Conal Mullen Elliott

conal@conal.net http://conal.net/

EDUCATION

◇ **Carnegie Mellon University, School of Computer Science**
Ph.D. 1990 in computer science. My dissertation was *Extensions and Applications of Higher-Order Unification*. There I derived unification algorithms for various typed lambda calculi and demonstrated applications to program manipulation and theorem proving. Other work included λ-PROLOG, a higher-order logic programming language.

◇ **University of California at Santa Barbara, College of Creative Studies**
B.A. in mathematics with honors 1982.

EMPLOYMENT

◇ **Researcher,** Independent, February 2020–current.
I am investigating scalable design and synthesis of efficient, massively parallel computing hardware in tandem and their formal proofs of correctness. This exploration combines denotational design and category theory and is built in Agda, a dependently typed programming language and proof assistant.

◇ **Distinguished Scientist,** Target, May 2016–February 2020.
I continued to explore elegant and principled techniques from math and programming language theory for building fast, correct, and beautiful software, now with applications including machine learning and other large-scale optimization problems. Much of my recent work is based on category theory, particularly automatic translation of Haskell programs into various categories for enhanced abilities such as automatic differentiation and for massively parallel execution on GPUs or FPGAs. My main working language is Haskell, as it has been since 1995.

◇ **Principal Engineer,** Tabula Inc, January 2012–March 2015.
Tabula developed the *Spacetime* 3D programmable logic device (3PLD), which was a significant advancement over mainstream FPGA technology. My main activities at Tabula:

- Taught classes on functional programming in Haskell, emphasizing elegance, reuse, correctness, and fun.
- Helped extend and redesign the in-house system for chip specification.
- Developed a prototype optimizing compiler from Haskell to hardware, together with several modest applications. The compiler converts functional programs into an algebraic/categorical form, which is then interpreted as sequential circuits. Haskell is a radically non-sequential language and hence well suited to massively parallel architectures such as Spacetime.

◇ **Mentor,** Tsuru Capital LLC, August 2010–August 2010 (three weeks).
Taught advanced Haskell programming and design skills. Reviewed existing code and offered detailed suggestions for improving simplicity, and reliability. Outlined a new software architecture for high-frequency trading, aimed at improving correctness, performance, and ease of experimentation.

◇ **Architect,** Anygma, June 2008–September 2008.
Software architecture and design of artist-friendly but very expressive authoring tools for interactive 3D graphics, written in Haskell. Mentored group members and taught them "denotational design" which is a mathematically elegant, robustly composable, and rigorous methodology for software design.

◇ **Founder,** LambdaPix, Sep 05–January 2012.
Following a rich personal sabbatical, I applied much of my energy to computer science again, with renewed enthusiasm. My projects included the following:

- Consulting with companies in the application of functional programming, and Haskell in particular. I teach and mentor in mastery of functional programming techniques, Haskell, elegant & rigorous design & implementation, and denotational thinking/design.
- Generation of optimized graphics processor code from highly composable, purely functional, specifications.
- iPhone/iPad software development.
- Extensive technical blogging. See http://conal.net/blog.
- Many Haskell-based software libraries, shared on http://hackage.haskell.org.
- Design and implementation of authoring tools for non-programmers, allowing them to construct modular, efficient software components. My goal for such a tool is that it feel nothing like programming (certainly not like imperative programming), but still have the expressive power of programming. This is a vision I have been evolving since the early 1990s. The paper "Tangible Functional Programming" describes the foundation for my current approach.
- *Pajama* – a compiler to produce highly optimized Java applets from simple, declarative specifications of interactive media (e.g., imagery, animation, sound). While there are a few interesting design issues, this work mainly serves to make some of my past research openly accessible.
- 2D Interlaced designs in Java (http://conal.net/Interlace).

