



Erlang e-learning

Automatic Assessment of Erlang Code

Simon Thompson
Roberto Aloï

About me

- Roberto Aloï
- roberto-aloi.com
- roberto@erlang-solutions.com
- @robertoaloi

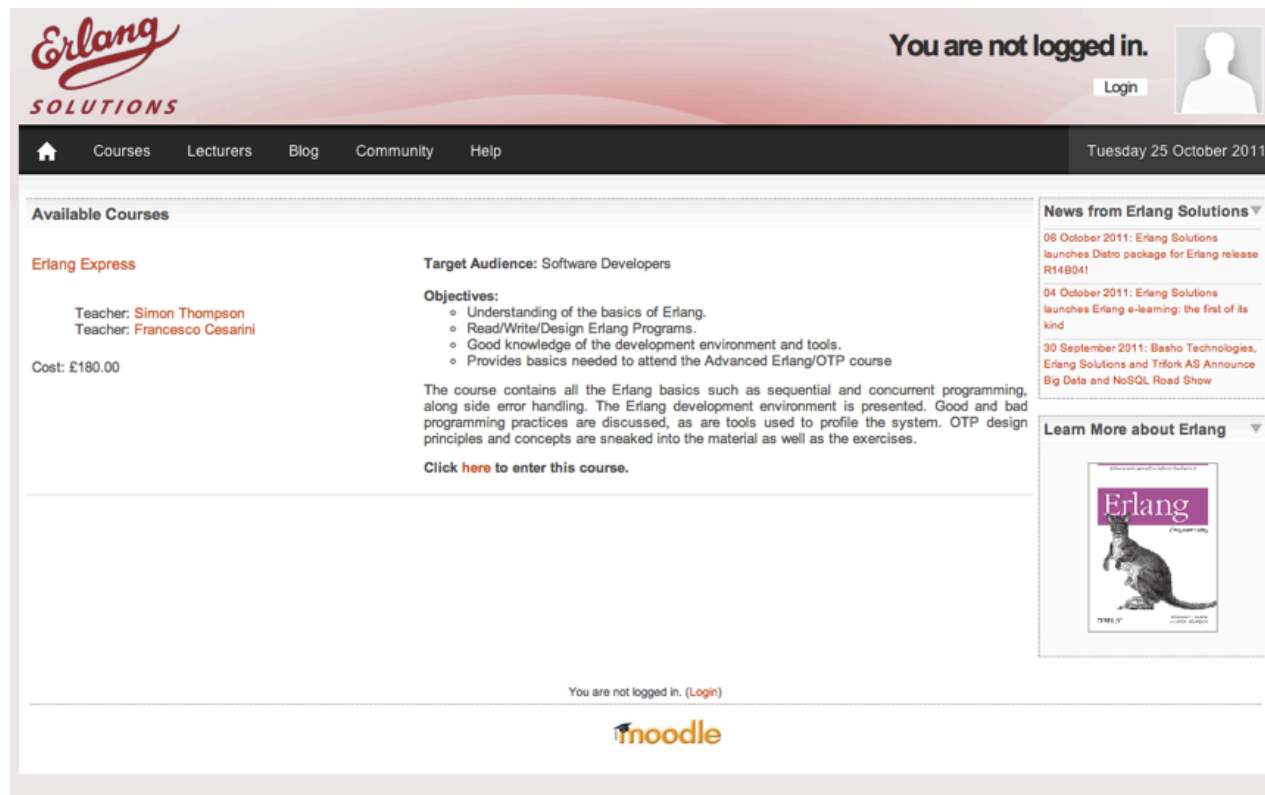
Is this talk a sales pitch?



About Erlang e-learning

- Learn anytime, anywhere
 - No temporal constraints
 - No geographical constraints
 - No economic constraints
- Enhanced face-to-face training
- Asynchronous learning
- Knowledge Transfer Partnership (KTP)

What is Erlang e-learning?



