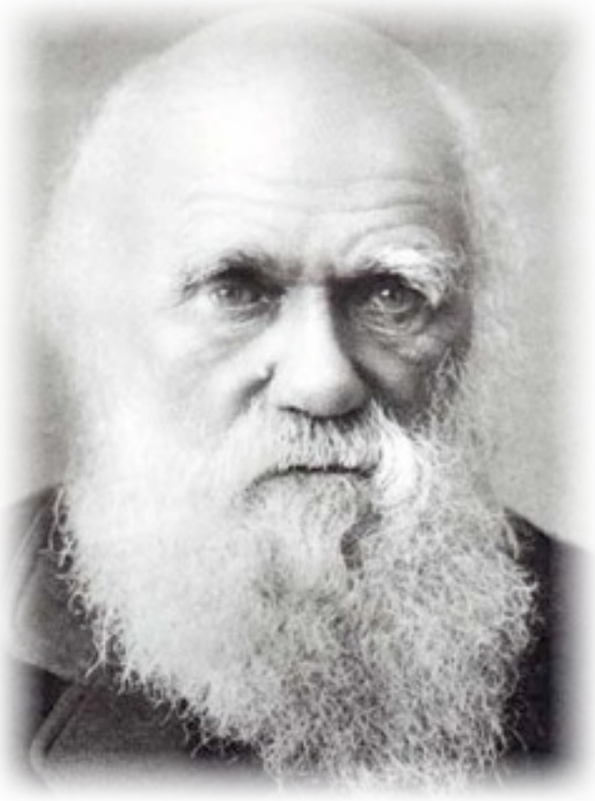
Evolution in Programming Language Design?

Tom Van Cutsem



The Evolution of Language

The Evolution of Language

$2x$ (Descartes)

The Evolution of Language

$2x$ (Descartes)

$\lambda x.\, 2x$ (Church)

The Evolution of Language

$2x$ (Descartes)

$\lambda x.\, 2x$ (Church)

`(LAMBDA (X) (* 2 X))` (McCarthy)

The Evolution of Language

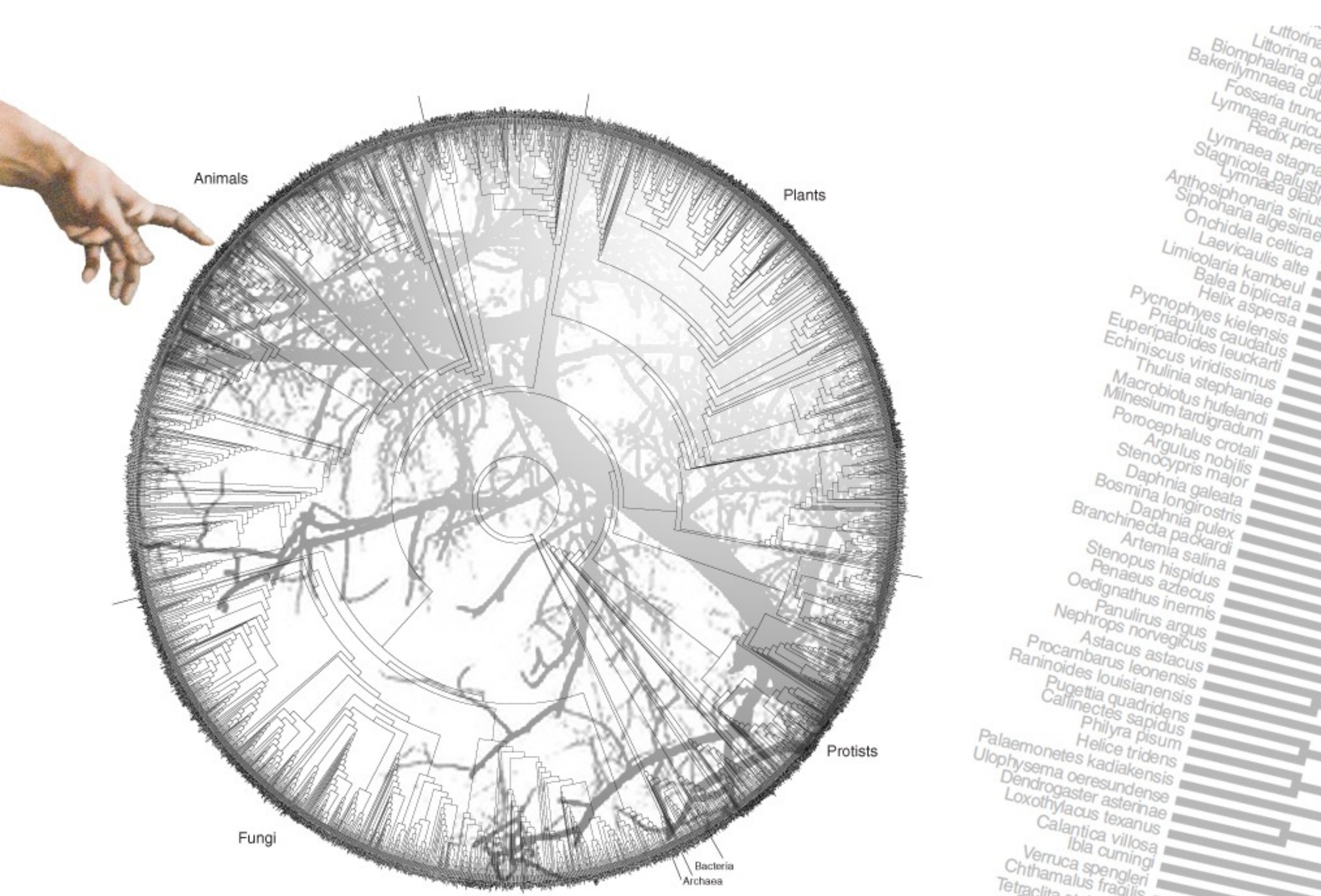
$2x$ (Descartes)

$\lambda x.\, 2x$ (Church)

```
(LAMBDA (X) (* 2 X))
```
(McCarthy)

```
<?xml version="1.0"?>
<LAMBDA-TERM>
  <VAR-LIST>
    <VAR>X</VAR>
  </VAR-LIST>
  <EXPR>
    <APPLICATION>
      <EXPR><CONST>*</CONST></EXPR>
      <ARGUMENT-LIST>
        <EXPR><CONST>2</CONST></EXPR>
        <EXPR><VAR>X</VAR></EXPR>
      </ARGUMENT-LIST>
    </APPLICATION>
  </EXPR>
</LAMBDA-TERM>
```
(W3C)

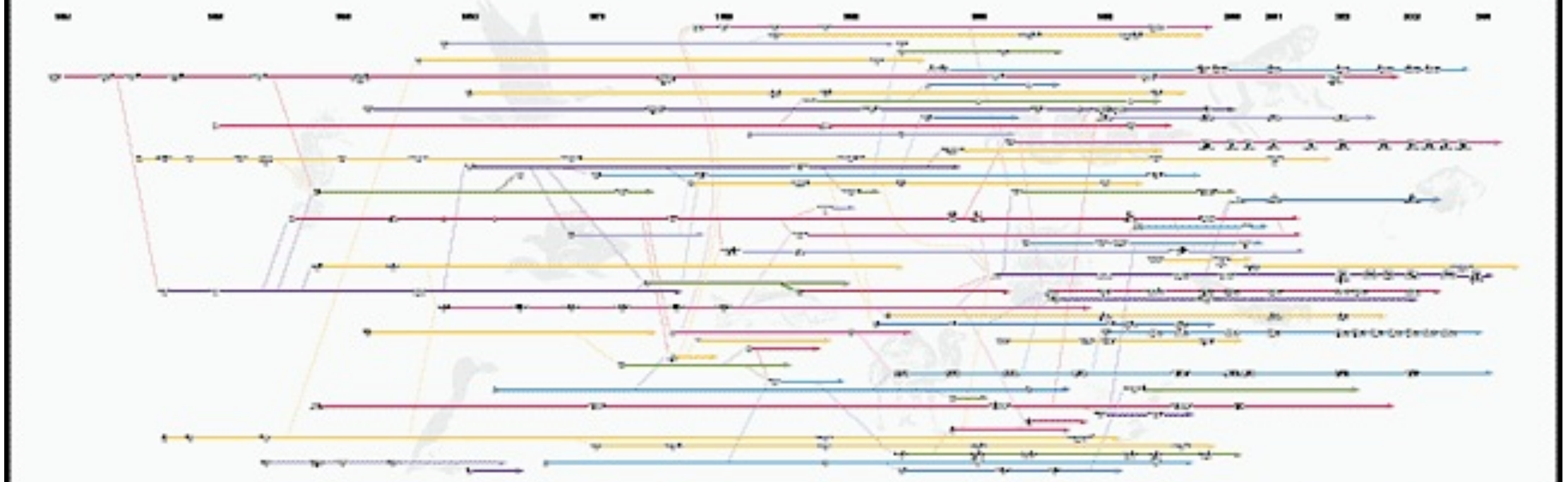
Language design: creation or evolution?