◇ **Instructor, coach, mediator,** Awakening Compassion, May 2005–Jan 2012.
I taught, coached and mediated conflict resolution in Compassionate Communication (also called "Nonviolent Communication" or "NVC", see http://conal.net/awakeningcompassion). NVC is a small set of inner and outer practices that reawaken compassion and support fully meeting everyone's needs by clarifying needs and inspiring compassionate support. It is used to great benefit with couples, families, work settings, and in resolution of ongoing violent conflicts. NVC provides an alternative to our culture's orientation, which focuses on judgment and blame. In this alternative, we focus our attention on what we are needing in the present (in a universal sense, as separated from particular strategies) and how we can enrich each other's lives. By retraining our attention away from what is "wrong" with someone else and back onto what is alive in us, we learn to communicate with clarity and vulnerability and encourage compassionate response in those who hear us. Similarly, we learn to listen to others so as to tune into the living feelings and needs that are always present beneath blame, judgment and demand. My partner and I presented introductory seminars and multi-week classes, lead practice groups, and worked with individuals and couples and companies.

◇ **Personal sabbatical,** Sep 02–Sep 05.
During my sabbatical, I explored mainly communication, spirit, psyche, music, dance and art.

- Compassionate Communication. Studied and practiced intensively. Attended nine-day intensive training.
- Shamanism. I did a 16-month intensive mentorship under a shaman, beginning in April 2003.
- Jungian work. I worked with a Jungian analyst for five years, centered on dreamwork. During this period, I have also participated in a men's support group and

there have learned about psychodrama and group processes. I am interested in facilitating similar groups in the future.

- Jazz singing. I studied under the jazz vocalist Andrienne Wilson and have had a few local performances.
- Art matting and framing. I make my own mats (single and double, beveled, rectangular and elliptical opening, some novel designs) and frames (not very often), and cut glass, for PAN-based images, photographs, and collages. So far, the results have been for gifts and my own home display.
- Ballroom dance. At my daughter's request, I took up ballroom dancing in November, 2003 (waltz, foxtrot, cha cha, rumba, east & west coast swing, tango, salsa, hustle). I love it.

◇ **Researcher,** Microsoft Research, Sep 94–Sep 02.
Most of my work at Microsoft Research was in design and implementation of languages and systems to make graphics programming simpler and more powerful. Some highlights:

- Original architect for the DIRECTANIMATION product. Conceived and refined the original design for this powerful media library that integrated audio, video, 2D and 3D graphics and animation. DIRECTANIMATION (originally called "ActiveVRML") was the first incarnation of the paradigm that has come to be known as "functional reactive programming".
- FRAN (http://conal.net/fran), a research system for improving on the ideas explored in DIRECTANIMATION. It introduced and explored declarative specification of interactive animation over a variety of types (2D images, 3D models, sound, and all attributes of these types) in a unified model of continuous-time behaviors and events. FRAN has been used in many college courses and has served as the basis of a commercial product, at least four Ph.D. theses, and many follow-up papers. The fundamental ideas have been further applied to robotics, vision, user interfaces and stage lighting control. (See, e.g., http://haskell.cs.yale.edu/publications/#FunctionalReactiveProgramming.)
- PAN (http://conal.net/Pan), a high-level language and optimizing compiler for image synthesis and manipulation. Worked with three product groups toward integration with their products.
- VERTIGO (http://conal.net/Vertigo), a language and compiler for computations on graphics processors. It generates C♯ and vertex shader assembly code. Supports procedural geometry and textures and programmable shaders, all in a simple and consistent framework, based on first-class functions.
- Mentored several interns successfully.

◇ **Architect/technical lead,** Sun Microsystems Inc., Dec 93–Aug 94.
MEDIAFLOW product, integrating *distributed* traditional multimedia (audio/video) with other media: graphics, synthetic sound, gesture, etc. Worked with associates from IBM and HP, in an effort to win support for industry standardization.

◇ **Architect/technical lead,** Sun Microsystems Inc., Sep 90–Nov 93.
TBAG project (http://conal.net/tbag), a paradigm and toolkit for rapid prototyping of interactive, animated 3D graphics programs. The paradigm has its roots in declarative (constraint and functional) programming, emphasizing immutable values, first class functions, and relations, applying these concepts to a broad range of types, including points, vectors, planes, colors, transforms, geometry, and sound. The narrow role of modifiable state in this paradigm allows applications to be run in a collaborative setting (multi-user and multi-computer) without modification.

