



ooErlang: A programmer-friendly approach to OOP in Erlang

Emiliano Firmino
emiliano.firmino@gmail.com

Who I am?

- Current Maintainer of ooErlang
- Software Developer @ INDT
- Master's Student in Electrical Engineering @ UFPE
- Degree in Computer Engineering @ UEA

Who I am?

- Programming Language:
 - C++, C#, Erlang, Java, JavaScript
- Experience:
 - Programming Languages
 - Distributed Systems
 - Mobile
 - Multimedia

Where I come from?



- From Manaus (Heart of the Amazon)

Where I come from?



- Living in Recife (Brazilian Venice)

Where I come from?



- Just 1759.21 Miles apart

How I get there?



- UEA - State University of Amazonas (Manaus, 2001)
- UFPE - Federal University of Pernambuco (Recife, 1946*)
- UFPE-UEA Inter University Doctoral Program
- ooErlang is one of the doctoral thesis from this program

Who worked in this Project?

- 2 Researchers Supervisor:
 - Prof. Rafael Lins (UFPE) [current supervisor]
 - Prof. Francisco Jr. (UFPE)
- 1 Ph.D. Student:
 - Prof. Jucimar Jr. (UEA) [former supervisor]
- 3 Graduate Students (UEA):
 - Me, Daniel Aquino, Rodrigo Bernadino,

Agenda

1. Background
2. Our approach
3. ooErlang 101
4. Implementation
5. Previous Works
6. Future Work
7. Questions?

Background

Shocking Fact:

Objected Oriented Programming (OOP) is popular!

Background: TIOBE

May 2015	May 2014	Change	Programming Language	Ratings	Change
1	2	▲	Java	16.869%	-0.04%
2	1	▼	C ←	16.847%	-0.08%
3	4	▲	C++	7.875%	+1.89%
4	3	▼	Objective-C	5.393%	-6.40%
5	6	▲	C#	5.264%	+1.52%
6	8	▲	Python	3.725%	+0.67%
7	9	▲	JavaScript	3.127%	+1.34%
8	11	▲	Visual Basic .NET	2.968%	+1.70%
9	7	▼	PHP	2.720%	-0.67%
10	-	▲	Visual Basic	1.893%	+1.89%

TIOBE Index: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Background: RedMonk



- Top 10:
1. JavaScript
 2. Java
 3. Python
 4. C#
 5. C++
 6. Ruby
 7. CSS
 8. C
 9. Objective-C
 10. Perl

RedMonk: <http://redmonk.com/sogrady/2015/01/14/language-rankings-1-15/>

Background: PYPL

Rank	Change	Language	Share	Trend
1		Java	24.4 %	-0.9 %
2		PHP	11.5 %	-1.3 %
3		Python	10.9 %	+1.5 %
4		C#	9.1 %	-0.6 %
5	Only C	C++	7.8 %	-0.1 %
6	Not Support OOP	C	7.2 %	-0.2 %
7		Javascript	7.1 %	-0.5 %
8		Objective-C	5.5 %	-0.4 %
9		Matlab	3.0 %	+0.4 %
10	↑↑	R	2.8 %	+0.7 %

