
Education

Jan 2013 – present **B.Sc Mathematics**, *Université Laval*, Québec.

Experience

- 2014 (ongoing) **GSoC student with Boost**, *Google Summer of Code*.
Work on [Boost.Hana](#)
- 2013 **Student volunteer**, *C++Now*, Aspen.
Help the conference staff with logistics, recording talks and other tasks
- 2012 **Intern**, *Coveo Solutions*, Québec.
 - Conception and implementation of a deadlock detection system for internal use
 - Presentations on C++ techniques and idioms to co-workers:
 - The Boost.ConceptCheck library and associated TMP techniques
 - C++11 rvalue references
- 2010 **Intern**, *Department of Electrical Engineering of Université Laval*, Québec.
Organization of the 10th international conference on [Quantitative Infrared Thermography \(QIRT\)](#)

Talks

- 2014 **Towards painless metaprogramming**, *C++Now*, Aspen.
- 2013 **A system for resource deadlock prevention**, *C++Now*, Aspen.
- 2013 **Deadlock detection with d2**, *OpenCode #12*, Québec.
- 2013 **Concept based overloading in C++**, *OpenCode #9*, Québec.

Personal Projects

- [Boost.Hana](#) **An experimental C++1y library for heterogeneous computation**
Design and implementation of a library to manipulate heterogeneous sequences at compile-time or at runtime. The library provides the functionality of the Boost.MPL and Boost.Fusion behind a single purely functional interface. The C++ type system is also extended with type classes inspired by Haskell.
- [mpl11](#) **Conception and implementation of a C++11 replacement for the Boost.MPL**
Reimplemented the functionality of the Boost.MPL library using new template metaprogramming techniques made possible by C++11. Redesigned the API of the library using ideas from Haskell to make it more powerful, easier to use and to extend.
- [d2](#) **Conception and implementation of a deadlock detection system in C++**
Detects deadlocks that would have happened under different thread scheduling conditions by performing intrusive dynamic analysis on a non-deadlocking run of a program. Additionally, provides statistics about lock and thread usage.
- [joy](#) **Implementation of a preprocessor metaprogramming library**
Implemented associative sequences and other utilities for preprocessor metaprogramming on top of the [Chaos preprocessor library](#).
- [nstd](#) **Conception and implementation of a generic algorithm library in pure C**
Implemented a basic name mangling system and “preprocessor-based classes” using PMP techniques. Using these facilities, implemented a subset of the C++ standard library algorithms. The result is a collection of generic algorithms instantiable and usable from pure C without sacrificing type safety or performance by using traditional techniques like pointers to void.

- [duck](#) **Implementation of a minimal concept-based overloading library**
Implemented a subset of Boost.ConceptCheck's concepts as metafunctions, which allows overloading based on the modeling of a concept by a type.
- [cisp](#) **Implementation of a minimalist object system with the preprocessor**
Created a system to manipulate complex preprocessor objects using associative sequences imbued with object semantics.
- [nstd-lang](#) **Implementation of a translator for a toy language in Python**
Implemented basic parsing, semantic analysis and code generation to C.
- Contributions to other projects**
- Contribution of the [hawick_circuits](#) algorithm to Boost.Graph
 - Occasional patches to Boost (Spirit, Graph, Archive, MPL and others)
 - Active on the [Boost.Dev](#) mailing list
 - CMake port of the [FastPFor](#) integer compression library's build system