

# Eliot Miranda

344 Brazil Street  
San Francisco, CA 94112

cell/sms +1 415 533 8868

[eliot.miranda@gmail.com](mailto:eliot.miranda@gmail.com)

[eliotmiranda@acm.org](mailto:eliotmiranda@acm.org)

[www.mirandabanda.org/cog](http://www.mirandabanda.org/cog)

[www.mirandabanda.org/cogblog](http://www.mirandabanda.org/cogblog)

## PROFILE

World-class virtual machine architect. Experienced and creative software engineer with extensive expertise in dynamic languages, language implementation, virtual machines, garbage collection, optimization, machine-level programming, cross-platform portability, language interoperability, object-oriented programming, compilers, debuggers, programming tools, and interactive development environment and framework design. A finisher able to undertake projects from design through to deployment and maintenance. Productive at high and low levels of system design. Leading active participant in open-source Smalltalk community. Able to participate in business development, partner relationships and team management. Past experience as a lecturer in tertiary education and as a research assistant in high-energy physics using holography.

## KEY ACHIEVEMENTS

- Author of the Cog JIT Virtual Machine and closure bytecode compiler for Squeak and Pharo Smalltalks, Newspeak and the Terf virtual worlds teleconferencing system.
- Architect of the Spur memory manager that provides efficient support for live programming, and the Sista speculative optimizer that exists outside of the confines of the virtual machine.
- Technical lead for VisualWorks, the industry-leading commercial Smalltalk platform for nine years, achieving significant revenue growth and profitability.
- Co-author of a high-performance JavaScript implementation (performance  $\geq 4 \times$  Firefox).
- Designer and implementor of the first commercial 64-bit Smalltalk implementation.
- Designer of VisualWorks' parcel system, a uniquely flexible code-loading component model.
- Producer of VisualWorks Non-Commercial, the mainstay of VisualWorks marketing efforts since 1998.
- Inventor of a set of "look Ma, no assembler" C optimization techniques allowing medium-performance VMs to be written in portable C.
- Author of the BrouHaHa series of medium-performance Smalltalk interpreters.

## QUALIFICATIONS

BSc Computer Science, University of York, U.K. 1985

Higher National Certificate in Applied Physics, Oxford Polytechnic, 1979

## WORK HISTORY

### **Architect; Cadence Design Systems. September 2011 to date**

Lead developer of the Cog Smalltalk/Newspeak, Squeak/Pharo/Terf JIT VM; see my blog. Collaborating with a small but beautifully-formed team to provide a high-performance portable execution platform for Smalltalk family dynamic languages.

Designed and implemented new object representation and garbage collector for Cog VM, nicknamed Spur [Spur15,Spur14], featuring a common compact object header between 32- and 64-bit implementations, lazy object mutation for become, compaction, pinning, etc; see my [blog](#). 32-bit Spur achieves a 2 fold performance increase for typical small benchmarks, and early adopters report speedups in the 2x to 6x range for real codes.

Currently co-designing and implementing an adaptive optimizer for Cog called Sista (for Speculative Inlining Smalltalk Architecture) [Sista] with Clément Bera at INRIA Lille. The optimizer exists at the image level, outside of the VM, examining the run-time stack using Smalltalk's intrinsic reification of execution state, "contexts". Sista optimizes from bytecode-to-bytecode, where possible emitting bytecodes that encode for unsafe operations for which the underlying JIT generates short efficient instruction sequences. Hence the optimizer is platform-independent, can save a system in the hot state, and can be interactively developed and tested and upgraded in the field without changing the VM. Sista is on target to be deployed in 2015. This is the same architecture I defined in 2002, finally nearing completion.

Designed and implemented a framework for multiple bytecode sets in the system that allows two bytecode sets to coexist, hence easing development of new sets. The new sets use extension bytes to lift implementation limits such as jump distances, number of literals, etc. This has been applied to a new bytecode set for Newspeak, providing absent receiver sends, and to Sista, providing bytecodes for guards, and unsafe operations [Sista Bytecode].

Completed Cog's JIT back end for 32-bit ARMv6 with a colleague, completing work started in a GSoC project in 2012.

With colleagues, fronted the SoaC application with a web interface. Designed and implemented a line-oriented debugging and conditional breakpointing facility. Designed and implemented a bytecode (dis) assembly system for Newspeak & Squeak bytecode, and applied it to basic-block coverage.

Maintenance, reliability, profiling and performance work. Mentored a Google Summer of Code (GSoC) project to add an ARM back-end to the Cog JIT. Frequent contributor to Squeak development of releases 4.3, 4.4, 4.5 & 4.6.