- PYPL: <http://pypl.github.io/PYPL.html>

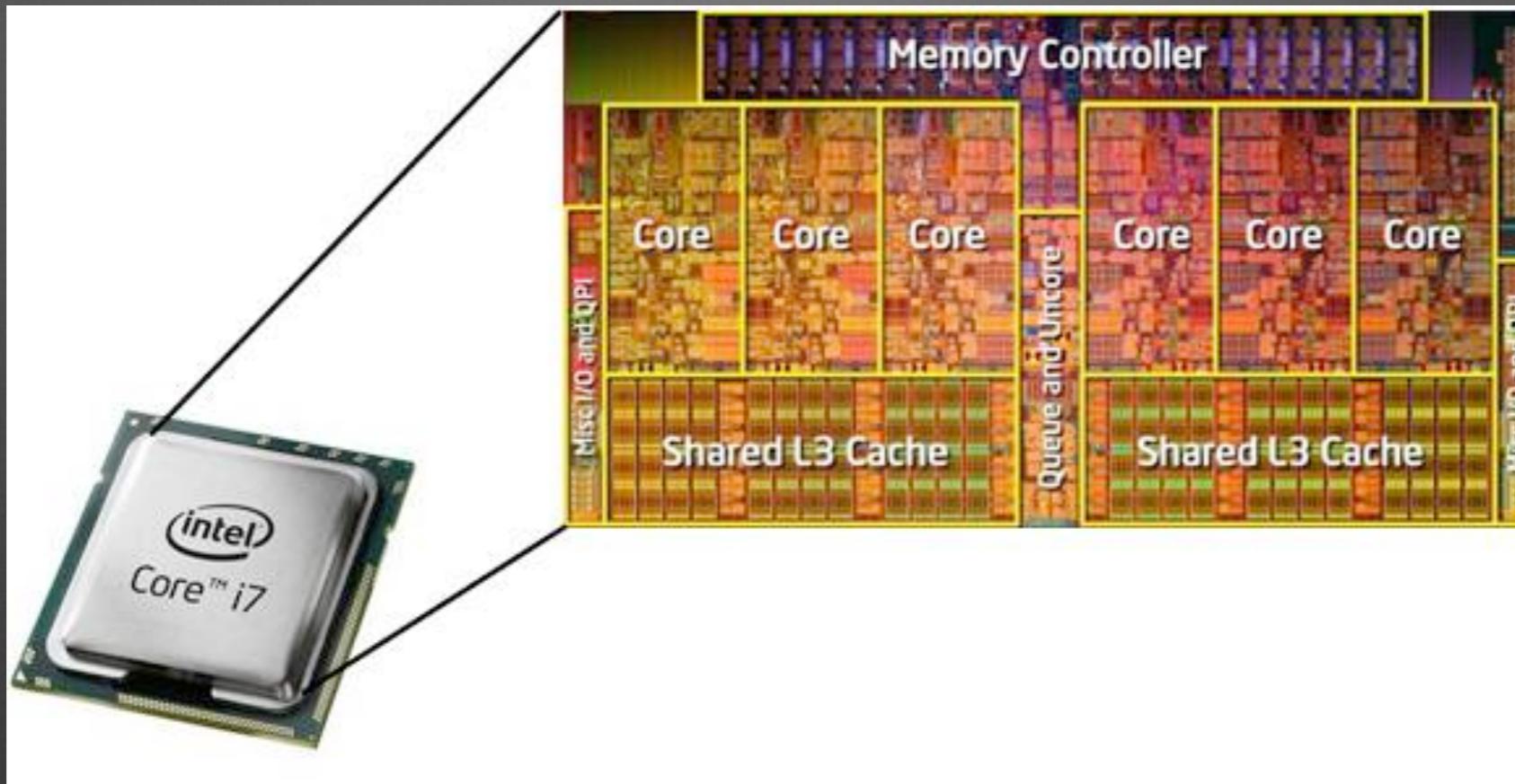
Background

- Most programmers have experience with OOP
- Most popular programming languages support OOP
(With Exception of C, but there is C++ or Object-C)
- OOP is extensively used in the software industry
- Part of many Computer Science or Computer Engineering University Undergraduate Course

Background

But have flaws

Background: Flaw



- Multicore Programming is Hard
- Communication is Hard
- Error Prone in Many OOP Languages

Background

Not in Erlang

Background: Erlang

- Everything is a Process
- Processes are cheap
- Process shares nothing
- Process communicate through explicit messages

Background: Erlang

- How much cheap?
 - Java: ~200us
 - C Pthread: ~50us
 - Erlang: ~15us

Background: Erlang

- TIOBE Index: #36
- Alien Prolog like Syntax
- Not well known in OOP Community

Background

- How help bring more OOP developer to Erlang?

Help them apply knowledge and tools from OOP in Erlang by supporting OOP

Our Approach

ooE ooErlang - Erlang to OO Programmers

[!\[\]\(7750f6b14c60f968a2bc65812f3f92db_img.jpg\)](#)

ooErlang is an object oriented extension to the Erlang programming language. Objects are introduced with a syntax close to Java, making easier its adoption by object-oriented programmers.