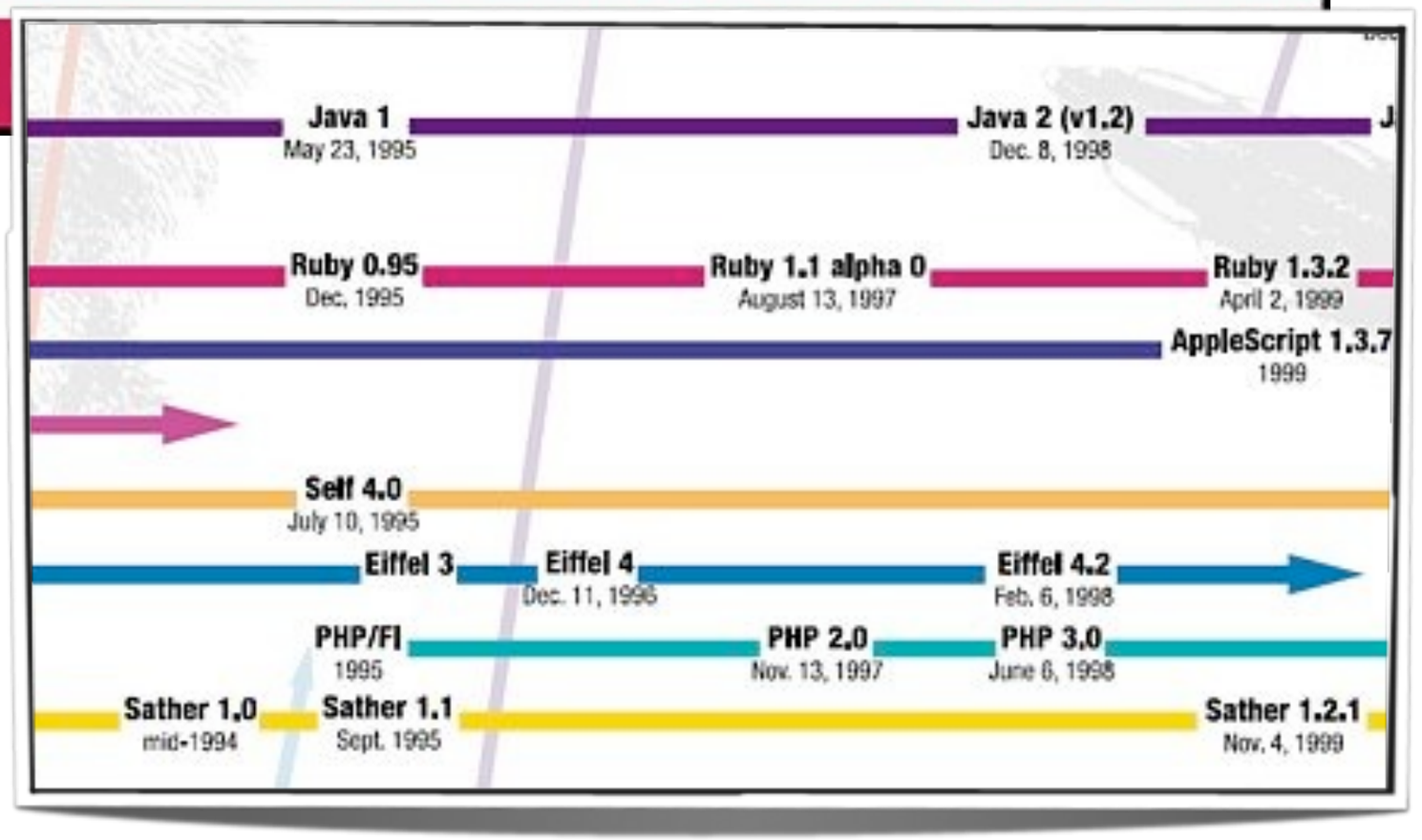
Higgins plot

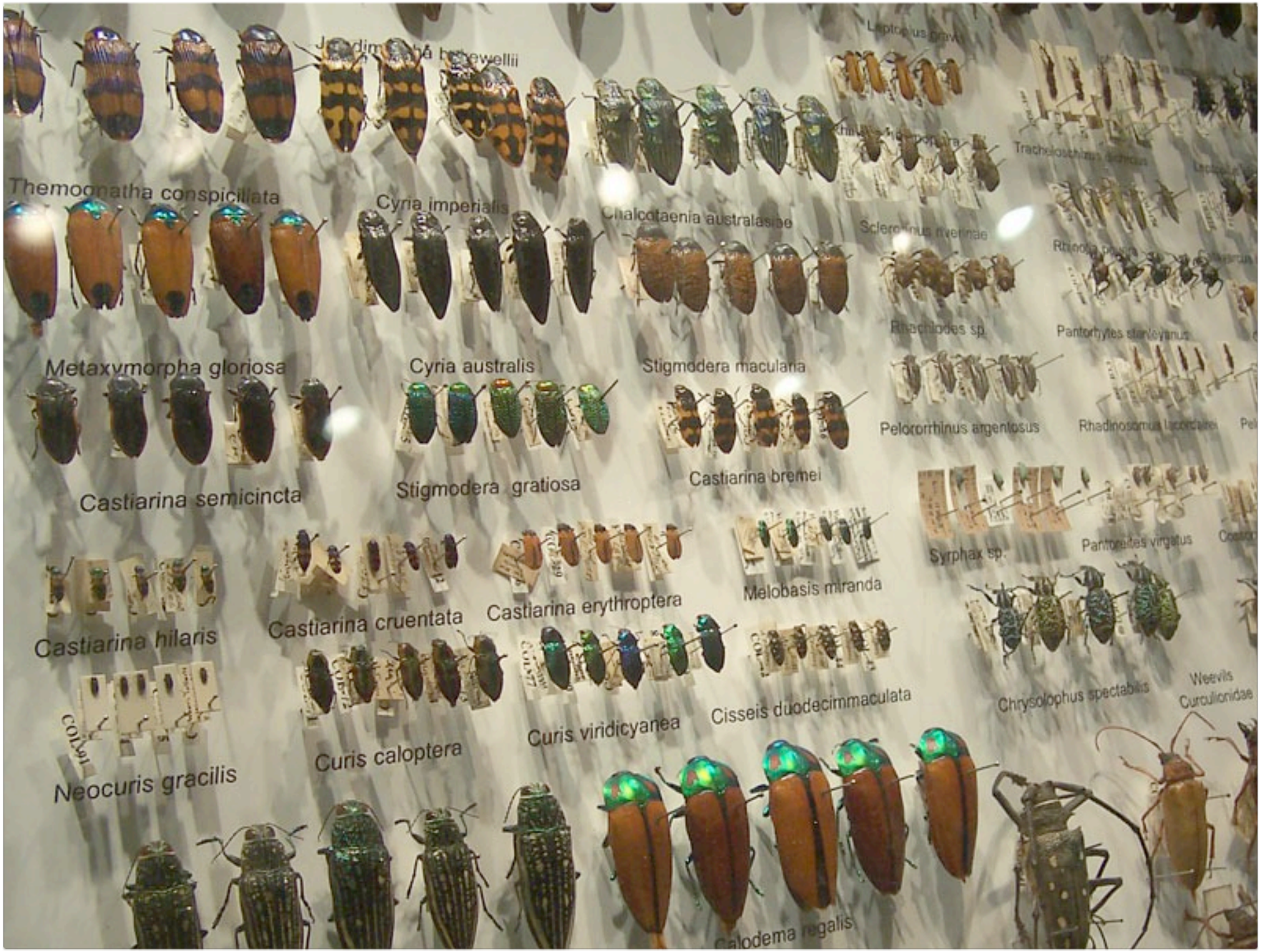
History of Programming Languages

O'REILLY

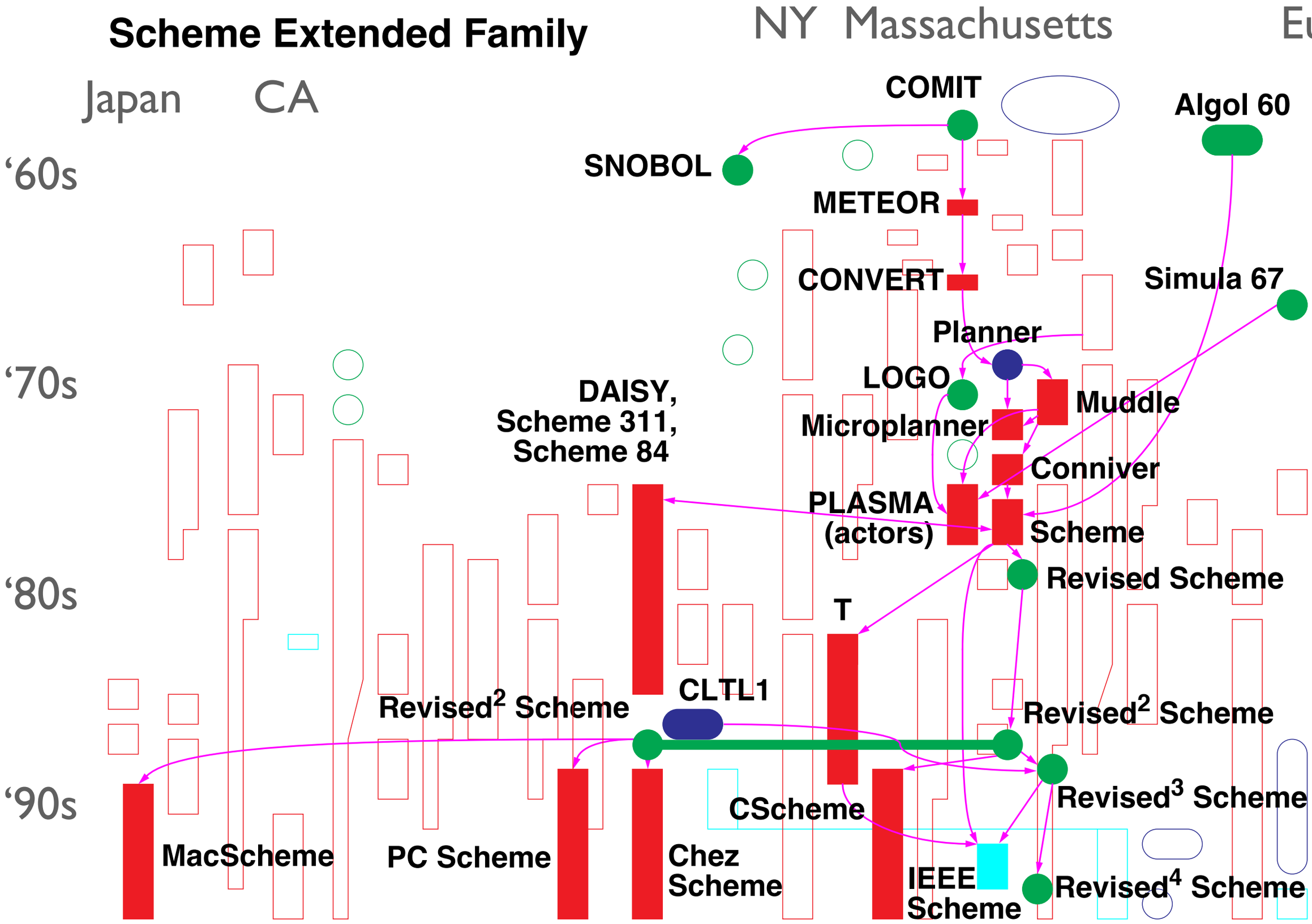


www.oreilly.com



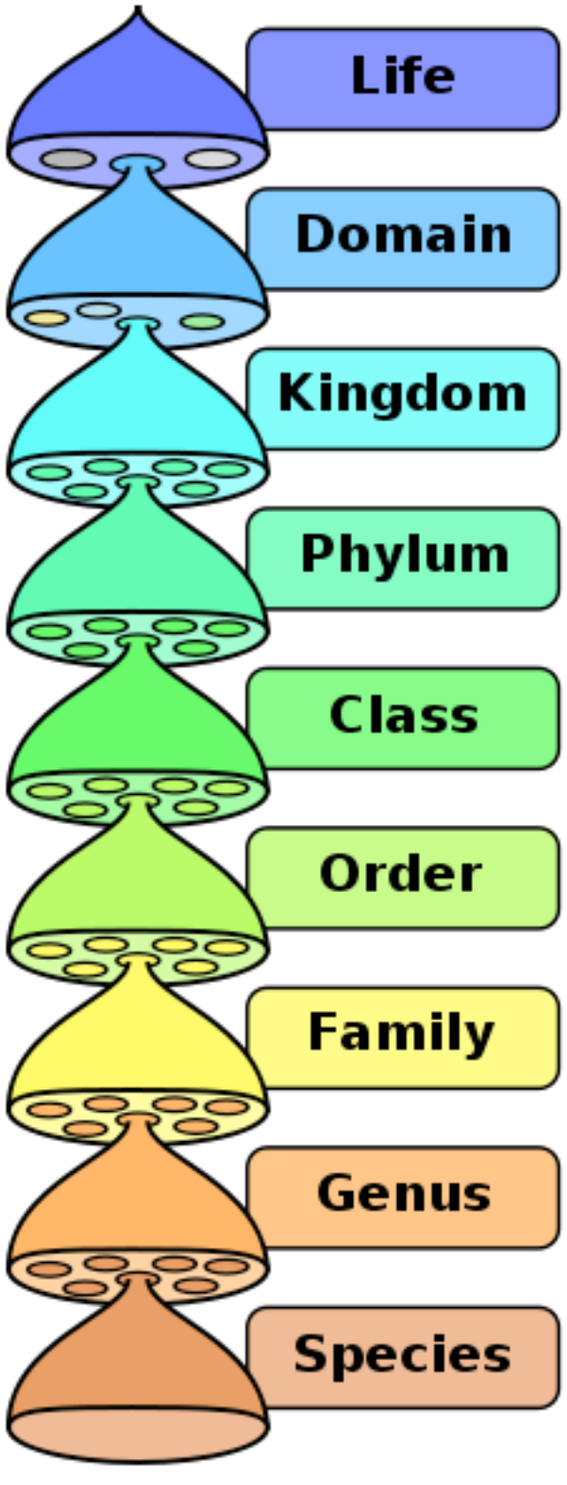

Themoonatha conspicillata
Cyria imperialis
Chalcotaenia australasiae
Metaxymorpha gloriosa
Cyria australis
Stigmodera maculana
Castiarina semicincta
Stigmodera gratiosa
Castiarina bremei
Rhachiodes sp.
Pelocorrhinus argentosus
Syrphax sp.
Castiarina hilaris
Castiarina cruentata
Castiarina erythroptera
Melobasis miranda
Neocuris gracilis
Curis caloptera
Curis viridicyanea
Cisseis duodecimmaculata
Chrysolophus spectabilis
Weevils Curculionidae
Calodema regalis

