The Real World

Wouter Swierstra

AFP 15-03-07

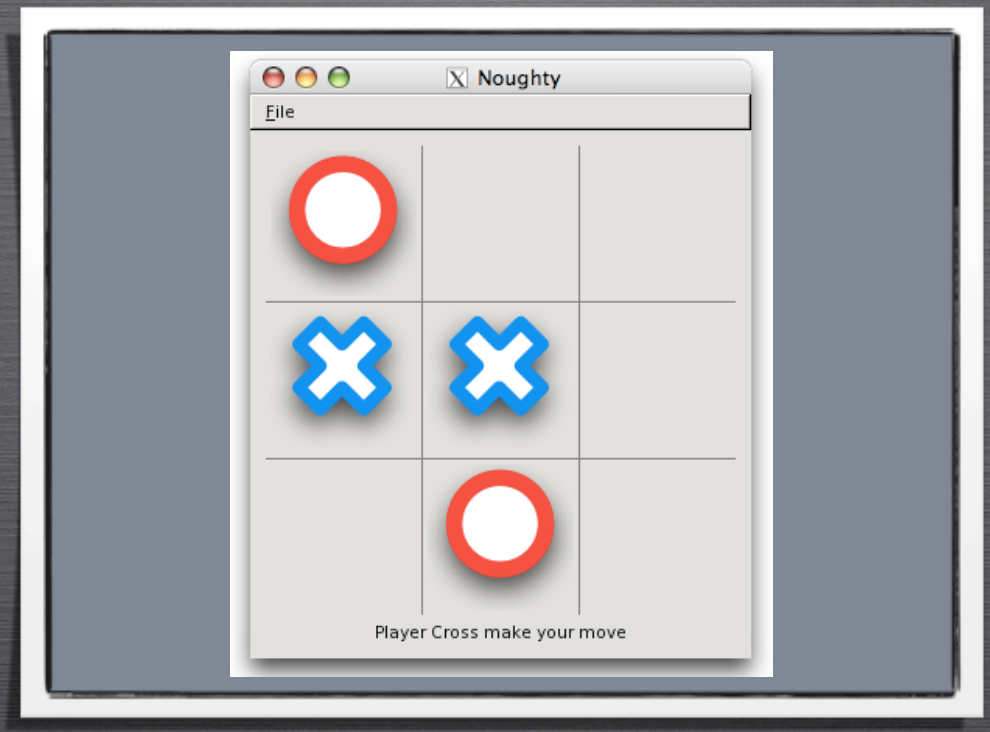
A Common Misconception

Haskell is great for writing beautiful code...

... but cannot be used for real programming!

The Awkward Squad

- Input and output
- Mutable state
- Interfacing to other languages
- Concurrency
- Exceptions
- GUI programming



Noughts and Crosses

Noughts And Crosses

- Completely implemented in Haskell
- About 130 lines of code
  - Core logic: 45 lines
  - Action handlers: 25 lines
  - GUI: 45 lines
- Adding simple AI: 20 lines
- How does it work?

Foreign Function Interface

- Haskell can call C procedures:

```
foreign import ccall
  sin :: Float -> Float
```

- and even call side-effecting functions:

```
foreign import ccall
  drawPoint :: Int -> Int -> IO ()
```

- Dual version foreign export
- Only works for numbers `Int, Float, ...`

Marshalling

- How do you get data to and fro?

```
drawLine :: Point -> Point -> IO ()
```

  - Do it yourself
  - Use a tool

- An ugly one-off investment

wxWidgets

- Popular open source GUI toolkit
- Lots and lots of widgets
- Cross-platform
- Written in C++

wxHaskell

- Use the Foreign Function Interface to call methods from wxWidgets.
- Haskell friendly interface
- Pleasant abstractions
- High-level layout combinators
- Type safe
- Installed here

Hello World

```
import Graphics.UI.WX
-- Fire up the GUI
main = start hello

-- Make a window and a staticText widget
hello = do
  f <- frame []
  message <- staticText f [text := “Hello”]
  return ()
```

Mutable State

- Haskell has mutable state:

```
newIORef :: a -> IO (IORef a)

readIORef :: IORef a -> IO a

writeIORef :: IORef a -> a -> IO ()
```

The Counter Example

```
main = start counter

counter = do
  f <- frame []
  var <- newIORef 0
  display <- staticText f [text := “0”]
  b <- button f [text := “Increment”
       , on command := incr var display]
  set f [layout := ...]

 incr var display = do
  x <- readIORef var
  newX = x + 1
  writeIORef var newX
  set staticText [text := (show newX)]
```

Layout Combinators

- GUI layout is a pain.
- wxHaskell has layout combinators:
  - `widget :: Widget w => w -> Layout`
  - `margin :: Int -> Layout -> Layout`
  - `column, row :: Int -> [Layout] -> Layout`
- Some widgets have a layout attribute:

```
set f [layout := margin 20
  (column 10 [widget display,widget b])]
```

GTK2HS

- Gtk is an alternative widget library, with a set of Haskell bindings, Gtk2Hs
- Gtk2Hs is very well supported.
- Low level interface

