Caml Trader:
Adventures of a functional programmer on Wall Street

Yaron M. Minsky
Managing Director
Jane Street Capital

CUFP Talk, September 2006

How I got here...

- Finished PhD in Distributed Systems at Cornell – No finance background
- Spent a year consulting 1/2 time with Jane Street, doing quantitative research
- Trading is more fun! Jumped ship in 2003

Jane Street (circa 2003)

- Small proprietary trading company
  (No customers!)
- Quantitative trading (esp. arbitrage and market making)
- Very hush-hush
- Basic technology – Mostly Excel and VB, (and some C#, Java and C++)

Excel and VB?

- Excel is a great tool!
    - Rapid development
    - Makes (some) errors easy to catch
    - Most common FP language out there
- VB fits Excel like a glove

...but it couldn’t last

- Poor performance
- Nearly impossible to version control
- Too much cut&paste
- Attempts to rewrite key infrastructure in C# had stalled
  - Too verbose
  - Too complex

The OCaml Experiment

- Quant group had been using OCaml since 2002, with good results
- Early 2005, management decided to give OCaml a try
- Experimental Project: rewrite key trading systems in OCaml

FP as a recruitment strategy

- Lots of great Java programmers out there, but how to find them?
- Small number of great FP hackers, but fewer great jobs in industry
- OCaml is a signal of quality, in both directions
- See Paul Graham's "The Python Paradox"

Readability

- Code quality taken very seriously
- Partners review much of the code that goes into production systems
- OCaml was easier to review and reason about.
  - Terse, easy to avoid boilerplate
  - Type system as proof assistant

Robustness

- It's hard to write reliable imperative code (but we do)
- Types are essential
- Small features matter: labeled arguments and polymorphic variants
- Type-inference makes it easier to change code quickly and reliably

Performance

- High performance requirements: 100,000's of txns/sec, sub-millisecond latency
- OCaml generates fast code
- Performance is easy to understand (mostly)
- Good and fast FFI

How did it go?

- Within 6 months, a number of key systems had been rewritten
- Performance far better
- Better modularity (most code reused between systems)
- Much shorter (even not counting reuse)
- New systems implemented strategies more complex than previously possible

Problems

- Harder to support Windows
- Hard to take advantage of SMP systems
- OCaml has limited GUI toolkit support

Jane Street Now

- OCaml Everywhere:

  Research, Trading Strategies, Systems Administration, Monitoring tools...
  (VB and Excel still around)

- Team of 10 programmers and researchers working primarily in OCaml

- 1/2 of core infrastructure is now rewritten (and working on the other half)

- 250kloc of OCaml, and growing fast

Lessons

- Languages matter, people matter more
- Leadership that understands technology is a must
- FP is a real competitive edge for a company nimble enough to use it

We’re Hiring

yminsky@janestcapital.com