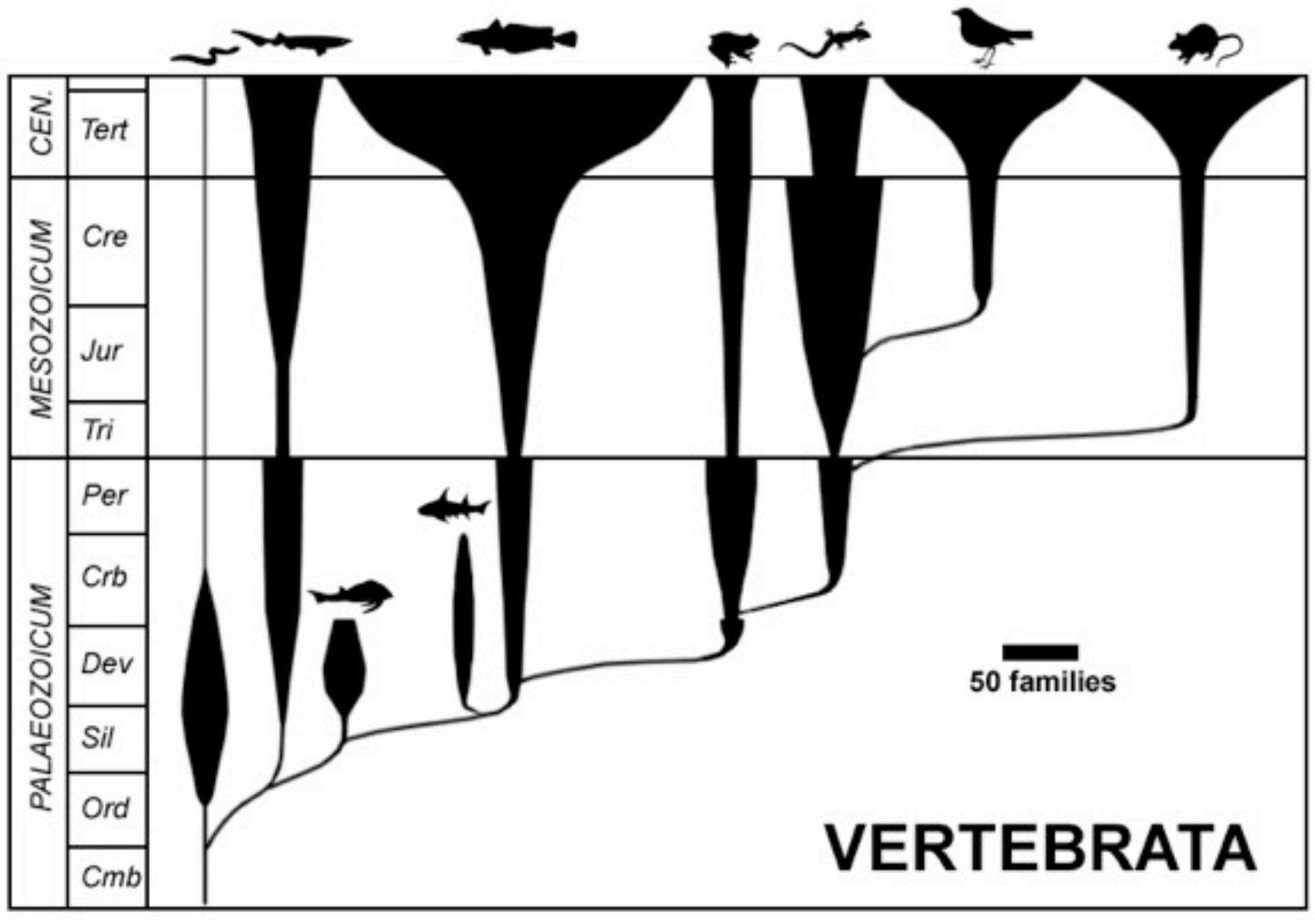
Scheme Extended Family



(Richard Gabriel - http://www.dreamsongs.com/NewFiles/Hopl2Slides.pdf)

Biological classification





(http://en.wikipedia.org/wiki/Biological_classification)

The principal programming paradigms

"More is not better (or worse) than less, just different."

v1.08 © 2008 by Peter Van Roy

Data structures only
Turing equivalent

Observable nondeterminism? Yes No

record
Descriptive declarative programming
XML, S−expression
+ procedure
First−order functional programming
+ closure
Functional programming
Scheme, ML

+ cell (state)
Imperative programming
Pascal, C
Imperative search programming
+ search
SNOBOL, Icon, Prolog

+ unification (equality)
Deterministic logic programming
+ search
Relational & logic programming
Prolog, SQL embeddings
+ solver
Constraint (logic) programming
CLP, ILOG Solver
+ thread
Concurrent constraint programming
LIFE, AKL
+ by−need synchronization
Lazy concurrent constraint programming
Oz, Alice, Curry

+ by−need synchron.
Lazy functional programming
Haskell

+ continuation
Continuation programming
Scheme, ML

+ thread
+ single assign.
Monotonic dataflow programming
Declarative concurrent programming
Pipes, MapReduce
+ thread
+ single assignment
+ by−need synchronization
Lazy dataflow programming
Lazy declarative concurrent programming
Oz, Alice, Curry

+ name (unforgeable constant)
ADT functional programming
Haskell, ML, E
+ cell
ADT imperative programming
CLU, OCaml, Oz

+ nondeterministic choice
Nonmonotonic dataflow programming
Concurrent logic programming
Oz, Alice, Curry, Excel, AKL, FGHC, FCP
+ synch. on partial termination
Functional reactive programming (FRP)
Weak synchronous programming
FrTime, SL
+ instantaneous computation
Strong synchronous programming
Esterel, Lustre, Signal

+ port (channel)
Multi−agent dataflow programming
Oz, Alice, AKL

+ port (channel)
Event−loop programming
E in one vat
+ thread
Multi−agent programming
Message−passing concurrent programming
Erlang, AKL
+ local cell
Active object programming
Object−capability programming
CSP, Occam, E, Oz, Alice, publish/subscribe, tuple space (Linda)

+ cell (state)
+ closure
Sequential object−oriented programming
Stateful functional programming
Java, OCaml
+ thread
Concurrent object−oriented programming
Shared−state concurrent programming
Smalltalk, Oz, Java, Alice
+ log
Software transactional memory (STM)
SQL embeddings

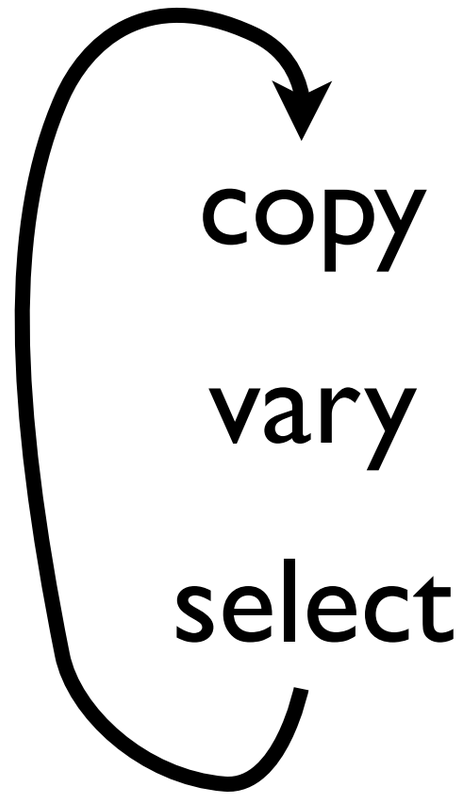
Logic and constraints
Functional
Dataflow and message passing
Message passing
Shared state

Unnamed state (seq. or conc.)
Nondet. state
Named state

More declarative ◄ ► ***Less declarative***

(Peter Van Roy)

Evolutionary process



Universal Darwinism

- Evolutionary processes outside of biology:
  - Economics (competition among firms)
  - Human culture (competition among ideas)
  - Human language (competition among words)
  - ...

Universal Darwinism

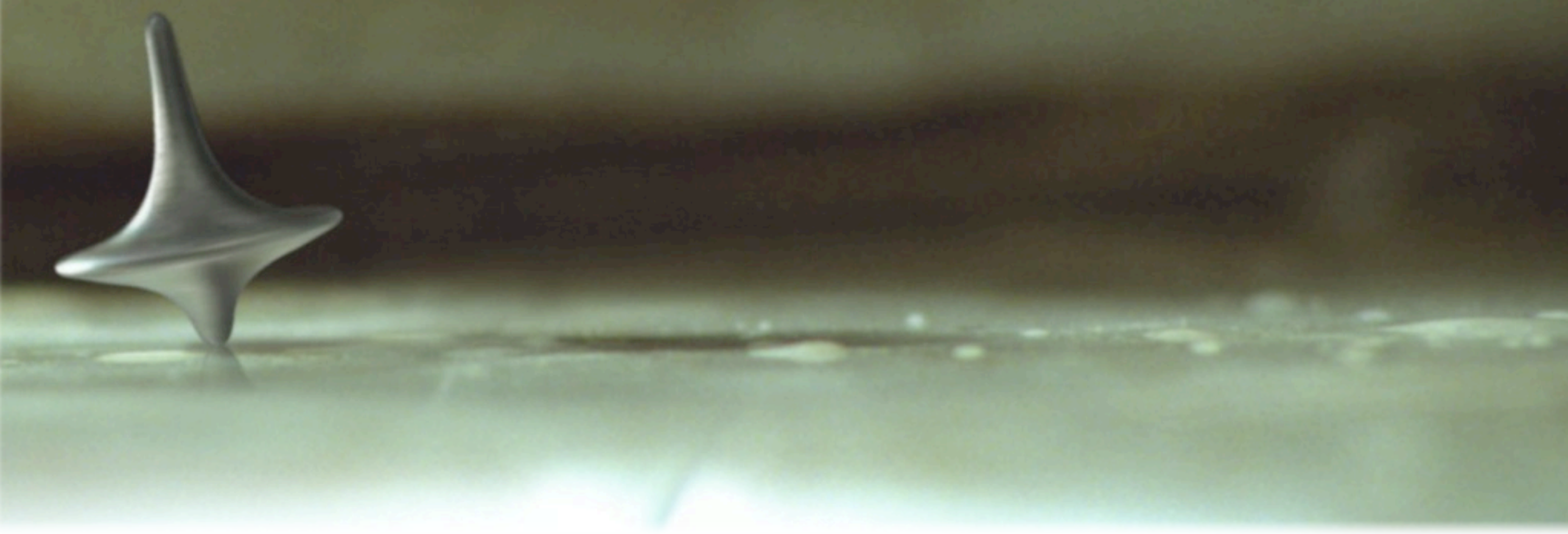
"Any unit of information that is subject to variation and selection *must* produce design"

(Susan Blackmore)

Memes

(Richard Dawkins)

A meme is "an idea [...] that spreads from person to person within a culture." A meme acts as a unit for carrying cultural ideas, symbols or practices, which can be transmitted from one mind to another through writing, speech, [...] or other imitable phenomena. (Wikipedia)



Dom Cobb: What is the most resilient parasite? A bacteria? A virus? An intestinal worm?