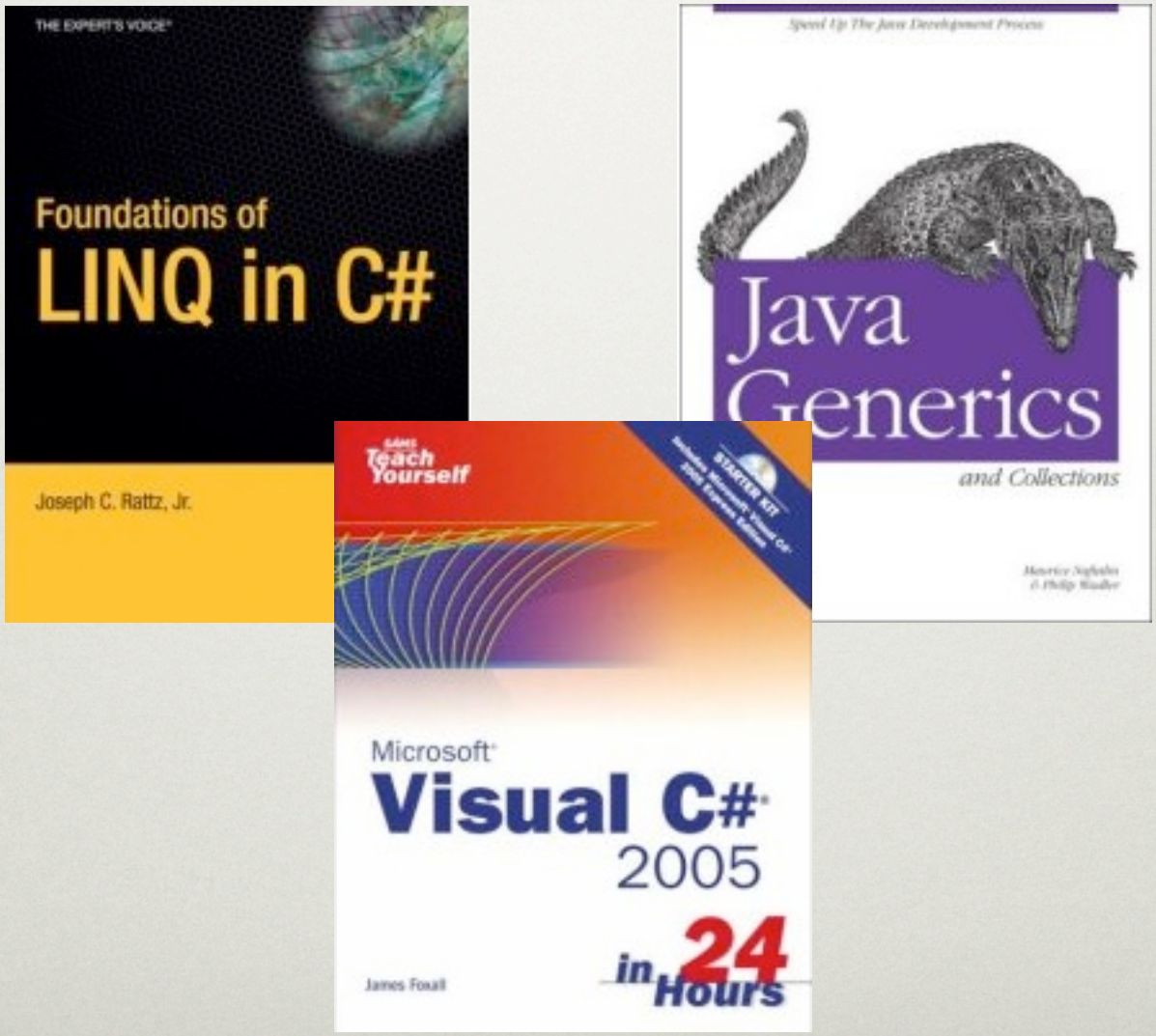
What Else Can Haskell do?

- Concurrency (STM)
- Database bindings (HaskellDB)
- Web programming (Wash)
- XML processing (HaXML)
- Hot swappable code (hs-plugins)
- ...

Where Is Haskell Used?

- Darcs - revision control system
- Pugs - Perl 6 implementation
- Dazzle - Bayesian network editor
- Frag - Quake 3 implementation
- Lava - hardware design

Haskell's Influence on Mainstream Languages



Where Haskell Pays off

- Aren't real Haskell programs just ugly imperative programs? Monads?
- No!
- Examples:

```
squares <- replicateM 9 (button f [])

mapM (set [enabled := False]) squares

chainWizardPages (x:xs)
  = do zipWithM setNextPage (x:xs) xs
       zipWithM setPreviousPage xs (x:xs)
```

Conclusion

Haskell is the world's finest imperative language!

Recommended Reading

- *Simon Peyton Jones*. Tackling the Awkward Squad.
- *Daan Leijen*. wxHaskell: A Portable and Concise GUI Library for Haskell.
- Sources of demos will be available soon.