Using Contize for Web Applications

Part I: Continuation-based Web Programming
Part II: Contize
Part III: Example Application
Part I: Continuation-based Web Programming

• What is a continuation?
  – “goto where I came from”
  – “coderef for where I came from”
Part I: Continuation-based Web Programming

• What is a continuation?
  – “goto where I came from”
  – “coderef for where I came from”

• Example:

```perl
sub a { print "a";
sub b { print "b";
a();
b();
print " done.";
```
Part I: Continuation-based Web Programming

• What is a continuation?
  - "goto where I came from"
  - "coderef for where I came from"

• Example:

```perl
sub a {
  print "a";
  goto L1;
}
sub b {
  print "b";
  goto L2;
}

a();
L1: b();
L2: print " done."
```
Part I: Continuation-based Web Programming

• What is a continuation?
  – “goto where I came from”
  – “coderef for where I came from”

• Example:

```perl
sub a { print "a";
    goto L1;
}
sub b { print "b";
    goto L2;
}
a();
L1: b();
L2: print " done."
```

```perl
sub a { $next = shift;
    print "a";
    &$next;
}
sub b { $next = shift;
    print "b";
    &$next;
}
a(sub {
    b(sub {
        print " done."
    })
})
```
...Continuations

- First Class Continuations
  - Continuations are built into the language, like 'this' in C++ or 'self' in smalltalk... except referring to the current continuation instead of the current object
  - Languages: Scheme, Ruby, StacklessPython, SML
  - Not Perl! ... well.... except for Coro::Cont :)

● First Class Continuations
Web Programming

• Stateless HTTP / CGI
  - Application must maintain its own state
  - CGI programs are re-started all the time

• Other Things
  - Model-View-Controller
  - Model: Database layer, such as Class::DBI
  - View: Template layer, such as Embperl
  - Controller: Business logic... and state management
Using Continuations

- Output to browser
- Save continuation (to disk)
- Exit program
- ... they fill out form, send result to application ...
- Load continuation (from disk)
- Pick up right where we left off!
The Result

sub addTwo {
    my $a = input("Enter first number");
    my $b = input("Enter second number");
    output("First + Second = ". ($a + $b));
}

Part II: Contize

- So we want continuations in Perl
  - Coro::Cont, XS magic to save stack state, etc
  - Could set up our own HTTP server with a managing agent. Match incoming input with suspended continuation
  - I want CGI through Apache, not my own server
  - Can't make Coro::Cont dump to disk (yet!)
- So what is to be done?
Contize.pm

• The Hack
  – While running a method, cache all method invocations
  – At some point the object suspends (itself), save cache
  – To resume the object, run its code. If a method has already been invoked, return its previous value

• The Worries
  – Too much cached?
  – Too much re-executed, especially DB operations?
Contize.pm

• The Implementation
  – Inheritance isn't enough... no way to catch existing methods
  – Contize wraps the target object, then uses AUTOLOAD to intercept and cache method calls
  – my $obj = new Contize(new Object);
  – Also adds 'suspend' and 'resume' methods

• Current Issues
  – Only works on hash-based objects
  – Ugly wrapper to manage the application
Show me the Code!

- Contize itself
- WebGuess
- (Part III) Panel application

At this point we switched to VIM and Firefox viewing for a demo.

See http://thelackthereof.org/wiki.pl/Contize for more information.