The screenshot shows the Erlang Solutions website interface. At the top, the Erlang Solutions logo is on the left, and a login prompt 'You are not logged in.' with a 'Login' button and a user icon is on the right. A navigation bar below the header contains links for Home, Courses, Lecturers, Blog, Community, and Help, along with the date 'Tuesday 25 October 2011'. The main content area is titled 'Available Courses' and features the 'Erlang Express' course. The course details include the teachers Simon Thompson and Francesco Cesarini, a cost of £180.00, a target audience of software developers, and a list of objectives: understanding Erlang basics, reading/writing/designing Erlang programs, and knowing the development environment. A paragraph describes the course content, covering sequential and concurrent programming, error handling, and OTP design. A link to enter the course is provided. On the right, a 'News from Erlang Solutions' section lists recent updates, and a 'Learn More about Erlang' section features a book cover titled 'Erlang'.

Erlang Solutions

You are not logged in. [Login](#)

[Home](#) [Courses](#) [Lecturers](#) [Blog](#) [Community](#) [Help](#) Tuesday 25 October 2011

Available Courses

Erlang Express

Teacher: [Simon Thompson](#)
Teacher: [Francesco Cesarini](#)

Cost: £180.00

Target Audience: Software Developers

Objectives:

- Understanding of the basics of Erlang.
- Read/Write/Design Erlang Programs.
- Good knowledge of the development environment and tools.
- Provides basics needed to attend the Advanced Erlang/OTP course

The course contains all the Erlang basics such as sequential and concurrent programming, along side error handling. The Erlang development environment is presented. Good and bad programming practices are discussed, as are tools used to profile the system. OTP design principles and concepts are sneaked into the material as well as the exercises.

Click [here](#) to enter this course.


News from Erlang Solutions

06 October 2011: Erlang Solutions launches Distro package for Erlang release R14B04!

04 October 2011: Erlang Solutions launches Erlang e-learning: the first of its kind

30 September 2011: Basho Technologies, Erlang Solutions and Trifork AS Announce Big Data and NoSQL Road Show

Learn More about Erlang



You are not logged in. ([Login](#))

moodle

The Feedback Tool

WHAT?

Automatic assessment of Erlang programming exercises

WHY?

Feedback to students in mass courses

Increase the quality of distance learning

Facebook Puzzles

facebook

Search

Roberto Alai Home

CareersBenefits + PerksLife at FacebookUniversityPuzzles

Do you like puzzles? So do we.


If you love puzzles like we do, become a fan of the new [Puzzle Master Facebook Page](#). Notes are regularly posted to answer questions, explain puzzles, and announce new things. While you're here, try your hand at the following puzzles. The larger the difficulty, the harder it gets (hors d'oeuvres are simple tests to help you out).

Puzzles	Difficulty	Keyword
Hoppity Hop!	Hors d'oeuvre	hoppity
Meep meep!	Hors d'oeuvre	meepmeep
Liar, Liar	Snack	liarliar
Breathalyzer	Snack	breathalyzer
Gattaca	Snack	gattaca
Simon Says	Snack	simonsays
Dance Battle	Snack	dancebattle
It's A Small World	Snack	smallworld
User Bin Crash	Snack	usrbincrash
Rush Hour	Meal	rushhour
Battleship	Meal	battleship

Submission directions

Our puzzle grading robot is currently down for maintenance. We aren't accepting submissions of new solutions at this time, but don't let that stop you from working on the puzzles for fun!


Solved and got hired



David Braginsky
This is my blurb. There are many like it; but this one is mine.



Alok Menghrajani
Solved a few puzzles, and now gets to create new ones for you...



James Leszczenski
Believes creating awesome puzzles is an art form.