### **Independent Consultant. March 2011 to September 2011**

Consulting to Cadence on Newspeak. Ported Newspeak to Cog, implementing the first efficient inline caching support for Newspeak's lexically-scoped dynamic send bytecodes, and speeding up Newspeak execution by 3 to 4 fold, depending on usage. Implemented above sockets a foreign function interface including callbacks to interconnect Newspeak/Smalltalk and Perl. Assisted in the development of design-support software for system-on-a-chip (SoaC) development, including advanced debugging support using method wrappers. Maintained the Cog Newspeak VM across three platforms, integrating with licensing software and implementing copy-protection for the SoaC application.

Presented for one and a half days at the 5 day Deep Into Smalltalk workshop at INRIA-Lille, <http://rmod.lille.inria.fr/web/pier/blog/2011-01-23>. Paper on Cog [Cog] accepted at VMIL '11.

Consulting to Teleplace on the Cog VM. . Frequent contributor to Squeak development of release 4.2.

Implemented support for adaptive optimization/speculative execution at the language level in Cog. Support provides for introspecting on inline cache state and counting un/taken conditional branches, combining hot-spot identification with basic block frequency collection.

**Virtual Machine Architect; Teleplace, Inc (née Qwaq, now 3DICC). May 2008 to March 2011**

Designed and implemented “Cog”, a high-performance version of the Squeak virtual machine for Croquet. December 08 to May 09 implemented a simple just-in-time compiler with aggressive inline caching; macro benchmark performance up to 5x of Squeak interpreter; actual application frame-rate increase of 3x. Implemented yet-another-VM-profiler for the Squeak VM. July 08 to October 08 implemented context-to-stack-mapping interpreter as step towards JIT; performance up to 2x of Squeak interpreter. June 08 to July 08 implemented full closures for Squeak, replacing non-reentrant (“BlueBook”) blocks as enabling step towards Stack and JIT VMs, and as improvement over BlueBook blocks. Blogged about the project in a number of extensive posts. The Cog JIT VM was released as open source in June 2010. It is the standard VM for Squeak, Pharo and Newspeak.

Made prototype multi-threaded VM (but not concurrent) to achieve a fast non-blocking FFI.

Implemented a 2nd generation JIT (simple stack to register mapping, constant folding and arithmetic/comparison inlining) that produces an across the board 7%-10% performance increase with certain microbenchmarks showing 2x to 4x increases.

Ported the Teleplace client to linux, dealing with the lack of multiple thread priorities for non-super-user programs, implementing a webcam interface and an X11 screen scraper for application capture.

Implemented file system component of a Windows Terminal Server client for screen-scraping Windows applications into Teleplace.

Guest speaker at Smalltalks 2010 conference in Concépcion del Uruguay, Argentina.

Member of the Squeakland board (<http://www.squeakland.org/about/people/>).

Member of the program committee for the SIGPLAN Workshop on Self-sustaining Systems, Tokyo 2010.

**Architect; Cadence Design Systems, Inc., U.S. January 2007-through April 2008**

Ported the Squeak OSProcess interface to Windows. Provided the Squeak Monticello SCM system with a Subversion backend. Participated in the design of the Newspeak language [Newspeak].

Eliminated closures from a Smalltalk implementation of Combinatorial Parsing, yielding code very similar to BNF [1]. Added multiple panes to the Squeak browser. Extended a novel orthogonal synchronisation scheme to synchronise code as well as data. Added per-object immutability support to the Squeak VM, along with support for primitive error codes. Implemented a minimal FFI for the Squeak VM above Windows, Linux and Mac OS X. Implemented an interface to Objective-C on Mac OS X above the FFI. Added a bytecode for lexically-scoped implicit receiver sends (here sends) to the Squeak VM [2].

## **Independent Project Summer 2006**

Implemented a high-performance JavaScript prototype with Vassili Bykov and Claus Gittinger as a “vacation project” using Extreme Programming (test-driven development with refactoring). Performance at least 4 times faster than Firefox across a small range of [computer language shootout](#) benchmarks. System maps JavaScript to Smalltalk, mapping properties to slots by analysing constructors and creating classes for objects created by suitable constructors.

## **Technical Lead for VisualWorks; Cincom Systems, Inc., U.S. September 1999-to December 2006**

VisualWorks is the industry-leading commercial Smalltalk development platform with customers such as JPMorganChase and AMD; see [www.cincomsmalltalk.com](http://www.cincomsmalltalk.com). Continued to lead the VisualWorks engineering team through recovery from VW5i nadir<sup>1</sup> to VW7.x renaissance achieving significant revenue growth and profitability. VisualWorks team recognized within Cincom as having the best development process in the company. Won protracted political battle to retain and enhance team and equipment in Santa Clara (Cincom is based in Cincinnati, Ohio). Motivated incorporation of Refactoring Browser and Professional Debug Package into VW. Designed and implemented first commercial 64-bit Smalltalk platform providing access to a 64-bit address space at the cost of only 15% in performance and 50% growth in memory footprint, compared to the 32-bit platform on x86-64. Added extension bytes to bytecode set, lifting various limits on number of literals, branch span, etc. Grew number of VW production platforms from 7 to 19. Provided significant performance increases through Polymorphic Inline Cache and Garbage Collector work. Added Ephemeron (application-oriented weakness facility) and Immutability support. Presented at Stanford University on history of Smalltalk implementation and Adaptive Optimization. Designed and implemented infrastructure for bytecode-to-bytecode adaptive optimization.