[Install](#)
[Hello World!](#)
[Getting started!](#)
[Examples](#)
[Developers](#)
[Publications](#)
[How to contribute](#)
[Roadmap](#)

Contact: ooerlang@gmail.com

[Sign in](#) | [Recent Site Activity](#) | [Report Abuse](#) | [Print Page](#) | Powered By [Google Sites](#)

- Site: <https://sites.google.com/site/ooerlang1>

Our Approach

- Lightweight Extension
- Support to OOP Pillars:
 - Abstract Data Type (e.g.:Class)
 - Inheritance
 - Dynamic Call*

ooErlang 101: Source

The screenshot shows the GitHub repository page for 'ooerlang / ooerlang'. The repository has 347 commits, 1 branch, 0 releases, and 7 contributors. The master branch is selected. The commit history lists various changes, including updates to files like ebnf, examples, include, src, test, .gitignore, .travis.yml, AUTHORS, LICENSE, and Makefile. The latest commit was made a day ago. The right sidebar provides links to Code, Issues, Pull requests, Wiki, Pulse, Graphs, Settings, and clone options via HTTPS, SSH, or Subversion.

File	Description	Time Ago
ebnf	novos arquivos	9 months ago
examples	Removed examples that have own repo	8 months ago
include	Refactor include header	8 months ago
src	Change ooc to be concise	a day ago
test	remove compiled files from repository	9 months ago
.gitignore	Ignore generated .erl in examples/	11 days ago
.travis.yml	Configured to support Travis-CI	8 months ago
AUTHORS	add AUTHORS and LICENSE	9 months ago
LICENSE	add AUTHORS and LICENSE	9 months ago
Makefile	Silent build	11 days ago

- Source: <https://github.com/ooerlang/ooerlang>

ooErlang 101: Installation

```
$ git clone https://github.com/ooerlang/ooerlang.git
$ cd ooerlang
$ make
$ export OOE_HOME=$(pwd)
$ export PATH=$PATH:$OOE_HOME
```

ooErlang 101: How to Use

```
# Class Erlang File: <class>.cerl
```

```
$ ooec *.cerl
```

```
# Call static main() from class
```

```
$ ooe <class_with_static_main>
```

ooErlang 101: Example

Hello World!

ooErlang 101: Class

Person
- FirstName: string
- LastName: string
- Age: number
+ new(string, string, number): Person
+ print(): atom

ooErlang 101: Class

```
// Person.java
```

```
public class Person {  
    private String firstName;  
    private String lastName;  
    private int age;
```

Definition

Attributes

```
public Person(String firstName, String lastName, int age) {  
    this.firstName = firstName;  
    this.lastName = lastName;  
    this.age = age;  
}  
  
private void print() {  
    System.out.printf("%s %s years %d old.\n",  
        this.firstName, this.lastName, this.age);  
}
```

Methods

ooErlang 101: Class

```
// Person.cpp
#include <string>

class Person {
private:
    std::string m(firstName;
    std::string m(lastName;
    int m_age;

public:
    Person(std::string firstName,
           std::string lastName,
           int age) :
        m(firstName),
        m(lastName),
        m_age(age) {
    }
};
```

Definition

Attributes

Methods

ooErlang 101: Class

```
% person.erl
-class(person).
-constructor([new/3]).
-export([print/0]).
```