Project Euler

Problems

1 2 3 4 5 6 7 8

Go to Problem:

ID	Description / Title	Solved By
1	Add all the natural numbers below one thousand that are multiples of 3 or 5.	175941
2	By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.	145772
3	Find the largest prime factor of a composite number.	105970
4	Find the largest palindrome made from the product of two 3-digit numbers.	98000
5	What is the smallest number divisible by each of the numbers 1 to 20?	110234
6	What is the difference between the sum of the squares and the square of the sums?	112072
7	Find the 10001st prime.	94009
8	Discover the largest product of five consecutive digits in the 1000-digit number.	83207
9	Find the only Pythagorean triplet, $\{a, b, c\}$, for which $a + b + c = 1000$.	83159
10	Calculate the sum of all the primes below two million.	75516
11	What is the greatest product of four adjacent numbers on the same straight line in the 20 by 20 grid?	57370
12	What is the value of the first triangle number to have over five hundred divisors?	50393
13	Find the first ten digits of the sum of one-hundred 50-digit numbers.	56992
14	Find the longest sequence using a starting number under one million.	54997

The Project Euler Flow

Problem 1

05 October 2001

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.
Find the sum of all the multiples of 3 or 5 below 1000.

The Project Euler Flow

```
#include <iostream>

using namespace std;

int main()
{
    const long long NUM = 1000;
    long long res = 0;

    for (int i = NUM - 1; i > 1; i--)
        if ( ( i % 3 == 0 ) || ( i % 5 == 0 ) )
        {
            res += i;
            cout << i << "  ";
        }

    cout << "A soma dos números é: " << res << "\n";

    cout << "Pressione qualquer tecla para continuar";
    cin.get();
    return 0;
}
```

The Project Euler Flow




The Project Euler Flow

31 Oct 2004 06:05 pm

Quote 0

PHP



You almost had it.

As you correctly pointed out...

There are $[100/3]=33$ multiples of 3, but this also includes multiples of 12 and 21. Similarly there $[100/4]=25$ multiples of 4, but these include multiples of 12 and 28. And there are $[100/7]=14$ multiples of 7, which includes multiples of 21 and 28.

So $33+25+14=72$ includes all multiples of 3, 4, and 7, but it includes multiples of 12, 21, and 28 twice.

As $[100/12]=8$, $[100/21]=4$, and $[100/28]=3$, and $8+4+3=15$, we get $72-15=57$.


However, the multiples of 3, 4, and 7, also contain multiples of $3*4*7=84$; in fact there is only one. So each of the multiples of 3, 4, and 7 contains 84 (three times in total), and each of the multiples of 12, 21, and 28 contains 84 (three times in total). So subtracting 15 from 72 completely removes 84, and we need to add it back in. That is, there are $72-15+1=58$ numbers that ARE multiples of 3, 4, or 7, and we deduce the probability of NOT being a multiple of these numbers would be 42%.

Venn diagrams often help to visualise these types of problems:
<http://mathworld.wolfram.com/VennDiagram.html>

Of course, you could write a simple programme to count the number of numbers that are multiples of 3, 4, or 7; it will return 58.

PHP

Show Code



SUMMATIVE ASSESSMENT

Measure the level of success or proficiency that has been obtained at the end of an instructional unit, by comparing it against some standard or benchmark

vs

FORMATIVE ASSESSMENT

Serving to form something, especially having a profound and lasting influence on a person's development

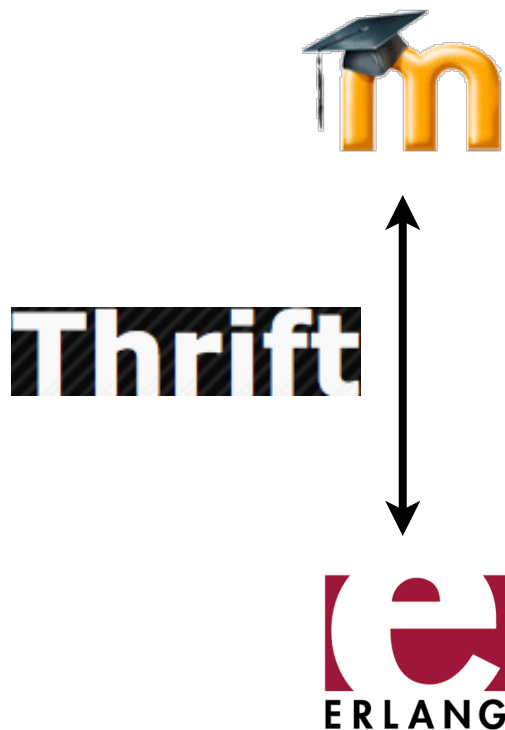
What kind of feedback?

