Nicolas Pierron

Computer Science Engineer - NixOS' Developer - Modern Physics Teacher at EPITA

nicolas.b.pierron@gmail.com

Summary

Before EPITA: I ...
- Wrote my first "Hello, World" in 1997 on a TI-85.
- Searched for prime numbers with programs written in Pascal.
- Created games for TI-89 written in C.
- Published these games on the web with Javascript and HTML.
- Solved Einstein problem (with houses, pets, ...) with Prolog.
- Built a game emulator in C++ with a person met on the web.

At EPITA: (see all details below)
In addition to (yet another list of) computer languages like: OCaml, Delphi, Shell script, Sed, M4, Lisp, Java, etc ...

In addition to EPITA: I ...
- Learnt IA-32 in two months to understand & patch a MMORPG server.
- Went to Australia to improve my English and to do an internship in a small company.
- Worked at LRDE (Laboratory of Research and Development of EPITA).

In my Internships: I ...
- Worked with Stratego (a language made for program transformation) to automatically introduce design patterns in Java by solving coupling constraint.
- Introduced an attribute grammar inside the Stratego compiler, and fix the staged compilation issues to easily extend the compiler in the future.

After EPITA: I ...
- Studied Fundamental Physics to have a better understanding of new technologies.
- Introduced modularity in NixOS.

My papers:
- NixOS: A Purely Functional Linux Distribution (Journal of Function Programming: waiting for approval)
- Semantics driven disambiguation: A comparison of different approaches (LDTA 2008)

I speak:
- French (mother tongue)

- English (fluent, 2 months in Australia & 6 months in the Netherlands)

My Interest:
- Computer Science
- Physics

My Goals are:
- A lot of technical problems (R&D if possible).
- Functional Programming
- Open Source Development

Specialties

Computer Science:
- Functional programming (FP) languages: OCaml, Haskell, Nix, Lisp, ...
- Assembly languages: IA-32, Motorola 68K, ...
- GNU/LInux distributions: Debian, Gentoo, NixOS, ...
- Strategic programming (Stratego)
- Domain Specific Languages.

General:
- Understand and generalize complex codes/problems.
- Understand theoretical aspects.
- Bad jokes (sometimes good).

Experience

Contributor at NixOS

August 2008 - Present (1 year 2 months)

NixOS[1] is an open source Linux distribution based on a lazy functional programming language called Nix, which has maximal sharing. These features have helped me to implement a new writing style of NixOS that allow all users to have a system which is modular with non-intrusive extensions. This new writing style give the opportunity to NixOS users to develop their own device specific configurations which can be included in any other configuration. Thus, if person have the same device, they just have to include the corresponding device configuration file to configure it correctly.

[1] http://nixos.org/nixos/index.html

Teacher at EPITA

April 2009 - June 2009 (3 months)

The goal of this course is to entice people to learn by themselves on technological advances. In this course, understanding is put forward in relation to learning. This course is based on modern

physics and its origin and concluded with unusual technologies to give students a desire for creativity. (third-year students, ~47p.)

Summary:
1/ Light: The experiments that led to the conclusion that « c » is a constant.
2/ Relativity: Proof of relativity and the equation $E = mc^2$.
3/ Quantification: Black bodies, Electron charge and Spin.
4/ Wave-particle duality: Remarkable experiences.
5/ Wave function: The mathematical tools and the Schrödinger equation.
6/ Uncertainty principle of Heisenberg: The diffraction and the Casimir effect.
7/ Unusual propulsions: Ramjet, lifter...

Develop Stratego extensions at Delft University of Technology
February 2008 - August 2008 (7 months)

The goal of this 6 months internship, in which Eelco Visser, Tony Sloane and Lennart Kats are involved, is to add an attribute grammar system in Stratego that handles term rewriting. Many issue arose after the implementation of a left-to-right rewriting based on the idea of the attribute grammar system of Transformers.

I fixed the bootstrap sequence of the whole compiler and its libraries to handle changes in the generated code. This issue took several months with some hacks in the Autotools to handle staged compilation. Thus, I worked on dynamic strategies with heap allocated frames. In the same way, I fixed a bug to merge dynamic libraries at runtime in order to fit the language specification.
1 recommendation available upon request

Student/researcher at EPITA/LRDE
January 2006 - July 2008 (2 years 7 months)

Parallel scholarship at EPITA through the LRDE, the R&D laboratory of the school.

the Transformers project[1] aims at making transformation of C++. My work on this project was mainly to improve tools used to do this job. I have introduced a new model to represents Attribute Grammars. This model based on hybrid logic was used to declare relations between attributes and rules to populate the attribute grammar in order to fill up more than 70% of the code[2]. This work has been rewritten, with the help of Sigoure Benoit and Akim Demaille, to be submitted to the LDTA conference[3].