[...]

An idea. Resilient, highly contagious. Once an idea has taken hold of the brain, it's almost impossible to eradicate. [...]

(From the movie "Inception")

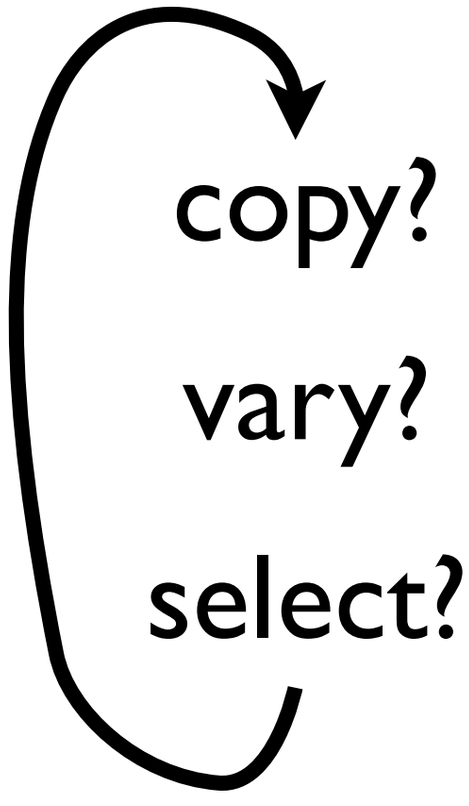
Language feature = a meme

Language = a memeplex

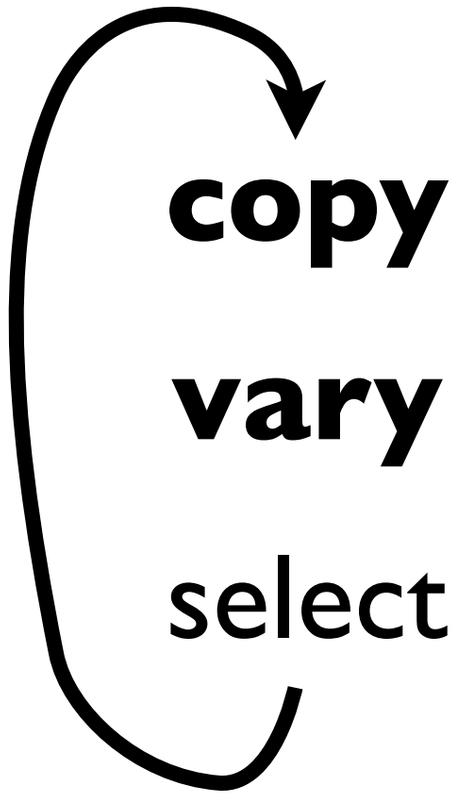
Language feature = a meme

- Literal constants, built-in math operators
- Pattern matching
- Functions, closures
- Static & dynamic typing
- Objects, structs, tuples
- Iterators
- Indentation, delimiter tokens
- Control structures
- Classes
- null, void, nil
- (Lexical) scoping
- Backtracking

Evolution of Language features



Evolution of Language features



Chromosomal crossover

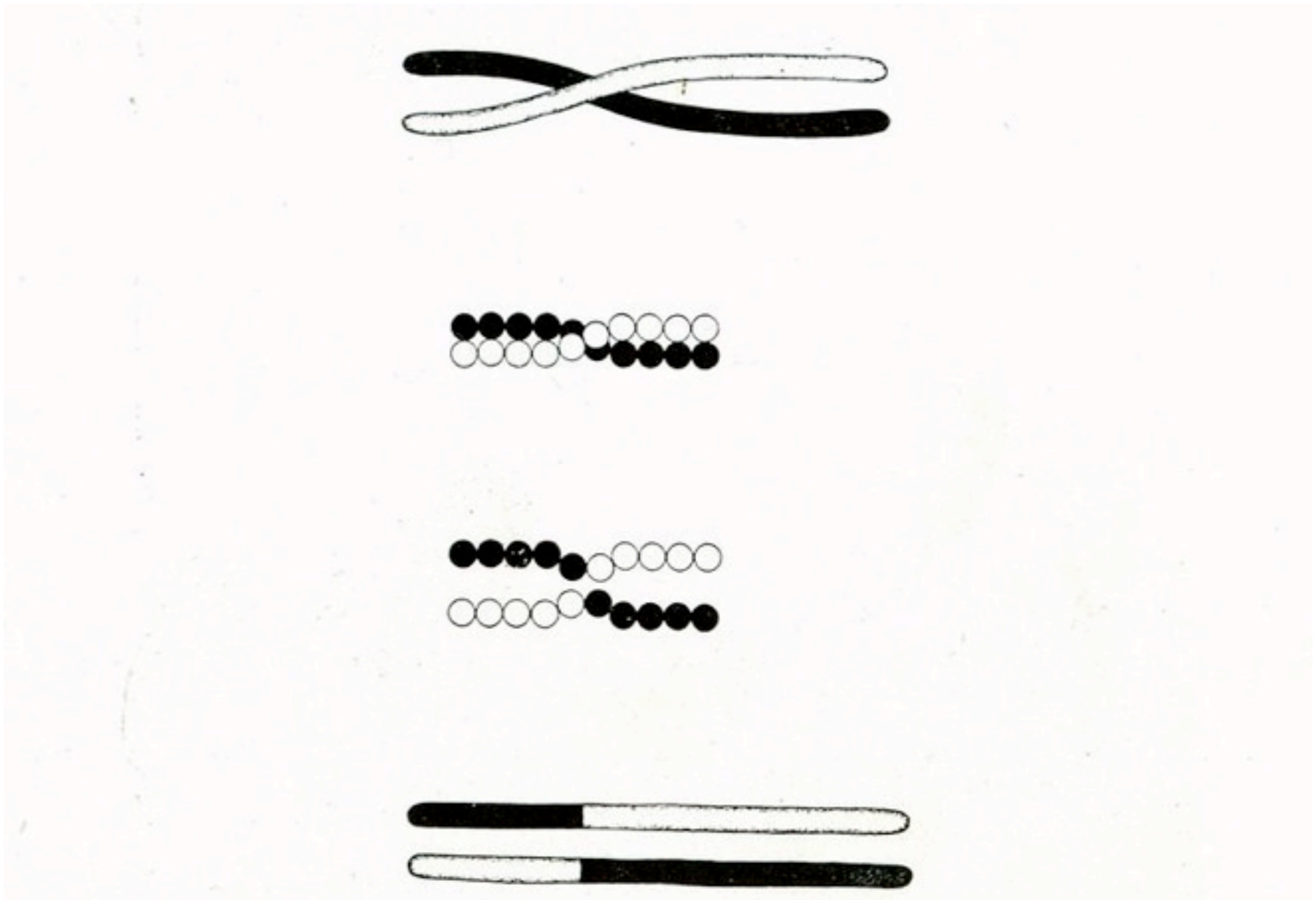
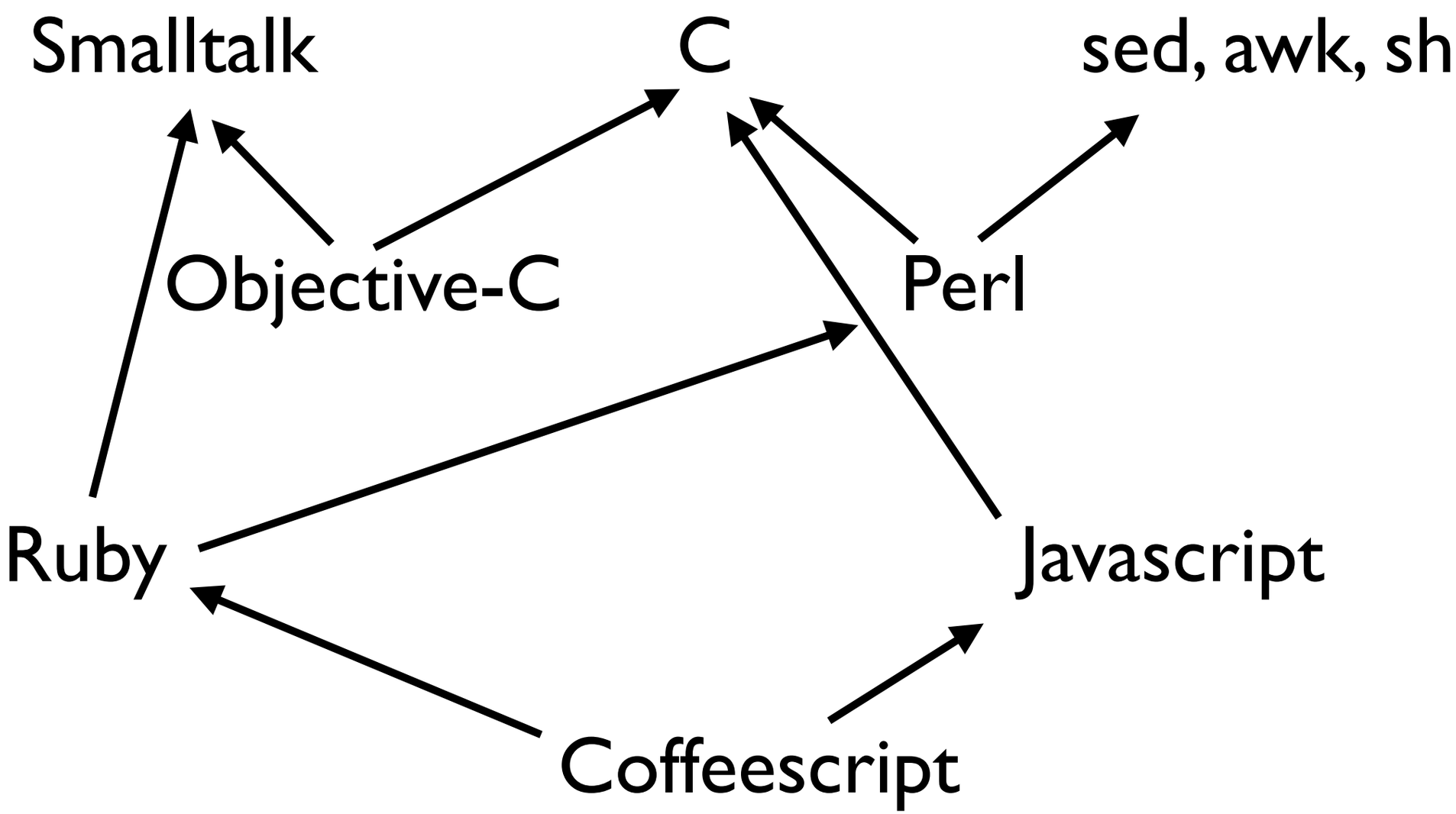