- Syntactic
- Logical
- Stylistic

Let's not re-invent the wheel!

- Compiler
- Dialyzer
- Wrangler
- EUnit
- QuickCheck / PropEr
- Tracer

Behind the Scenes



```
$ft = new Feedbacktool();  
$job = new FTJob();  
[...]  
$job_id = $ft->compute($job);
```

```
compute(Job) ->  
[...]  
JobId.
```

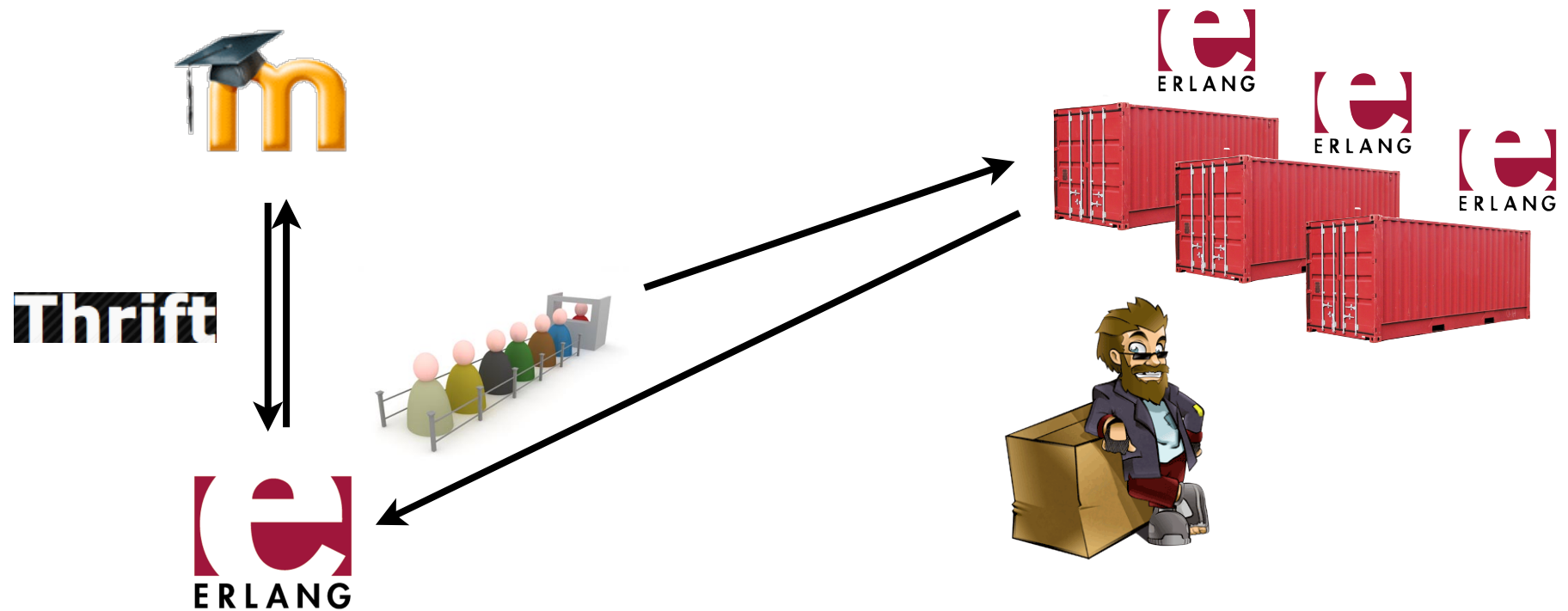
Sand-boxing



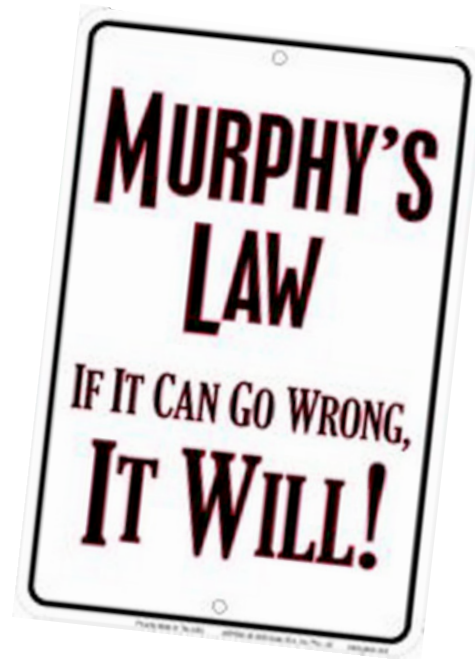
Sand-boxing



Sand-boxing



DEMO TIME



Things fail...

... so do automatic tools

- Manually monitor selection of solutions
 - Identify common error patterns
 - Incrementally improve automatic feedback
- “Ask the teacher” button
- Traditional forums

This was the student's perspective



The Teacher's Perspective

- Invent an exercise (or use an existing one)
- Write a solution for it
- Write tests for your solution
- Plug it into the Feedback Tool
- Configure

Invent an exercise

“Write a *create* module which contains a function named *create* which takes an integer N as an input and returns a list of the format:

$[1, 2, \dots, N-1, N]$

Add to the same module a second function, named *reverse_create*, which returns a list of the format:

$[N, N-1, \dots, 2, 1]$

You may not use the `lists:reverse/1` function.”

Write a Solution for It...

```
-module(create).  
-export([create/1, reverse_create/1]).  
create(0) -> [];  
create(N) ->  
    create_acc(N, []).  
create_acc(0, Result) -> Result;  
create_acc(N, Result) ->  
    create_acc(N - 1, [N|Result]).  
  
reverse_create(0) -> [];  
reverse_create(N) ->  
    reverse_create_acc(N, 1, []).  
  
