




Stephan Brandauer

Computer Science and Software Development.

Address	Bredmansgatan 4A, 752 24 Uppsala, Sweden	Phone	+46 700 599 236
Nationality	Austrian	E-Mail	stephan.brandauer@it.uu.se
Languages	German (<i>native</i>), English (<i>fluent</i>) Swedish (<i>basic working proficiency</i>)	Web	http://stbr.me
		Other	 sbrandauer ,  kaeluka  stephan-brandauer

Personal Profile

I'm a last year PhD student at Uppsala University. My research has focused on aliasing (several variables holding references to the same datum) in imperative programming languages. Aliasing makes both writing, understanding, and optimising code hard. Aliasing has been the common theme that has been tying together my research spanning from **type system/language design** to **dynamic analysis of program corpora** to a domain-specific language for **data structure-design and -optimisation**.

I deeply care about program performance, but I also think that most programmers shouldn't need to. To this end, I've been trying to find ways to constrain high level languages in just the right way: constraints that don't hurt writing code in practise, yet give enough information to compiler, runtime, or framework to do optimisations that can provide great program performance.

I'm especially, but not exclusively, interested in opportunities that include any or several of the following:

- Building tooling support for program development. This includes performance measurement, bug finding, testing, data-visualisation and -analysis, framework- and runtime-development.
- Doing research in an industrial setting.
- Teaching/mentoring others.
- Work on language design- and implementation, including domain-specific languages for domains I'd get to learn about in the process!
- Working on programming runtimes or garbage collectors.

Education

2013-2018 PhD in Computer Science - Uppsala University

Research on programming language design, analysis, and implementation.
During my PhD education I have:

- ... designed *Disjointness Domains*, a type system to express fine grained alias invariants in data structures.
- ... implemented *Spencer*, a dynamic analysis tool that runs real world Java software, collects extensive program traces and lets users analyse these traces using a specifically designed *domain-specific language* executed by a *web service*.
- ... worked on (*name withheld*), a domain-specific language and compiler using Java's annotation framework that let users implement data structures that are simple, have good performance, and can adapt their performance to match a wide range of use cases. (*name withheld*) is currently unpublished.
- ... worked on the compiler of the *Encore* research language, an object oriented programming language with concurrently executing actors as objects.

- 2011-2013** Master of Science, Computer Science. Uppsala University.
Degree project: design, implement, and benchmark a mailbox data structure for an actor-based language (called “Joelle”) that permits *parallel processing of messages within an actor*.
- 2007-2011** Bachelor of Science, Cognitive Informatics. Bielefeld University.
Degree project: Build a 2D rigid body and particle physics engine with an interactive UI.

Selected Publications

- ??? *(name withheld)*
(under submission)
A domain specific language to develop data structures that are simple, have good performance, and can adapt to many different use cases.
- MSR 2017** [Spencer: Interactive Heap Analysis for the Masses](#)
Int'l Conf. on Mining Software Repositories (MSR) 2017. Buenos Aires, AR.
The paper that introduces the Spencer project.
<http://stbr.me/spencer>
- QAPL 2017** [Mining for Safety using Interactive Trace Analysis](#)
Workshop on Quantitative Aspects of Programming Languages and Systems (QAPL) 2017. Uppsala, SE.
An application of Spencer to a corpus of programs, looking for safety properties of objects (like uniqueness, immutability, etc).
<http://stbr.me/spencer>
- SFM 2015** [Parallel Objects for Multicores: A Glimpse at the Parallel Language Encore](#)
15th Int'l School on Formal Methods f. Design of Computer, Communication and Software Systems (SFM) 2015. Bertinoro, IT.
An overview of the Encore language.
<http://stbr.me/Encore-Glimpse>
- OOPSLA 2015** [Disjointness Domains for Fine-Grained Aliasing](#)
Object-Oriented Programming, Systems, Languages and Applications (OOPSLA) 2015. Pittsburgh, PA, USA.
A novel type system for alias control.
<http://stbr.me/Disjointness-Domains-for-Fine-Grained-Aliasing>

Employment History

- 2013 - Present** [PhD Student in programming language design, implementation, analysis.](#)
Uppsala University
See in Education section above.
- Feb 2009 - Jun 2010** [Research Assistant](#)
Bielefeld University, AI Group
Work on, and maintain, virtual reality applications for cognitive science studies. Teaching assistant (run lab sessions, grade home work assignments).

- Sep 2007 **Freelance Programmer, C#**
 Mar 2008 Comet Consulting, Salzburg (now part of Rhomberg)
- Develop 3D image recognition algorithms and software in **C#** for 3D LIDAR scanners to monitor safety procedures at railway tunnel construction sites. Much of the job was on-site, but overlap with the social work job was during nights and evenings.
- Sep 2006 **Social Work**
 Mar 2008 Laube Sozialpsychiatrische Aktivitäten GmbH
- Austrian civil service, as an alternative to being drafted for the military. Work with chronically mentally ill people. Learned lots.

Software Engineering Skills

■ Programming Languages

(roughly in order of familiarity)

Java - proficient

Scala - have used it on several occasions, for example the *Spencer* and (*name withheld*) DSL-compilers.

Haskell - used (and loved) it lots during the first 3 years of my PhD studies, working on *Encore*; not much since.

Rust - Have implemented a throw-away prototype version of (*name withheld*). I know enough to be dangerous.

C - have been teaching basic C to university students every year of my PhD.

C++ - know the core principles that distinguish it from C (smart pointers, RAII, objects, ...), but would like to know more.

SQL (Postgres-SQL) - have used **Postgres-SQL** to implement complex graph queries in *Spencer*.

■ Miscellaneous

Data Analysis: I have used **python/pandas**, **Apache Spark** (on a single node; including **GraphX**), and **Postgres**.

Optimisation of JVM code: optimisation is a big part of my research work on (*name withheld*). I have used the **JMH** framework, **VisualVM**, and **JITWatch**.

Version control: mostly using **git**. Also, although rusty, **Mercurial** and **SVN**.

Compilers: I have worked on several **compilers**: one for a general purpose language (*Encore*) and have developed two compilers for domain-specific languages (for *Spencer*, compiling to SQL; and for (*name withheld*), compiling to Java bytecode).

Java Bytecode: I have used both the **ASM framework** and **bytebuddy** to a) **programmatically modify Java programs as they are running**, and b) **generate Java code** from high level specifications.

References

Tobias Wrigstad PhD advisor Associate Professor at Uppsala University
 tobias.wrigstad@it.uu.se

He has supervised both my master's and my PhD thesis work.

Sophia Drossopoulou Research collaborator Professor at Imperial College London
 sophia.drossopoulou@gmail.com

We were both members of the *Encore* team that involved groups from several universities.