I made almost 2 K-lines of C++ basic tests from the C++ standard. These tests are highlighting differences between C++ parsers (g++, EDG C++ parser).

[1] https://trac.lrde.org/transformers/
[2] http://www.lrde.epita.fr/dload/papers/pierron.07.seminar.formal.def.pdf

[3] http://www.lrde.epita.fr/dload/20080116-Seminar/pierron-propagation.pdf

3 recommendations available upon request

Automatic transformations on Java to improve the maintainability at Lip6
September 2006 - December 2006 (4 months)

Lutin is a project initiated by Mikal Ziane which has to factor Java codes depending on constraints. My job was to reproduce transformations made on a graph extracted from the original source code. This project improves the maintainability of Java projects by making small transformations, which can be assimilated to design patterns small bricks. This practical work has trigger my curiosity about models and their completeness. In this internship, I have made several suggestions about unnecessary transformation and missing information inside the abstract model. This work has been concluded by a presentation at the Stratego Users Day 2006[1].

[1] ftp://ftp.stratego-language.org/pub/stratego/SUD06/sud06-lutin2.pdf

Game developer at eAcceleration Corp.
August 2006 - September 2006 (2 months)

This job was to maintain the competitiveness of the game T4C ("The 4th Coming", ~450 K-lines of ugly C++) by improving it. This job allowed me to work in coordination with the constraints of graphical designers and gamers. To coordinate the development and deployment of new releases, I taught other developers how to use Subversion to synchronize their work. The team was working remotely from many countries around the world.

I started this job after I had made two assembly patches on the binary of the game server. These patches took me two months to learn IA-32 by myself from the Intel documentation (a recent version is available at [1]) and a few weeks to create them.

[1] http://www.intel.com/products/processor/manuals/

Teacher at EPITA
October 2005 - June 2006 (9 months)

Practical work in computer science (OCaml / Delphi) for first-year students (~30 p.).

Web services extension at This Planet
August 2005 - August 2005

One month internship in Australia. Find a way to spread Power Builder programs as web applet.

Education

Université Pierre et Marie Curie (Paris VI)
Fun, Fundamental Physics, 2008 - 2009

Ecole pour l'Informatique et les Techniques Avancées
Engineer, Computer Science, 2003 - 2008

Shafston University
English, 2005 - 2005

Interests

Stratego, Attribute Grammar, compilation, C++, parsing, optimization, quantum mechanics, new technology, unicycle, gliders pilot, plane pilot

Nicolas Pierron

Computer Science Engineer - NixOS' Developer - Modern Physics Teacher at EPITA

nicolas.b.pierron@gmail.com

---

4 people have recommended Nicolas

"Nicolas was my Stratego/XT teacher at the LRDE. He taught me everything about the Transformers project, and continue to give me some hints to address some issues. His work on Transformers is just brilliant. Working with him was a real pleasure because he share his passions through his speech, which makes him a good teacher."

— Vincent Ordy*, Student / Researcher, EPITA/LRDE*, worked indirectly for Nicolas at EPITA/LRDE

"Nicolas is a talented programmer. He is able to address extremely delicate problems in hostile environments with creative solutions. He understands very well theoretical aspects of programming. He is a nice person with whom working was a pleasure. He was praised by his fellow students who he regularly helped."

— Akim Demaille, was Nicolas's client

"Nicolas is a very skilled R&D type of software engineer. He knows how to read complex research papers and implement them. He's determined and will spend the time it takes to tackle hard problems that no one successfully solved before. I'd recommend him for any R&D lab kind of position."

— Benoît Sigoure*, Student / Researcher, EPITA Research and Development Laboratory*, worked directly with Nicolas at EPITA/LRDE

"Nicolas is smart and an experienced (Stratego) programmer. Whenever we had Stratego problems or other programming problems, Nicolas always came up with good solutions. He was extremely helpful."

— Zef Hemel*, Ph.D. Student, Delft University of Technology*, worked directly with Nicolas at Delft University of Technology

---

Expertise

Software Development

Contact Nicolas on LinkedIn