reverse_create_acc(N, M, Result) when N + 1 == M -> Result;  
reverse_create_acc(N, M, Result) ->  
    reverse_create_acc(N, M + 1, [M|Result]).
```

...And the tests for it.

```
-module(create_tests).  
  
-include_lib("eqc/include/eqc.hrl").  
-include_lib("eunit/include/eunit.hrl").  
  
[...]  
  
prop_create() ->  
  ?FORALL(I, int(),  
    ?IMPLIES(I /= 0,  
      begin  
        N = abs(I),  
        lists:seq(1,N) == create:create(N)  
      end)).
```

Plug it into the Feedback Tool

```
-module(test_create).  
-behaviour(ft_adapter).  
-include_lib("ft_worker/include/ft_lib.hrl").  
-export([test_it/1, compile_required/0]).  
  
test_it(_Options) ->  
    TestSuite = ft_lib:eunit(create_tests, []),  
    [build_message(TestCase) || TestCase <- TestSuite#testsuite.testcases].  
  
compile_required() ->  
    [{"create_tests.erl", []}].  
  
% Internal function  
build_message(TC) ->  
    ft_lib:build_message(TC).
```

Configure

FeedbackCompiler	<input checked="" type="checkbox"/>
feedback_compiler_filename	<input type="text" value="create.erl"/>
FeedbackDialyzer	<input checked="" type="checkbox"/>
FeedbackTest	<input checked="" type="checkbox"/>
feedback_test_solution_path*	<input type="text" value="test_create.erl"/>
FeedbackWrangler	<input checked="" type="checkbox"/>
feedback_wrangler_long_functions	<input 10"]"="" type="text" value="["/>
feedback_wrangler_calls_to_specific_function	<input type="text" value="[{lists, reverse, 0}]"/>
feedback_wrangler_classify_pattern_match	<input type="text" value="[]"/>
FeedbackVisualizer	<input type="checkbox"/>