## **Technical Lead for VisualWorks; ParcPlace-Digital / ObjectShare, Inc, U.S. 1997-99**

Architected VisualWorks 3.0, the first major release since the ParcPlace-Digital merger of 1995. The system is now focussed around “parcels”, a binary-code loading technology to which I added unique compositional and conflict-resolution facilities which support maintenance and programming in the large (see [Parcels]) and the ability to carry source. Implemented the Parcel Browser, an extension of the Smalltalk browser for authoring parcels. Invented “menu pragma” scheme which avoids the combinatorial explosion of different colliding extensions of the application by separate loadable modules. Implemented a prototype of the VM as a DLL and used it to implement a prototype web browser plugin. Led the company to release VisualWorks Non-Commercial in 1998, which has been the mainstay of marketing efforts from then on. I wrote the introductory workspace, added copy-protection facilities to allow full product to be released non-commercially, and co-wrote the Linux x86 port. Rearchitected VMs’ implementation of contexts and blocks (reified stack frames and anonymous functions), achieving significant performance increases (see [Contexts]). Motivated work on Namespaces and StORE, a parcel-oriented team-programming tool, which together defined the VisualWorks 5i release (Q2 1999) and freed VisualWorks from ENVY, a third-party team-programming tool bought by IBM in 1996.

---

<sup>1</sup> ParcPlace timeline: ParcPlace and Digital merge in August of 1995 as ParcPlace-Digital. PPD renamed to ObjectShare in 1997. ObjectShare’s Smalltalk team and technology sold to Cincom Systems, Inc in August of 1999. Same desk, four company names.

**Lead VM Engineer; ParcPlace-Digital, Inc, U.S. 1996-97**

Released THAPI work in vw2.5.2, the first VisualWorks release after the PPD merger, on Windows NT, Solaris and Digital UNIX. Implemented heap growth and shrinkage via memory-mapping. Implemented interactive statistical profiler for VM.

**VM Engineer; ParcPlace/ParcPlace-Digital, Inc., U.S. 1995-96**

Extended Foreign Function Interface (FFI) to allow call-outs and call-backs on separate threads, allowing green-threaded VM to implement non-blocking server applications (see [THAPI]). Extended FFI to handle 64-bit datatypes. Productised Digital UNIX AXP port (a 32-bit implementation on a 64-bit OS). Extensive work on the ChangeList, VisualWorks' source file browser and conflict checking tool. Improved Deutsch-Schiffman in-line cache implementation, achieving a ten-fold speedup on Takeuchi benchmark.

**Engineer; Harlequin Ltd, France & U.K. 1994-95**

Engineer in Harlequin's Dylan team. Co-designed and implemented a threaded interpreter for interactive use. Designed and implemented binary storage (object streaming) system. Designed and implemented a comprehensive stream library. Co-wrote two technical reports for the ESPRIT OMI/GLUE project reviewing single-address-space garbage collection techniques.

**Visiting Researcher; IRCAM, France Autumn/Fall 1993**

Disastrous and entirely unproductive sabbatical at avant-garde classical and electronic music research lab.

**Lecturer; Dept Comp Sci, Queen Mary College, University of London, U.K. 1990-93**

Provided BrouHaHa to various university departments in the U.K. Wrote second dynamic translator to direct-threaded-code version of BrouHaHa. Ported core engine to SUN4 SPARC in under 1 day. Performance on standard macro benchmarks 70% of ParcPlace's 2nd generation hps VM (the VisualWorks VM). BrouHaHa used for distributed systems research and teaching inside Queen Mary. Developed and taught MSc-level courses on Object-Oriented Programming and User Interface Management Systems. Developed and taught an undergraduate OOP course. Consulted for Harlequin Ltd for their Dylan project. Designed and implemented a conflict-resolution extension for the Ingalls-Borning Smalltalk-80 Multiple Inheritance scheme. Wrote various interactive applications including a terminal emulator and a chess interface.

