



Virtual Machines for High Level Feature Support

Dan Sugalski
dan@sidhe.org

September 17, 2004



Handing out New Toys

Dan Sugalski
dan@sidhe.org

September 17, 2004

Assumptions

- Languages don't live in a vacuum
- We like interoperability
- Unshared features are of limited use

Important safety tips

- Nothing in this presentation is
 - new
 - completely impossible with current systems
 - implemented in any mainstream systems
- Except exceptions and closures (and only barely)
- And no, neither Lisp nor Smalltalk are really mainstream

Foundations drive design

- Hardware, languages, libraries
- The point you say “I am (or can) going no deeper”
- Defines what’s easy, hard, or nearly impossible
- Defines what’s shareable

VMs play hardware, OS, and system

- Look like a hardware CPU
- Has OS features like security and device management
- Provides lots of libraries
- Provide a list of easy, hard, and nearly impossible things
- Make important things standard

Provide rich base functionality

- Things compiler writers *can't* do
- Things compiler writers don't *want* to do
- Things that can't be meaningfully shared otherwise

Features aren't without costs

- Everything has a price
- Some prices are too high
- Hardware won't always save us
- Things should be done wholeheartedly or not at all
- There is pleasant fallout
- Speaking from experience, don't do all these

Control Flow

- Anonymous closures
- Exceptions
- Co-routines
- Continuations
- Deep recursion
- Massive threading

Introspection and monitoring

- Proper serialization
- Deep freezing
- Ownership chains
- Function insertion
- Notifications

Dispatch

- Multimethod dispatch
- Quantum indeterminacy

Quantum indeterminacy

- Variables with multiple coincident states
- Actions act on all states simultaneously

```
sub foo(param)
```

```
    return param * 4
```

```
A = 3 | 5 | 7
```

```
B = foo(A)
```

```
if (any(B > 18))
```

```
    print "Yes"
```

Interesting data behavior

- Lazy evaluation
- Active data
- Proper operator overloading
- Dynamic type changes

Runtime reconfigurability

- Loading new opcode
- Changing existing opcode functions
- Scoped namespaces