Definition

attributes.

```
FirstName;
LastName;
Age.
```

Attributes

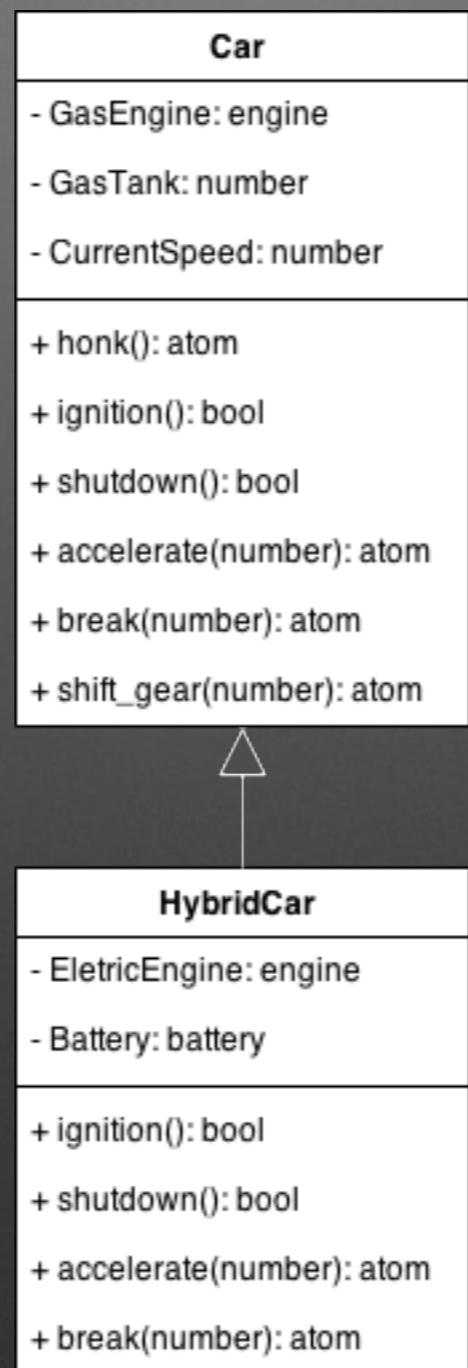
methods.

```
new(FirstName, LastName, Age) ->
    self::FirstName = FirstName,
    self::LastName = LastName,
    self::Age = Age.

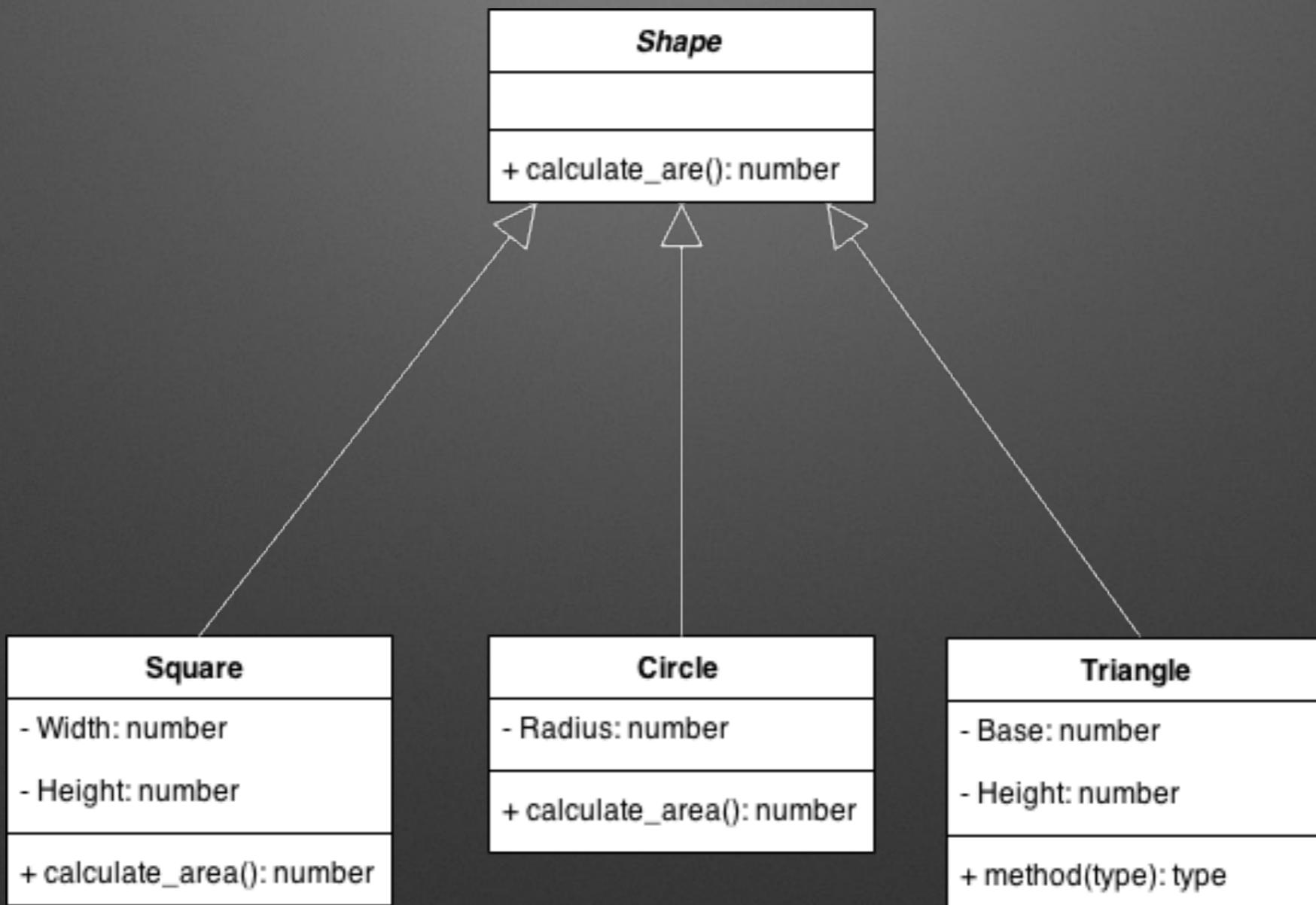
print() ->
    io:format("~s ~s ~w years old.~n",
              [self::FirstName, self::LastName, self::Age]),
ok.
```