FIG. 64. Scheme to illustrate a method of crossing over of the chromosomes.

(Thomas Hunt Morgan, 1916)

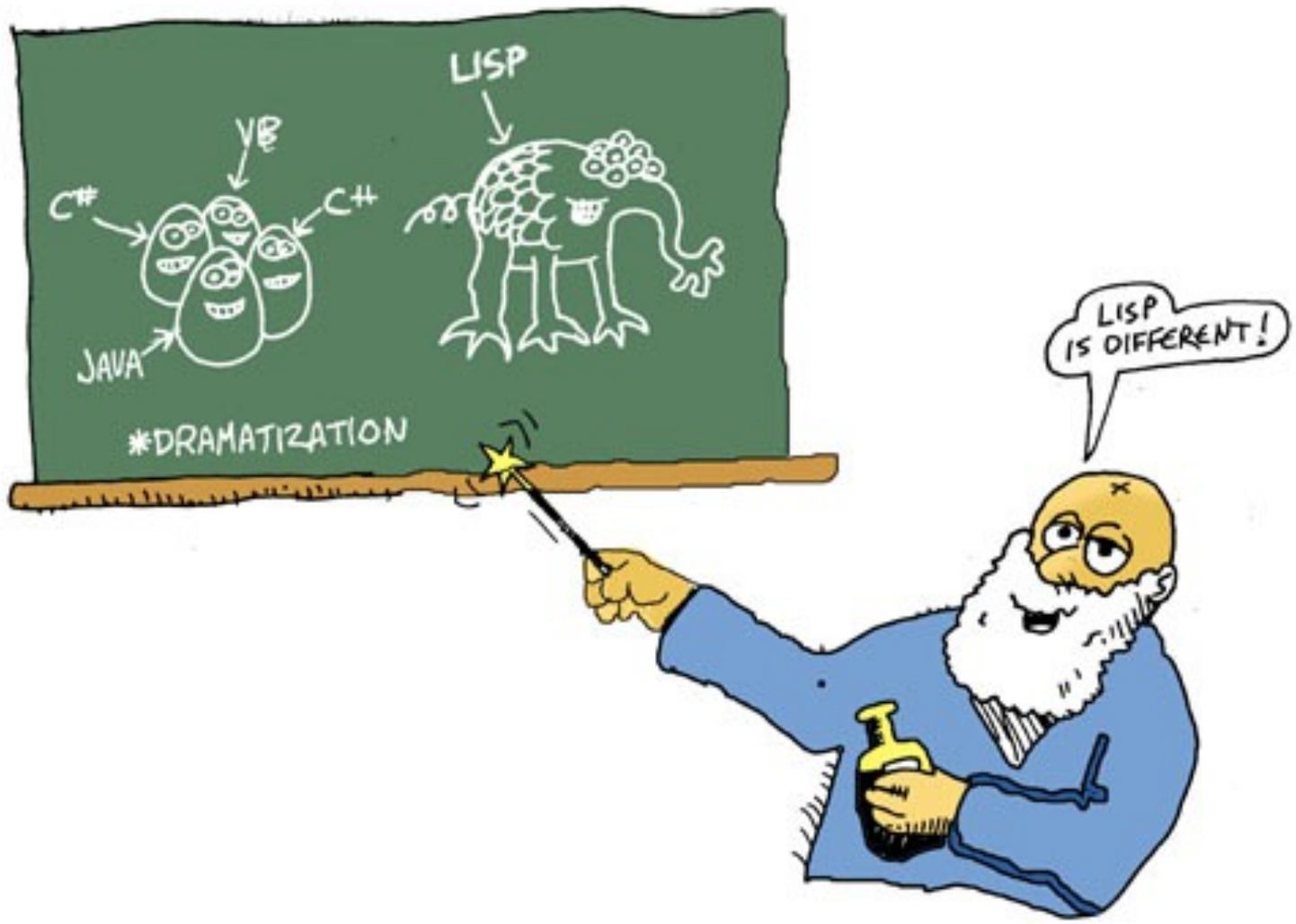
http://en.wikipedia.org/wiki/Chromosomal_crossover

“Lingual” crossover?



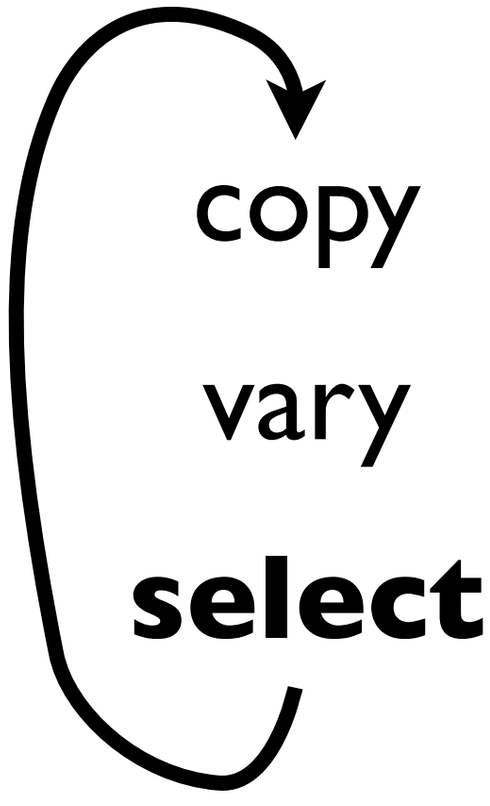
http://en.wikipedia.org/wiki/Timeline_of_programming_languages

“Lingual” crossover?