The developer's Perspective

(extending the Feedback Tool)

- Implement a plugin back-end
- Implement a plugin front-end

Implement a plugin back-end

```
-module(ftw_plugin).  
  
-export([behaviour_info/1]).  
  
behaviour_info(callbacks) ->  
    [{do, 2}, {stop, 1}];  
behaviour_info(_) ->  
    undefined.
```

Implement a plugin back-end

```
-module(feedback_custom).  
  
-behaviour(ftw_plugin).  
  
-include_lib("ft_util/include/feedbacktool_types.hrl").  
  
-export([do/2, stop/1]).  
  
-spec do([string()], [#fTOption{}]) ->  
    {ok, #fTFeedback{}}.  
do(Files, Options) ->  
    Messages = messages(),  
    Status = true,  
    {ok, #fTFeedback{status = Status, messages = Messages}}.  
  
-spec stop(term()) -> ok.  
stop(_Reason) -> ok.
```

Implement a plugin front-end

```
<?php
require_once("path/to/feedback_tool.php");

class FeedbackCustom
{
    private $code = '';

    public function FeedbackCustom($options = array()) {
    }

    public function render($message) {
        $this->code .= FeedbackTool::render($message);
    }

    public function terminate() {
    }

    public function getCode() {
        return $this->code;
    }

    public function supportedOptions() {
        return array();
    }
}
```

The Wrangler Example

```
test() ->  
    v = unnecessary_match,  
    v.
```

The Wrangler Example

```
unnecessary_match(SearchPaths) ->
  Funs=?FULL_TD_TU([?COLLECT(?T("Body@@, V@=Expr@, V@"),
                               {_File@,
                                wrangler_misc:start_end_loc(lists:nthtail(length(Body@@),
                                   _This@))},
                               api_refac:type(V@) == variable andalso
                               length(api_refac:var_refs(V@))==1)],
                   SearchPaths),
  Res = [begin
    Msg = lists:flatten(
      io_lib:format("This is an unnecessary match
                    expression at line ~p.",
                    [Ln])),
    {Msg, [{file, File},
          {line, integer_to_list(Ln)}}}
  end || {File, [{Ln, _}, _]}<-Funs],
  {ok, Res}.
```

Collect **info** from each **template** meeting the **condition**

Give a **message** for each **instance**

The Wrangler Example

```
-module(feedback_wrangler).  
  
-behaviour(ftw_plugin).  
  
-include_lib("ft_util/include/feedbacktool_types.hrl").  
  
-export([do/2, ...]).  
  
do(SearchPaths, Options) ->  
    api_wrangler:start(),  
    ...  
    M = inspect_feedback_wrangler:do_code_inspection(SearchPaths, Options),  
    api_wrangler:stop(),  
    {ok, #FTFeedback{status    = true,  
                     crash    = false,  
                     type     = ?MODULE,  
                     messages = build_messages(M)}}.  
  
...  
  
build_message(M) ->  
    #FTFeedbackMessage{  
        ...  
        message = list_to_binary(Msg),  
        options = ...,  
        custom  = ...}.
```

Options = [..., {unnecessary_match, []}, ...].

Future Improvements

- More augmented feedback
- More accurate tests
- Line numbers and visualizer
- Web-based IDE
- Cloud (Travis CI / Moebius)

Wish List

Math module written in Erlang

math.erl

language detected by filename

```
-module(math).  
-export([sum/2]).  
  
sum(A, B) ->  
    A + B.
```

github:gist

Add another file...

Create Private Gist

Create Public Gist

A special thanks to

- Simon Thompson
- Huiqing Li
- Fred Hebert
- Francesca Gangemi
- Roland Karlsson
- Kryzstof Rutka

<http://elearning.erlang-solutions.com>

roberto@erlang-solutions.com