Methods

ooErlang 101: Inheritance



ooErlang 101: Interface



Implementation: Object

OOERLANG	ERLANG
Object	{Class, Instance}
Class	atom
Instance	Process (PID)
Attributes	Entries in the Process Dictionary
Object::method(...)	Class:function(Instance, ...)

Implementation: Compiler

1. Read File
2. Extract Tokens
3. Parse Abstract Syntax Tree
4. AST Transformation
5. Code Formater
6. Write to File

Implementation: Compiler

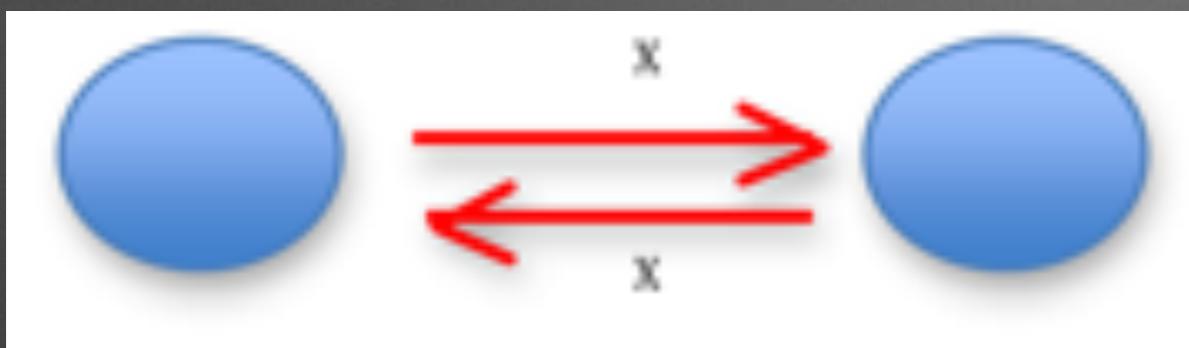
MODULE	DESCRIPTION
ST	Syntax Tree Manipulation
CORE	Compiler Main Engine
OOE	Support Library for Object Handling
OOEC	Compiler API
OOE_PARSE	Initial Version of Syntax Analyser
ALEPPO_PARSE*	Actual Version of Syntax Analyser
OOE_SCAN	Tokenizer
OE_ERRORS	Compilation Error Helper

Implementation: RunTime

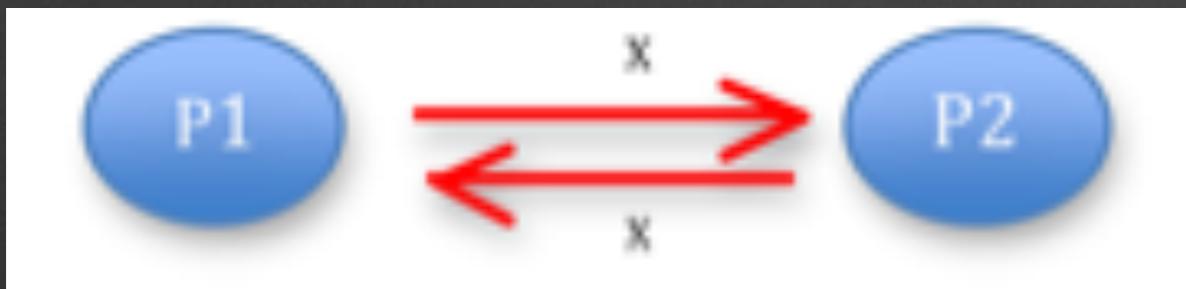
- ooe: Object Management Library
- ooec: Compiler ooErlang Tool
- ooe_utils: Tool to access Tokens and ASTs