- **Member of technical staff,** Sun Microsystems Inc., Apr 90–Sep 90.
  OpenWindows Graphics Extensions group. Ported sample implementation of multi-buffering X11 extension to Sun's X11/NeWS server.
- **Research Computer Scientist,** Kestrel Institute, Palo Alto, CA, Aug 89–Mar 90.
  Worked on programming language and program manipulation design. Investigated use of Mathematica for geometric program synthesis. Position eliminated because of funding cut.
- **Research programmer,** Computer Science Department, Carnegie Mellon University, Summers 85–88.
  Worked with other members of the ERGO project (which deals with program transformation and synthesis) in designing and implementing many of the components of the ERGO Support System (in Common Lisp).
- **Consultant,** Information Technology Center, Pittsburgh PA, Sep 86–Dec 86.
  Designed a framework for extending the Andrew window system to incorporate color.
- **Consultant,** Formative Technologies, Pittsburgh PA, Sep 84–Jan 85.
  Designed and implemented graphics and systems software.

Professional Activities

- Member of IFIP Working Group 2.8.
- Served on conference program committees: Onward! Essays, International Conference on Functional Programming, Practical Aspects of Declarative Languages, Principles and Practice of Declarative Programming.

Talks

- A Galilean Revolution for Computing: Unboundedly Scalable Reliability and Efficiency, 2023.
- Can Tensor Programming Be Liberated from the Fortran Data Paradigm?, 2021.
- Compiling Gracefully, 2020.
- A Functional Reboot for Deep Learning, 2019.
- The simple essence of automatic differentiation, 2018.
- Generic functional parallel algorithms: Scan and FFT, 2017.
- Compiling to Categories, 2017.
- Teaching New Tricks to Old Programs, 2017.
- Generic FFT, 2016.
- Functional programming and parallelism, 2016.
- From Haskell to Hardware via CCCs, 2015.
- The essence and origins of Functional Reactive Programming, 2015.
- A more elegant specification for FRP, 2015.
- Denotational Design: from meanings to programs, 2014.
- Elegant memoization, 2014.
- Understanding efficient parallel scan, 2013.
- Circuit timing analysis, linear maps, and semantic morphisms, 2012.

- ◇ Folds and unfolds all around us, 2012.
- ◇ Reimagining matrices, 2012.
- ◇ Beautiful differentiation, 2009.
- ◇ Tangible Functional Programming: a modern marriage of usability and composability, 2009.

PUBLICATIONS

- ◇ Conal Elliott. Timely Computation. In *Proceedings of the ACM on Programming Languages (ICFP)*, 2023.
- ◇ Conal Elliott. Symbolic and Automatic Differentiation of Languages. In *Proceedings of the ACM on Programming Languages (ICFP)*, 2021.
- ◇ Conal Elliott. Generalized convolution and efficient language recognition. *CoRR*, abs/1903.10677, 2019.
- ◇ Conal Elliott. The simple essence of automatic differentiation. In *Proceedings of the ACM on Programming Languages (ICFP)*, 2018.
- ◇ Conal Elliott. The simple essence of automatic differentiation (Extended version). *CoRR*, abs/1804.00746, 2018.
- ◇ Conal Elliott. Compiling to categories. *Proc. ACM Program. Lang.*, 1(ICFP), September 2017.
- ◇ Conal Elliott. Generic functional parallel algorithms: Scan and FFT. *Proc. ACM Program. Lang.*, 1(ICFP), September 2017.
- ◇ Conal Elliott. Push-Pull Functional Reactive Programming. In *Haskell Symposium*, 2009.
- ◇ Conal Elliott. Beautiful Differentiation. In *International Conference on Functional Programming (ICFP)*, 2009.
- ◇ Conal Elliott. Denotational Design with Type Class Morphisms. Technical Report 2009-01, LambdaPix, March 2009.
- ◇ Conal Elliott. Tangible Functional Programming. In *Proceedings of the International Conference on Functional Programming*, 2007.
- ◇ Conal Elliott. Programming Graphics Processors Functionally. In *Proceedings of the 2004 Haskell Workshop*. ACM Press, 2004.
- ◇ Conal Elliott. Functional Images. In *The Fun of Programming*, "Cornerstones of Computing" series. Palgrave, March 2003.
- ◇ Conal Elliott, Sigbjørn Finne, and Oege de Moor. Compiling Embedded Languages. *Journal of Functional Programming*, 13(2), 2003. Updated version of paper by the same name that appeared in SAIG '00 proceedings.
- ◇ Antony Courtney and Conal Elliott. Genuinely Functional User Interfaces. In *Proceedings of the 2001 Haskell Workshop*, September 2001.
- ◇ Conal Elliott. Functional Image Synthesis. In *Proceedings of Bridges*, 2001.
- ◇ Conal Elliott. Declarative Event-Oriented Programming. In *Principles and Practice of Declarative Programming*, 2000.

