

allejobsinleuven.be

Peter Vandenabeele
BarCamp Gent, 29 March 2008

- whoami
- the problem/solution
- business/architecture choice for Rails
- SEO
- 1 month after launch
- social networking dynamics ?

whoami

- Peter Vandenabeele (vandenabeele.com)
- studied/worked mainly in Leuven
- Mind (embedded Linux 2000-2006)
- Fedict (ODF Belgium 2006-2007)
- summer 2007: find a different job/project in Leuven area => where would I fit ??
- Jan-Feb: full-time (200%) on the site ...
- now: part-time consulting (financing further development ...)

The problem (1)

- choosing a job in Leuven
- from the job seeker perspective
- step 1: criterium of completeness:
am I aware of *all* opportunities?
- most interesting places (80%+ ?) do *not* advertize jobs on major job boards
 - 900 EURO/job/month and even then ...
 - very little qualitative responses
 - while I know a lot of people that might change

The problem (2)

- step 2: *choosing*, narrowing down
- where do I want to work?
company culture: soft values such as honesty, responsibility, perfectionism, ...
- how do I know?
 - 2 interviews of 45 minutes ... to decide over 5 year full-time involvement ?
 - facade is often different from interior ...

The solution (?)

- *all* jobs: allow gratis posting and actively collect job posting for Leuven companies to create critical mass
- anonymous for job seekers (no CV's)
- *choosing* a job: use social networking
 - I certainly have friends that have ideas about many employers (not only that work there)
 - people give *recommendations* for employers?
- thinking about “automatic” indications

Why Rails ?

- Easy, fast development
- greenfield ... (I had very little experience in developing for the web)
- it works, happy camper :-)

Rails risks: scalability

- Tom Klaasen: “How many *transactions* will you need?” => 0
- scalability becomes a non-issue
- 1 master database, N slaves
- Rails is just 1 interface to the database
- if ever needed, other “read-only” interfaces to slave database(s) can be set-up or caching can resolve most issues

Rails risks: security

- main risk is in our programming style
- Rails motivates clean styles
 - e.g. `before_filters` with “:except”
 - sensible defaults
- we found ZERO bugs in the production framework (but only used basic features)

SEO

- add keywords to titles, URL, h1, ...
- 1000 “nearly identical” job entries gave very bad results in Google ...
 - jobs with specific content /job_details/
 - exclude /jobs/ with robots.txt
- some “old” URL's continue to pop up
 - be careful with exposing temporary URL's
- use permanent redirects

1 month after launch

- initial interface had 3 search fields (job ; location ; region)
 - too complex for users
 - reduced to 1 box
 - AJAX not understood by many users ...
- number of visitors flat
- need to develop faster updating of jobs
- still thinking about the social networking aspect

Social Networking

- how to make a *selfish* action result in *global* benefit ...
- example LinkedIn:
 - A links to B (selfish for A: to see more details of B and to be linked to friends of B)
 - global benefit: friends A and B now also link to B and A
- example del.icio.us:
 - my storage of links creates global reputation system

Social networking in jobs

- How can a recommendation of an employer yield *selfish* result ?
 - find colleagues
 - receive money: not (now used if a person references a recruit ...)
 - add to reputation ?
 - altruism ??
 - private/friends only ?
 - contributors get “early access” ?

Conclusions

- Rails is great
- user + customer feedback to steer further developments
- thinking about social networking
- developing more automatic scraping
- first spontaneous jobs submitted :-)