Evolution in Programming Language Design?

Tom Van Cutsem
The Evolution of Language
The Evolution of Language

\[ 2x \quad (\text{Descartes}) \]
The Evolution of Language

\[ 2x \quad (\text{Descartes}) \]

\[ \lambda x. 2x \quad (\text{Church}) \]
The Evolution of Language

\[ 2x \quad \text{(Descartes)} \]

\[ \lambda x. 2x \quad \text{(Church)} \]

\[ \text{(LAMBDA (X) (* 2 X))} \quad \text{(McCarthy)} \]
The Evolution of Language

\[ 2x \quad (\text{Descartes}) \]

\[ \lambda x. 2x \quad (\text{Church}) \]

\[ (\text{LAMBDA (X) (* 2 X)}) \quad (\text{McCarthy}) \]

\[ \text{<?xml version="1.0"?>} \]
\[ <\text{LAMBDA-TERM}> \]
\[ <\text{VAR-LIST}> \]
\[ \quad <\text{VAR}>X</\text{VAR}> \]
\[ <\text{/VAR-LIST}> \]
\[ <\text{EXPR}> \]
\[ \quad <\text{APPLICATION}> \]
\[ \quad \quad <\text{EXPR}><\text{CONST}>*</\text{CONST}></\text{EXPR}> \]
\[ \quad \quad <\text{ARGUMENT-LIST}> \]
\[ \quad \quad \quad <\text{EXPR}><\text{CONST}>2</\text{CONST}></\text{EXPR}> \]
\[ \quad \quad \quad <\text{EXPR}><\text{VAR}>X</\text{VAR}></\text{EXPR}> \]
\[ \quad \quad <\text{/ARGUMENT-LIST}> \]
\[ \quad <\text{/APPLICATION}> \]
\[ <\text{/EXPR}> \]
\[ <\text{/LAMBDA-TERM}> \]
Language design: creation or evolution?
Higgins plot
Scheme Extended Family

T

Algol 60

COMIT

SNOBOL

METEOR

CONVERT

LOGO

PLASMA

Microplanner

Planner

Muddle

Conniver

Scheme

Scheme 311

DAISY,

Scheme 311,

Scheme 84

CLTL1

IEEE

CScheme

Chez

Revised Scheme

Revised 2 Scheme

Revised 3 Scheme

Revised 4 Scheme

Ny

Massachusetts

Europe

Japan

CA

‘60s

‘70s

‘80s

‘90s

MacScheme

PC Scheme

Chez Scheme

IEEE Scheme

Biological classification

(http://en.wikipedia.org/wiki/Biological_classification)
The principal programming paradigms

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

Data structures only
Turing equivalent

Observable nondeterminism? Yes No

More declarative
Less declarative

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

.copy
 vary
 select
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)
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. [...]
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

copy?  
vary?  
select?
Evolution of Language features

copy
vary
select
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?

Smalltalk → Objective-C → Coffeeescript → Javascript → Perl → sed, awk, sh

http://en.wikipedia.org/wiki/Timeline_of_programming_languages
“Lingual” crossover?
Evolution of Language features

copy
vary
select
Selection

↓

Survival

↓

Fitness
Fitness ~ Popularity?

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
Popularity: power law distribution with long tail

But we language designers know this area much better than the general public!
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?

copy
vary
select