(Conrad Barski)

Evolution of Language features



Selection

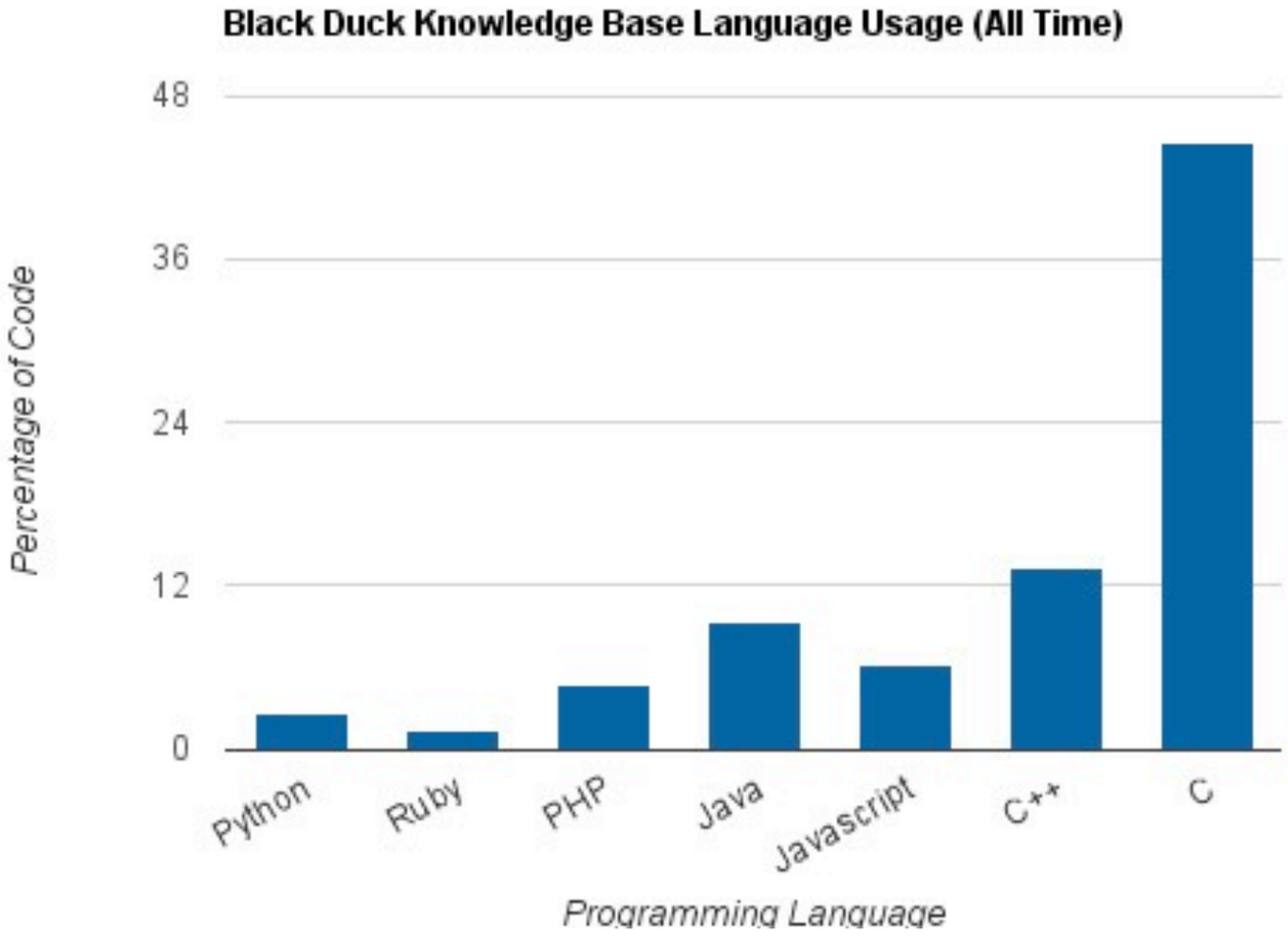
⇩

Survival

⇩

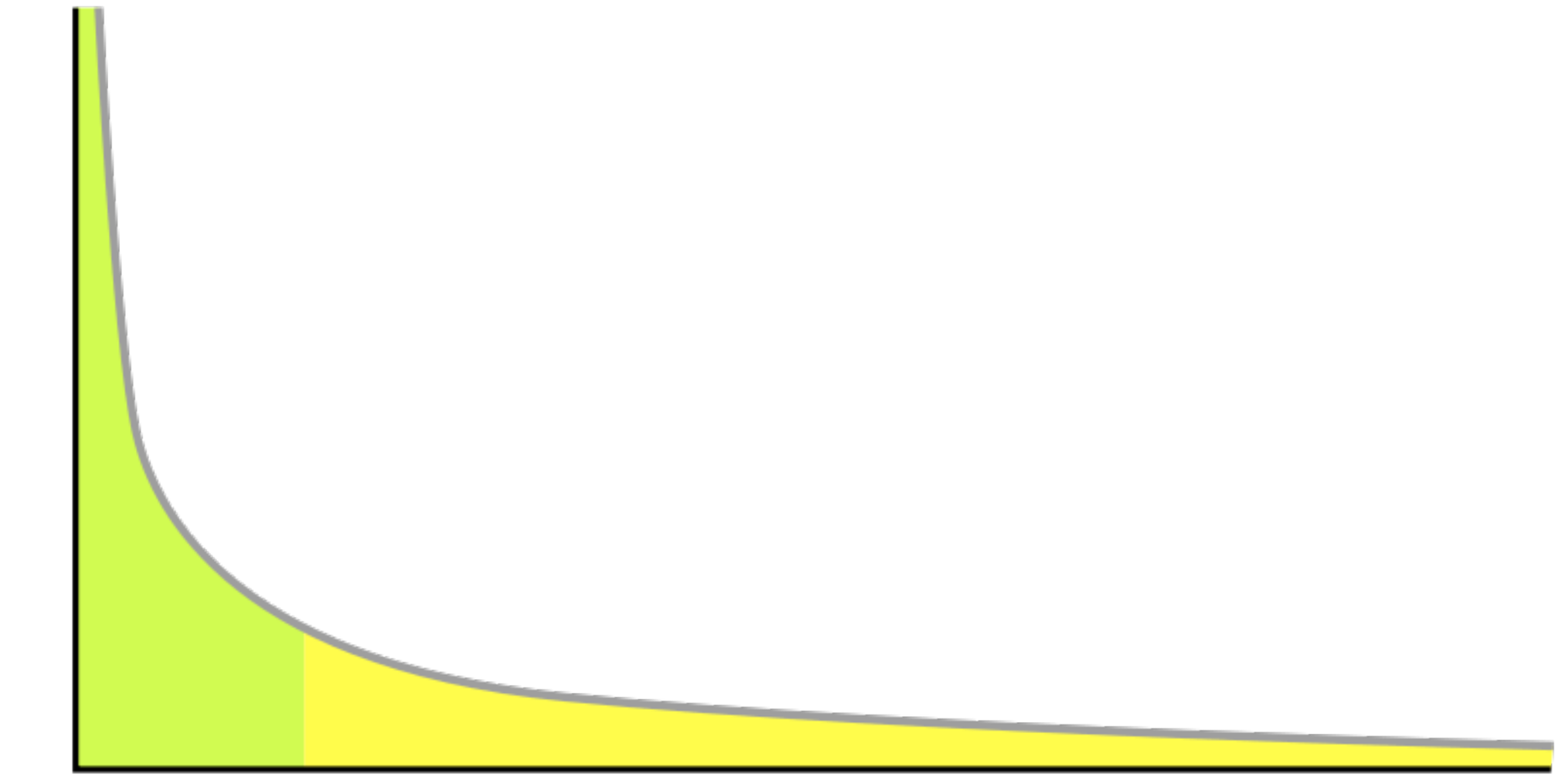
Fitness

Fitness ~ Popularity?