◇ Simon L. Peyton Jones, Simon Marlow, and Conal Elliott. Stretching the Storage Manager: Weak Pointers and Stable Names in Haskell. In *Implementation of Functional Languages*, 1999.

◇ John Peterson, Paul Hudak, and Conal Elliott. Lambda in Motion: Controlling Robots with Haskell. In *Practical Aspects of Declarative Languages*, 1999.

◇ Conal Elliott. An Embedded Modeling Language Approach to Interactive 3D and Multimedia Animation. *IEEE Transactions on Software Engineering*, 25(3), May/June 1999.

◇ Conal Elliott. From Functional Animation to Sprite-Based Display. In *Practical Aspects of Declarative Languages*, 1999.

◇ Conal Elliott. A "Fifteen Puzzle" in Fran. Technical Report MSR-TR-98-54, Microsoft Research, 1998.

◇ Conal Elliott. Two-handed Image Navigation in Fran. In *The Glasgow Functional Programming Workshop*, 1998.

◇ Conal Elliott. Functional Implementations of Continuous Modeled Animation. In *Proceedings of PLILP/ALP*, 1998.

◇ Conal Elliott. Composing Reactive Animations. *Dr. Dobb's Journal*, 1998. Also in expanded animated form.

◇ Conal Elliott. Modeling Interactive 3D and Multimedia Animation with an Embedded Language. In *The Conference on Domain-Specific Languages*. USENIX, 1997.

◇ Conal Elliott and Paul Hudak. Functional Reactive Animation. In *Proceedings of the International Conference on Functional Programming*, 1997. In 2007, this paper was awarded as the most influential paper of ICFP '97.

◇ Conal Elliott. A Brief Introduction to ActiveVRML. Technical Report MSR-TR-96-05, Microsoft Research, 1996.

◇ Conal Elliott, Greg Schechter, and Salim Abi-Ezzi. MediaFlow, a framework for distributed integrated media. Technical Report SMLI TR-95-40, Sun Microsystems Laboratories, 1995.

◇ Conal Elliott, Greg Schechter, Ricky Yeung, and Salim Abi-Ezzi. TBAG: A High Level Framework for Interactive, Animated 3D Graphics Applications. In *SIGGRAPH*, 1994.

◇ Greg Schechter, Conal Elliott, Ricky Yeung, and Salim S. Abi-Ezzi. Functional 3D Graphics in C++ with an Object-Oriented, Multiple Dispatching Implementation. In *Workshop on Object-Oriented Graphics*, 1994.

◇ Conal Elliott and Frank Pfenning. A Semi-Functional Implementation of a Higher-Order Logic Programming Language. In Peter Lee, editor, *Topics in Advanced Language Implementation*. MIT Press, 1991.

◇ Conal Elliott. *Extensions and Applications of Higher-Order Unification*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1990. Available as Technical Report CMU-CS-90-134.

◇ Conal Elliott. Higher-Order Unification with Dependent Types. In N. Dershowitz, editor, *Rewriting Techniques and Applications*, 1989.

◇ Frank Pfenning and Conal Elliott. Higher-Order Abstract Syntax. In *Programming Language Design and Implementation*, 1988.

◇ Conal Elliott. A Very General Problem Oriented CAI System. *Behavior Research Methods & Instrumentation*, 14(2):165–169, November 1982.