Birth of the Industrial Haskell Group

Duncan Coutts

Well-Typed

CUFP 2009

Outline

A shared infrastructure

Establishing a consortium

An idea for shared development

Economics of programming languages

The economics drives us to *shared* languages and implementations.

- Private languages are a private cost
- Shared languages:
 - · more public resources
 - more skilled people available
- Shared implementations:
 - share past development costs
 - opportunity to share future development costs

The more we do share, the more we can share

The more we share already, the greater the opportunity to share costs of new development

- Compilers
- Standard libraries
- Tools (profiling, testing, etc)

Models of programming language development

- Proprietary product
 - F#
- Open, central commercial vendor
 - Erlang
- Open, no central vendor
 - Haskell
 - ML
 - Lisp

Funding programming language development

Who do you pay?

- do it in-house
- central vendor
- consultants
- grad students...

How do we share costs?

Open community languages

Open community languages have particular advantages and disadvantages

- Loads of stuff for free
- Choice of consultants
- Academics and open source hackers can have different priorities and timescales
- Harder to share future development costs

We think a consortium model is a good match for the open languages.

Outline

A shared infrastructure

Establishing a consortium

An idea for shared development

Why start a consortium now?

Indicators of commercial use pointing upwards:

- Job postings
- Informal discussions
- CUFP attendance
- Mailing list traffic, downloads, feature request tickets

Planning discussions

Discussed it with Galois after CUFP last year

- Who does the organisation?
- Issue of cost and expected number of members

The Caml Consortium

- Aimed for around 20 members
- Cost: €3k- €10k for 12 months (\$4k-\$14k)
- Provides OCaml & libs under 4-clause BSD license
- Started with 4 members in 2002, 7 members by 2008
- Initially unable to fund full-time development
- Now has 10 members

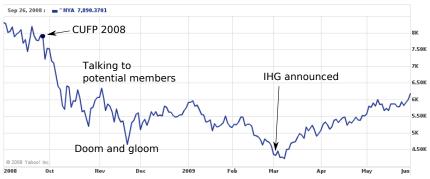
Our analysis:

not charging enough, aiming for too many members

The Industrial Haskell Group

- Aim initially for 5 members
- Cost £6k for 6 months (\$10k)
- No special license

Starting a venture in a recession...



"Great idea! Call us back in a year."

Birth of the IHG



- Started in March 2009 with 3 members
 - Including Galois and Amgen
- Funded 2 man-months of development work

How we decide what to do

- Internal mailing list
- Collect wish lists
- Look for overlaps and high priority tasks
- Collectively agree on the tasks

What IHG members asked for

- Short & medium term projects
- Feature additions
- System integration
- Development tools
- Not bug fixes
- Not releases
- Not language or core compiler issues

What the IHG has funded so far

- Dynamic libraries on Linux
- Ongoing work for dynamic libraries on Windows
- Allow building GHC without GMP lib
- Cabal improvement to reduce build times by increased sharing

Reflections on the process

- "Individual pots" have not been used much
- We would add our own suggestions for projects

Future aims

- Expand membership
- Add price-point for small companies
- Consider "sponsorship" level membership

Outline

A shared infrastructure

Establishing a consortium

An idea for shared development

What should consortia fund?

- Whatever the members want!
- Short and medium term projects of direct benefit: adding features
- Fixing bugs, testing, performance, making releases
- Development infrastructure

Investing in infrastructure

A modest investment in development infrastructure...

Potentially large benefit

- more open reusable code
- higher quality (code, tests, docs)

Mechanism: help the open community to do more

A quick poll...

Benefits of a community language

Hackage — Haskell's package archive

- 1,500+ packages
- 400+ developers
- Growing steadily
- Mostly uniform packaging

Hackage contains

- Robust reusable libraries and tools
- Latest academic research
- Plenty of chaff

The Hackage example

For example, extend Hackage by publishing

- Build results
- Test results
- Test coverage
- Quality metrics

Benefits

- Distinguish the good packages
- Encourage quality

Virtuous cycle between commercial and other users



Summary

- Opportunities to share development costs
- Consortium model for open languages
- Invest in development infrastructure