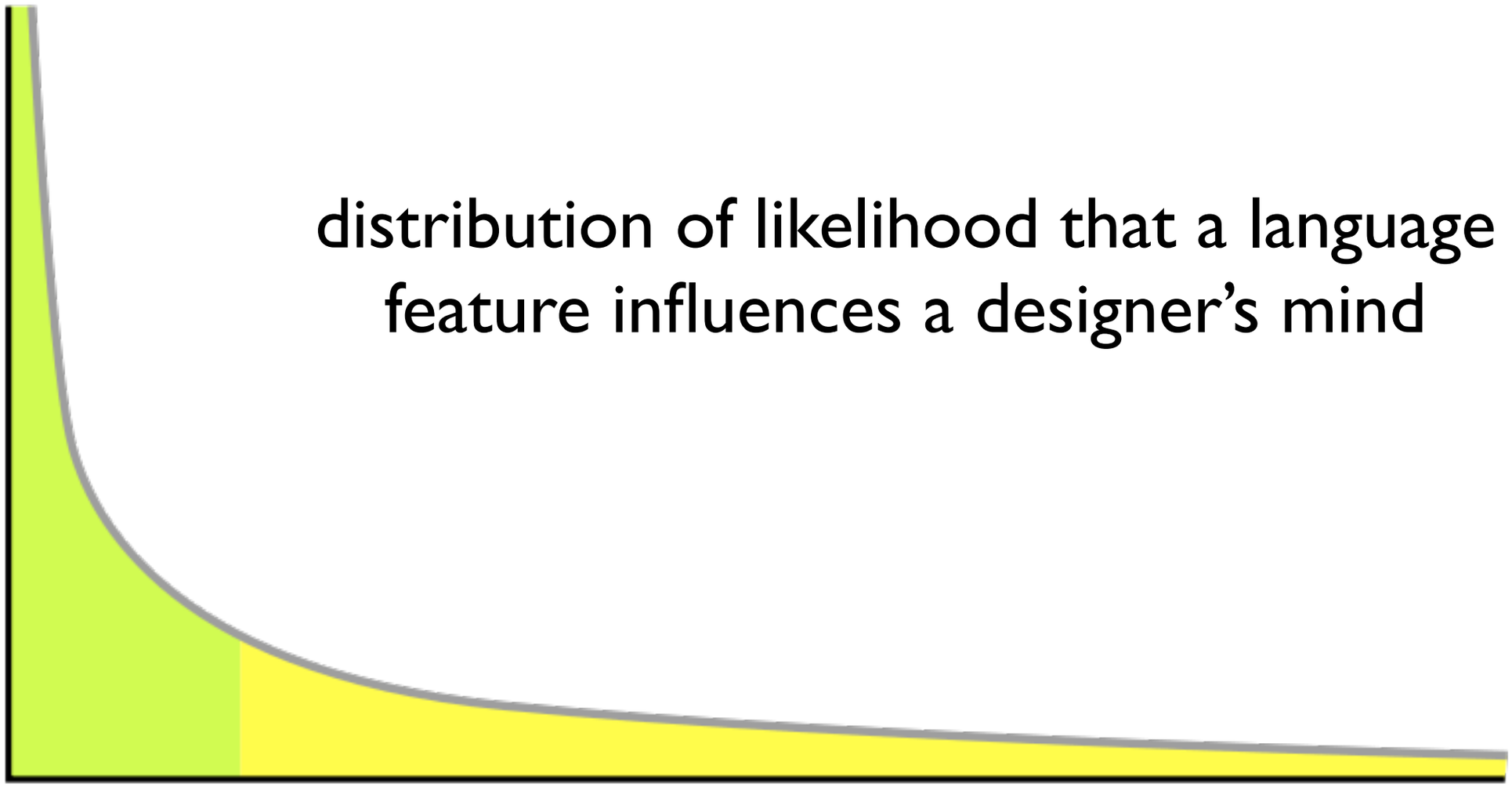


http://redmonk.com/sogrady/2011/04/04/changes-programming-languages/ *(April 2011)*

Popularity: power law distribution with long tail



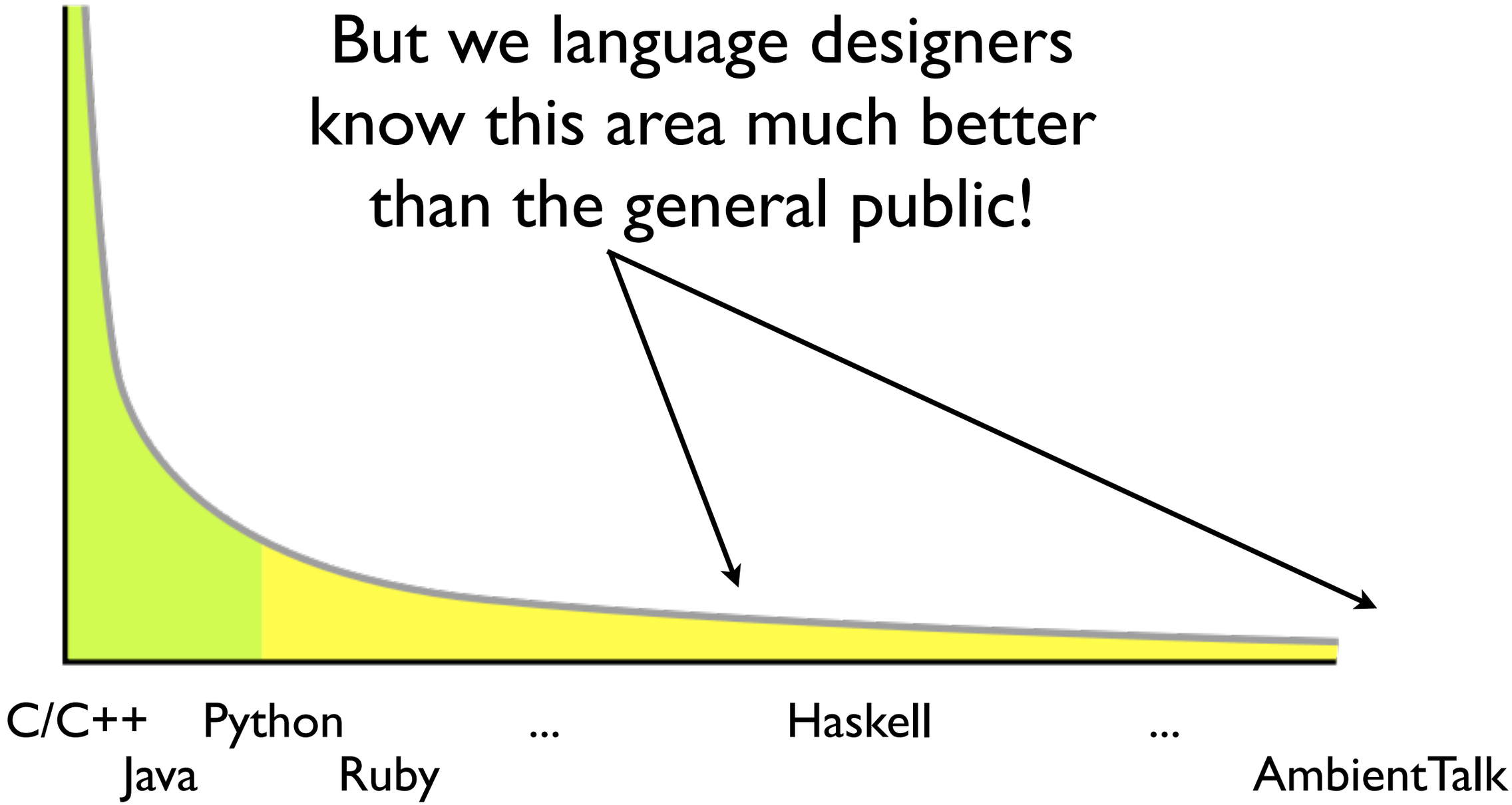
Popularity: power law distribution with long tail



distribution of likelihood that a language feature influences a designer's mind

C/C++ Java Python Ruby ... Haskell ... AmbientTalk

Popularity: power law distribution with long tail



Programming languages are like rock bands

Programming languages are like rock bands

"The first Velvet Underground album only sold 10,000 copies, but everyone who bought it formed a band."

(Brian Eno/Peter Buck)

Programming languages are like rock bands

"The Self environment was only installed on 1000 machines, but everyone who used it became a pldesigner."

(Me)

Popularity ~ Quality?

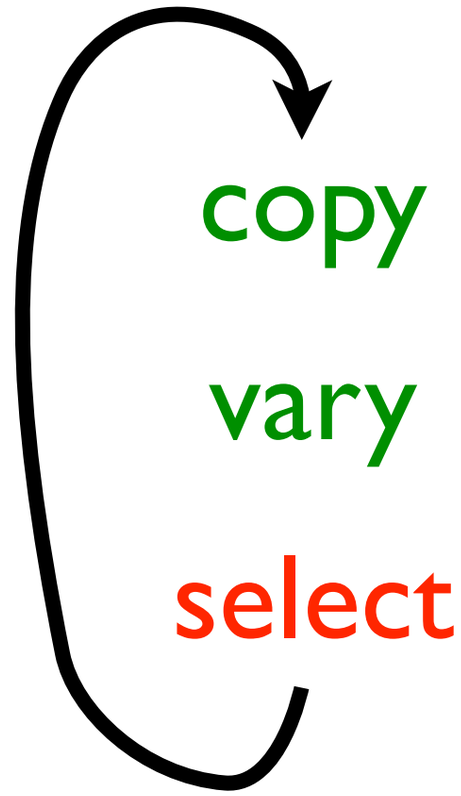
Think Format wars

Think Browser wars

Environmental factors

Differential survival of *ideas*, not languages

Measured by popularity,
FORTRAN and Lisp are nearly extinct

Measured by ideas,
the memes of both are still extremely vibrant and influential

Can we (pldesigners) steer this evolutionary process?