

The Way To Rakudo *



Jonathan Worthington

Italian Perl Workshop 2009

The Way To Rakudo *

OH HAI

The Way To Rakudo *

Who Am I?

- Originally from England
- Now living in Bratislava, Slovakia



The Way To Rakudo *

Who Am I?

- Originally from England
- Now living in Bratislava, Slovakia



About here-ish

The Way To Rakudo *

What Do I Do?

The Way To Rakudo *

What Do I Do?

- Drink beer



The Way To Rakudo *

What Do I Do?

- Drink beer
- Travel

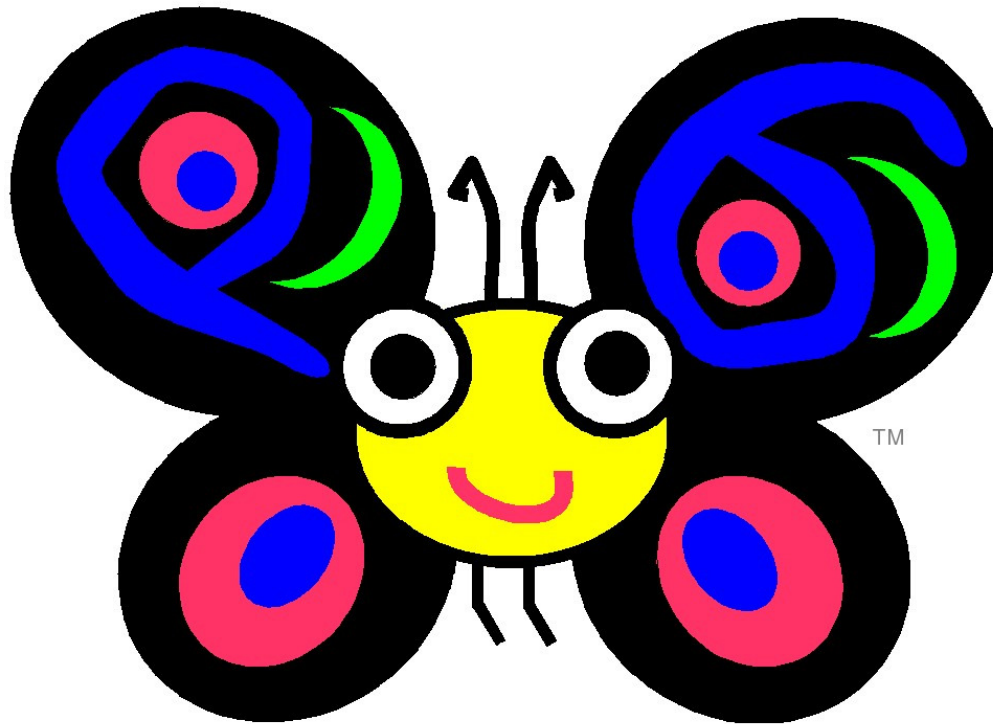


The Way To Rakudo *

What Do I Do?

- Drink beer
- Travel
- In my spare time from this...
 - I'm one of the lead developers of Rakudo
 - Work on implementation of the object model, multiple dispatch, the type system, signatures and other bits
 - Currently also working on support for using Perl 5 modules from Perl 6

Perl 6



What is Perl 6?

- Re-design and update of the Perl language
 - Not syntactically backwards compatible
 - Many changes, many new features, but aims to still "feel like Perl"
- Perl 6 is not an implementation, but rather a language specification
 - Part of it is a written specification
 - Part of it is a "specification test suite"

The Way To Rakudo *

Whirlpool Development

- Perl 6 isn't doing the waterfall model, but instead the "whirlpool"
- Feedback from implementers and users helps refine the language specification
- Pugs – the first implementation, but now unmaintained – provided much feedback
- smop helps the object meta-model evolve
- Rakudo benefits from both of these, and has led to many spec tweaks and clarifications

The Way To Rakudo *



The Way To Rakudo *

What is Rakudo?

- An in-progress implementation of the Perl 6 specification
- Targets the Parrot Virtual Machine at present, but is architected in such a way that it could target more backends in the future
- The most actively developed and most widely used Perl 6 implementation
- Currently passing over 27,500 spectests (about 72.5% of the suite)

The Way To Rakudo *

The Good News

- Very actively developed
- More than one person understands the guts
- Some developers have funding, to help things progress faster
- Growing module eco-system, and the early stages of module installation tools
- So far, 22 monthly development releases have been made

The Way To Rakudo *

The Problem

- Rakudo needs applications and modules written for it
 - They exercise Rakudo in ways that a test suite does not, uncovering bugs
 - They provide essential feedback on where the language specification is weak and/or needs tweaks
- However, few people will use Rakudo if we keep on making "development releases"

The Way To Rakudo *

Therefore...

- In Q2 2010, the Rakudo team will make a major, usable, useful release of Rakudo
- An implementation of a significant subset of the Perl 6 language specification
- Focus on nailing down things that will have a big semantic influence on your program
- Focus on delivering something people will feel is a tool rather than a toy
- The name: Rakudo * (Rakudo Star)

**What does Rakudo
do today?**

The Way To Rakudo *

The Basic Stuff

- Scalars, arrays, hashes...
- All of the basic operators
- Loops
- Conditionals (including chained: $1 < \$x < 10$)
- given/when (like a switch statement, and as seen in 5.10)
- Most of the built-ins that you just expect to be there (grep, map, push, abs, rand, sin...)

