

Matt Hellige
matt@immune.net
<http://matt.immute.net/>

2639 N Monticello Ave
Chicago, IL 60647
773-255-6483

OBJECTIVE

To write interesting software with a group of excellent programmers, in a company with a strong engineering culture. I'd be particularly interested in being hired to program in Haskell or Scala, in a Unix (or Mac) environment.

I would love to be paid to develop open source software, and particularly open source tools for programmers (compilers, IDEs, libraries and frameworks, etc.). I also deeply enjoy reading, writing and teaching about programming.

I'm not willing to relocate, but I am willing to travel periodically, and I'm willing and able to work remotely.

EXPERIENCE

Accenture Tech Labs
Chicago, IL

Researcher
June 2006 – Present

As a member of the Systems Integration and Software Engineering group, I conduct research in software engineering. My main areas of research interest are programming languages, components and modularity, architecture, configuration languages, and development tools. I'm particularly interested in applying functional programming languages and techniques to real-world problems.

Orbitz
Chicago, IL

Technical Lead
March 2004 – May 2006

In two years at Orbitz, I became the domain expert in hotel content management and delivery, and delivered a number of related projects, including new data load processes and integration of several new content sources. A highlight is the smooth implementation, integration and launch of hotel guest reviews.

I also took an active role in improving our versioning, release management and deployment systems, working with the SCM group as well as development to refine our processes and tools. While this effort was still in its infancy when I left, preliminary results were encouraging. Other work included a large-scale rearchitecture of the Hotel application infrastructure, and an ongoing refinement and formalization of our component development model.

Throughout my tenure at Orbitz I was also a part of numerous tangential efforts, including troubleshooting the hotel systems at all levels, debugging complex third-party source code, designing and reviewing systems architecture, and various smaller projects. I also contributed enhancements to core Orbitz platform components such as OJF (the Orbitz Jini Framework).

Center for Connected Learning
Northwestern University, Evanston, IL

Research Programmer
January 2003 – March 2004

I worked with a team of three full-time programmers and five graduate students on NetLogo, a language and programming environment for multi-agent simulation. NetLogo is used for education (curriculum development, constructionist learning) as well as scientific research in a wide range of settings. Areas of application include chemistry and physics, sociology, epidemiology, ecology and biodiversity, and economics. NetLogo is written in Java and offers scientifically repeatable results on Windows, Mac and Linux machines.

My responsibilities included participation in the design and implementation of the NetLogo language, improvements to the HubNet distributed simulation system, and development of the Connected Chemistry curriculum, as well as improvements to numerous NetLogo models.

I also provided support and programming help for end users, as well as supervising and mentoring undergraduate interns.

Curious Networks
Chicago, IL

Senior Software Engineer
February 2000 – April 2002

At Curious Networks I worked with a team of ten programmers on the conception, design and implementation of Continuum, a multi-channel presentation server which allows the developer to write an on-line application once and then deploy it for interactive use via numerous information access channels.

Continuum is written in Java and is deployed on Windows, Solaris and Linux systems. Supporting architecture has been implemented in Python, shell scripts, make and ant. We use cvs for revision control, and our process is a loose adaptation of Extreme Programming (XP).

My programming work included the design and implementation of an XML-based application specification language; the design of multiple independent rendering components; system architecture and request processing; fuzzy “best-match” classification of incoming requests by requesting device; design and implementation of a rule-based, channel-specific renderer customization system; a plug-gable extension system; API design and organization; implementation and use of custom Java class loaders.

Concurrently, I participated in both short and long-term product and development planning, requirements analysis, process and development environment management, quality assurance, and product documentation.

Motorola
Champaign, IL

Software Development Intern
May 1999 – October 1999

Neoglyphics Media Corporation
Champaign, IL

Software Engineer
August 1997 – February 1999

Natl. Center for Supercomputing Applications
Champaign, IL

Student Developer
September 1996 – October 1997

Fermi Natl. Accelerator Laboratory
Batavia, IL

Software Development Intern
August 1994 – June 1996

EDUCATION

University of Illinois at Urbana-Champaign
BS in Mathematics and Computer Science

August 1996 – May 2000

PUBLICATIONS

Qing Xie, Mark Grechanik, Matt Hellige. SMART: A Tool for Application Reference Testing, *International Conference on Automated Software Engineering (ASE)*, Tool Demonstration Track, November, 2007, Atlanta, GA.

Kevin Conroy, Mark Grechanik, Matthew Hellige, Edy S. Liongosari, and Qing Xie. Automatic Test Generation From GUI-Based Applications For Testing Web Services, *23rd International Conference on Software Maintenance (ICSM)*, October 2007, Paris, France.

OPEN SOURCE AND COMMUNITY

Site editor on Lambda the Ultimate (<http://lambda-the-ultimate.org/>)

Developer of mtail (<http://matt.immute.net/src/mtail/>)

Participation in several open source projects and mailing lists

SKILLS AND SPECIALIZATION

As a software engineer, I am very familiar with and interested in:

- Java, Scala, Haskell, C, Python, Ruby, Scheme, Standard ML, Lisp
- object-oriented design and design patterns
- functional programming
- network protocols, from the relatively low-level (TCP/IP, etc.) to the relatively high-level (HTTP, SMTP, Java RMI, SOAP)
- XML and related specifications (XPath, XSL, XQuery, etc.)
- UNIX (Linux and others), both from a user's perspective and from a programmer's perspective
- UNIX system and network administration
- shell scripting (bash and zsh in particular)
- build and SCM tools (make, ant, maven, cvs, subversion, accurev)
- web programming (HTML, JavaScript, CGI, servlets, etc.)
- Extreme Programming (XP), and agile process in general
- document formatting with \LaTeX

My main area of research interest is programming languages, ranging from theory to efficient implementation. Areas of particular interest include:

- functional and mostly-functional languages, e.g., Haskell
- very dynamic languages, e.g., Scheme, Python, Ruby
- the lambda calculus, pi calculus, combinatory logic and other foundational formalisms
- type theory
- module and component systems
- compiler/interpreter design and implementation, especially as they relate to the efficient implementation of very high-level languages
- programming education. How is programming best taught, and how can language design support and inform that process?
- language interoperability