Performance (MPI)

- Ping Pong:



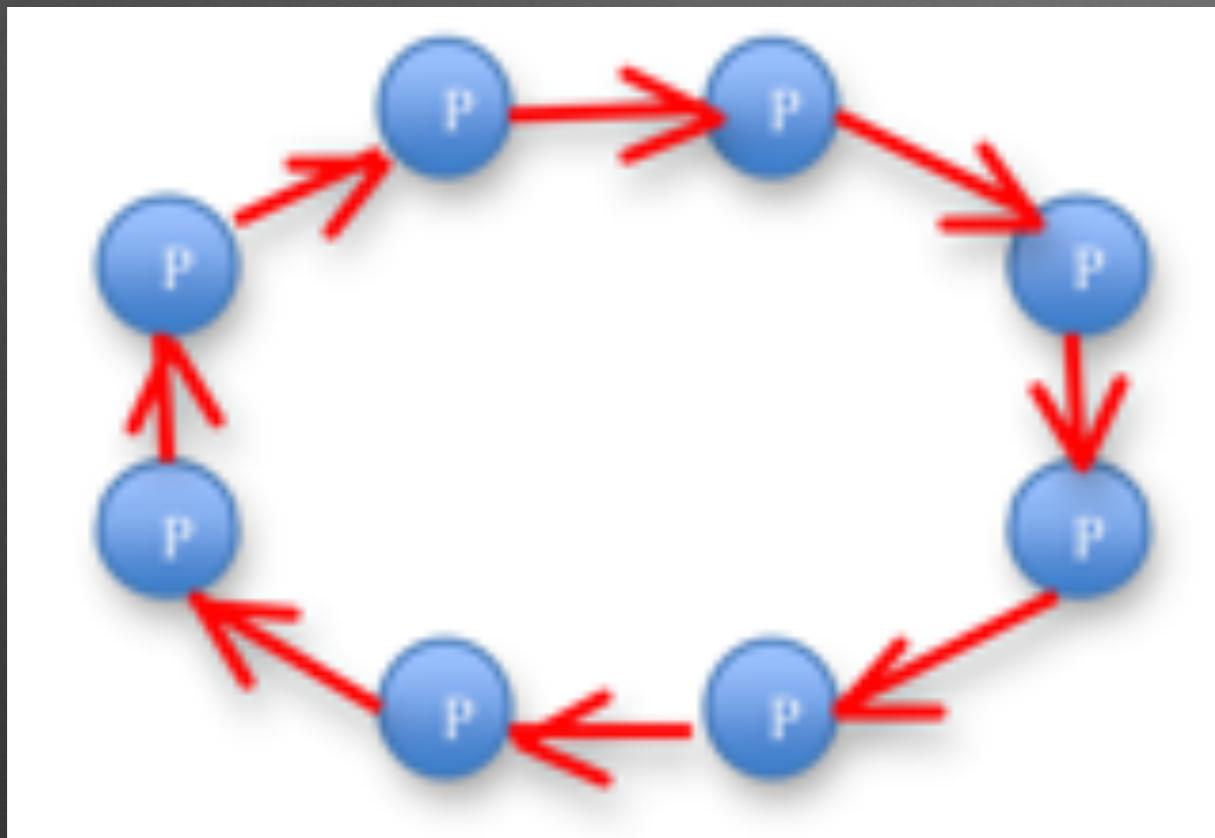
- Ping Pong:



- Source: <https://github.com/ooerlang/mpi-benchmark>

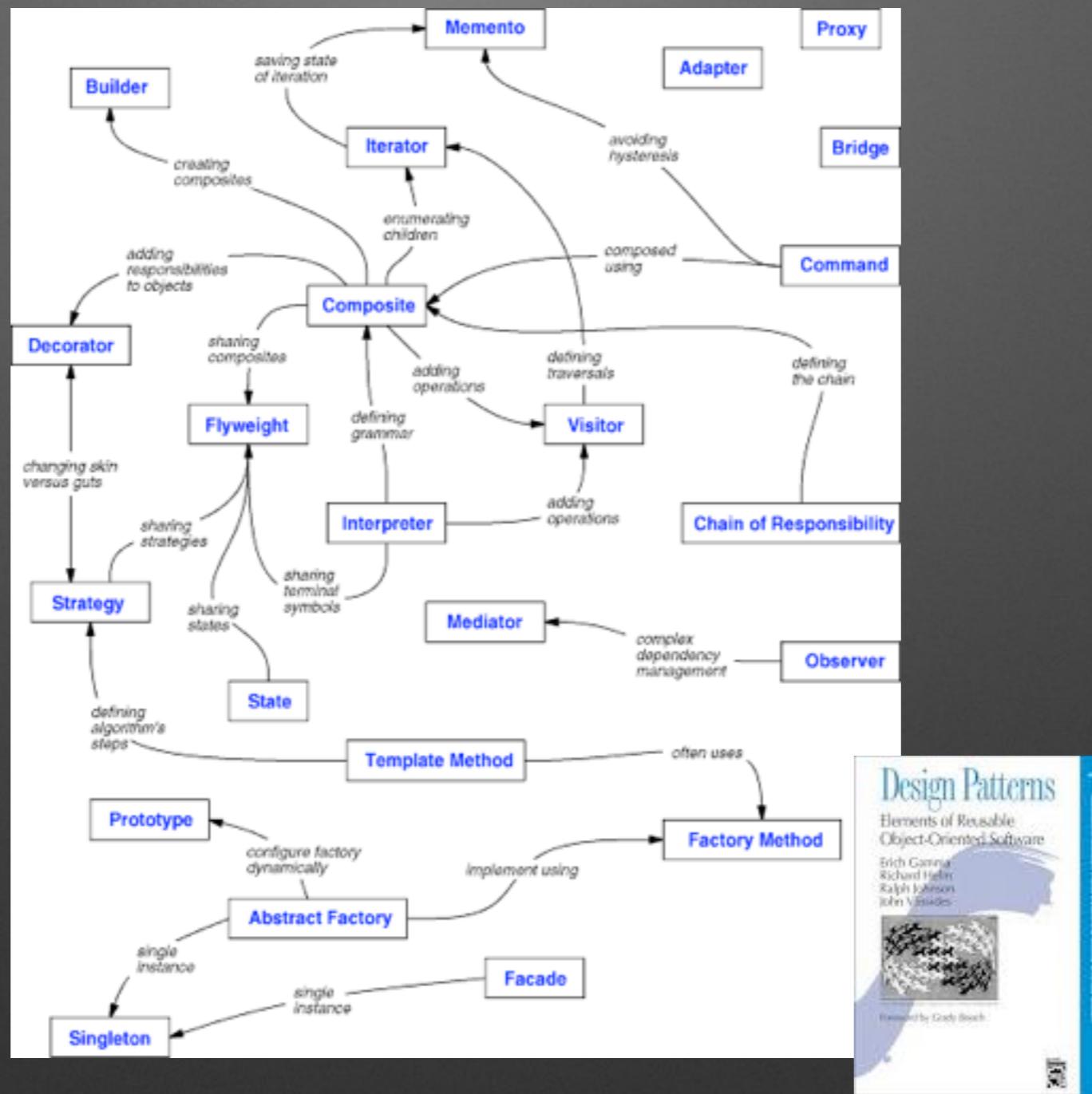
Performance (MPI)

- Send Receive:



- Source: <https://github.com/ooerlang/mpi-benchmark>

Design Patterns



- Source: <https://github.com/ooerlang/ooerlang-design-pattern>

Future Work

- Integrated Developer Environment
- Garbage Collector
- Standard Object Library
- Interactive Terminal
- Bug Fixes

Questions?