The Way To Rakudo *

Object Orientation

- Rakudo now has significant coverage of the object oriented parts of Perl 6
 - Classes (with methods, attributes, inheritance, delegation, etc)
 - Roles (for composition and as mix-ins)
 - Parametric roles
 - Introspection parts of the meta-model
 - Various fancy forms of method dispatch, including deferral

The Way To Rakudo *

Multiple Dispatch

- Choose what to call based on the signature (both arity and types)
- Rakudo has good and mostly complete support for multiple dispatch
- Many primitives in Rakudo already built on top of multiple dispatch (e.g. traits)
- Operator overloading is expressed in terms of multiple dispatch, and working

The Way To Rakudo *

Junctions

- Junctions allow many values to act like one

```
if $dice_roll == 2|4|6 { say "even" }
```

```
while $a & $b > 0 {  
    ...  
}
```

- Auto-threading of junctions works, both for operators and for subs/methods

The Way To Rakudo *

Grammars

- Perl 6 Grammars allow for powerful parsing
- Like regexes on steroids – you now have full grammar capabilities built right into the language
- Rakudo uses this same grammar engine to parse Perl 6, which is great 😊
- But we're far from the standard grammar at the moment ☹️

The Way To Rakudo *

**The way to
Rakudo ***

The Way To Rakudo *

Using STD

- When Rakudo * is released, we aim to be using the standard grammar or something very close to it
- Patrick Michaud is making great progress on a grammar engine re-write that supports a lot of what we're missing; we will integrate it into Rakudo during the next month, and move to a much more STD.pm-ish grammar
- Parsing performance will improve as a result of this too

The Way To Rakudo *

Signature Handling

- On Monday, work on a new signature binder was merged into master
- Gave some great performance wins (method dispatch benchmark ~4x faster, operator dispatch benchmark ~3x times faster)
- Also added some missing features and fixed many bugs
- Way ahead: optimization, more testing and tuning, and building other missing features on top of it (e.g. nested signatures)

The Way To Rakudo *

Laziness

- In Perl 6, lists are lazy by default
- At the moment, Rakudo completely lacks support for lazy lists ☹
- Since having eager vs lazy semantics affects program behaviour, we want to have it in place by Rakudo *
- The one bit of good news: grep, map and more already use gather/take construct, so will automatically become lazy when gather/take is made lazy

The Way To Rakudo *

Module Installation

- Won't have something so rich and complete as CPAN today...
- ...but at least some support for installing modules
- Various efforts are underway on this already
 - The Plumage project is producing a Parrot-level solution
 - Proto has been around for a little while and continues to evolve

The Way To Rakudo *

Perl 5 Integration

- Aim to have at least basic support for using Perl 5 modules from Rakudo
- Blizkost is a project to take the Perl 5 interpreter, embed it and wrap the standard Parrot compiler interface around it
 - Gives Rakudo + other Parrot languages access to the CPAN
- Have had some basic bits working already 😊
- But ignored for the last few weeks...oops 😞

**What will
likely be
missing in
Rakudo *?**

The Way To Rakudo *

Rakudo * Might Not Be Fast

- Focus has been on completeness *first*
- Even with recent improvements, Rakudo is still not particularly fast
- We expect to improve performance between now and Rakudo * (aided by the Parrot team recently producing profiling tools, which will help show up our bottlenecks)
- Rakudo * should be fast enough for tasks that are not runtime performance-critical

The Way To Rakudo *

Concurrency Support Comes Later

- Perl 6 specifies many opportunities for concurrency, and is designed with being able to be efficiently parallelizable in mind
- There's some specification work on various concurrency primitives, but they need fleshing out and real world testing
- Rakudo * will most probably not be able to provide support for concurrency

The Way To Rakudo *

Advanced Data-structures Still To-do

- Rakudo * will likely be missing most, if not all of the following:
 - Compact arrays
 - Compact structures
 - User defined array indexing
 - PDL support

The Way To Rakudo *

Why will these be missing?

- Partly lack of resources
- More importantly, the design still needs input based on real world use cases
- Rakudo * is a necessary step to getting them

Building the paths to Perl 6

The Way To Rakudo *

Making Perl 6 Accessible

- Right now, Rakudo is mostly used by those with some interest in Perl 6 development
- Rakudo * aims to make it more accessible
- It's a little like people visiting a mountain...

The Way To Rakudo *

Climb it yourself...



The Way To Rakudo *

Climb It Yourself

- Only appealing to a relatively small group of people
- Risky...
 - Still quite a few bugs lurking
 - Still kinda slow
- But, when it works out, rewarding – people are already creating some beautiful code using Perl 6
- This is where the Rakudo compiler is today

The Way To Rakudo *

Take a flight...



The Way To Rakudo *

Take A Flight

- Much easier = many more people can do it
- You can enjoy the mountain...
- ...though you know that standing on it would be better still
- This is where Rakudo * is aiming at

The Way To Rakudo *

The train to the top



The Way To Rakudo *

The train to the top

- Very accessible
- Even works for those with a fear of flying (e.g. early adoption)
- You're on the mountain
- But it takes a lot of time and effort to build
- Having had more people visit in other ways helps those doing the building to know where the pitfalls are and deal with the earlier

The Way To Rakudo *

A step on the journey

- Rakudo * isn't all of Perl 6
- However, it is a significant step towards it
- Delivers many of the features and improvements that the Perl 6 project has promised
- Enough of Perl 6 to be useful to many
- But we know it won't be for everyone

The Way To Rakudo *

**Want to help us build
the path to Rakudo *?**



The Way To Rakudo *

How to get involved

- Write modules and applications
 - Report the bugs you encounter
 - If something felt like it was harder work than it should have been, or if there seems to be a corner-case missed, report that too
- The spec test suite always needs work
- For more see: www.perl6.org
- Hang out on the friendly Perl 6 IRC channel: [#perl6](http://irc.freenode.org)

The Way To Rakudo *

Rakudo *

Q2 2010

www.rakudo.org