**Research Assistant; Dept Comp Sci, Queen Mary College, University of London, U.K. 1986-90**

Worked on a What-You-See-Is-What-I-See shared desktop prototype for IBM Hursley Labs. Initially used SoftSmarts Smalltalk on IBM PC AT but soon moved to BrouHaHa for the bulk of the project (4 to 5 people for 2 years). Did the first true closure implementation in Smalltalk-80, beating ParcPlace by a year. Performance of the Sun 3 port was equal to the Tektronix commercial implementation (largely written in assembler) due the use of global register variables. Added colour graphics support via my implementation of dynamically-compiled BitBLT (RasterOp). Implemented BitBLT in C, using macro and assembler-post-processing scheme to get global register variables, to remove prologs and epilogs from functions implementing code fragments, and then at run-time concatenating relevant fragments to construct a BitBLT invocation on the fly. Ported to AIX on IBM PCs, AUX on 680x0 Macs and (with Tim Rowledge) to the Acorn RISC machine (ARM). Wrote FAT16 support to allow reading DOS floppies. The ARM port was used by Active Book Ltd (which was subsequently bought by Go Corp./EO Inc) to implement a prototype tablet handheld. Added unwind-protect and ported ParcPlace's

exception system to support ActiveBook. Wrote first dynamic translator to direct-threaded-code version of BrouHaHa. Invented macro and assembler-post-processing scheme to allow writing the direct threaded-code interpreter in C.

**Research Assistant; Department of Computer Science, University of York, U.K. 1985-86**

Worked on the implementation of a formally-specified folding editor. Skunk-worked on BrouHaHa, my Smalltalk-80 interpreter [BrouHaHa], porting to the Sun 3. Did a dynamically-compiled BitBLT implementation for the WCW workstation written in C that generated ns32016 machine code. Invented a scheme using macros and assembler post-processing to keep important interpreter global variables in registers, achieving significant performance increase (~40%) in VM written entirely in C.

**Undergraduate; Department of Computer Science, University of York, U.K. 1982-85**

Summer job at end of first year at Rutherford Lab learnt C and implemented a “Blue Book” Smalltalk-80 interpreter on an ICL PERQ. Final year project was a Smalltalk-80 interpreter on a WCW ns32016-based workstation.

**Scientific Officer; Bubble Chamber Research Group, Rutherford-Appleton Laboratory, U.K. 1980-82**

Designed holographic cameras for the Big European Bubble Chamber (BEBC) and the Little European Bubble Chamber (LEBC). Implemented ray-tracing programs in FORTRAN and Pascal and used them for the camera designs. Had the idea to replay holograms through the BEBC lenses to eliminate their distortion on replay, hence raising resolution. Programmed an Intel SDK-85 8085 evaluation board loaned from Oxford Polytechnic to play Air on a G String in two parts given the board’s two programmable interval timers and 256 bytes of RAM.

**Assistant Scientific Officer; BCRG, Rutherford-Appleton Laboratory, U.K. 1978-80**

Worked on various LSI and VLSI components for the BCRG’s measuring network of 8086-based measuring tables. Wire-wrapped a 64kb dynamic ram board. Helped design and build a display unit diagramming the network for an open day. Learnt 8086 machine code.

**Undergraduate; Department of Physics, Imperial College, University of London, U.K. 1977-78**

Failed first year and learnt to play table football.

## **EDUCATION**

University of York, UK                    BSc Computer Science, Iii, 1985  
Oxford Polytechnic, Oxford, UK Higher National Certificate, Applied Physics, 1979

## **PROFESSIONAL MEMBERSHIPS**

Member of the Association for Computing Machinery (ACM) since 1988.

## **SKILLS**

Virtual Machine Design and Implementation, including JIT Compilers, Garbage Collectors, Foreign-Function Interfaces, 64-bit implementation, machine-level programming and optimization.

Object-oriented and functional language design and implementation.

IDE and OO Framework Design and Implementation.

Technical leadership.

Open source.

Effective and articulate technical and business communication.

Subsistence-level French.

Guitar playing. Music composition (jazz).

### **Programming Languages:**

Smalltalk, Newspeak, Objective-C, CLOS, Dylan, JavaScript, Ruby, Perl,  
C, C++, Objective-C, Pascal, Fortran,  
Iswim, Miranda (lazy functional languages), Lambda Calculus, Prolog,  
Bourne Shell, Many Assembly Languages and Machine Codes (x86, x86-64, ARM, PowerPC, etc).

### **Operating Systems:**

Unix (various including SunOS, Solaris, IRIX, AIX, Linux).  
Windows (various, including CE)  
Mac OS X, Mac OS 9, A/UX. RISC OS, VMS.

## **INTERESTS AND LEISURE ACTIVITIES**

Snorkeling, Skiing, High-Performance Driving. Camping/Hiking, Guitar playing/Jazz Composition, Fiction (esp. Nadine Gordimer, Martin Amis, Zadie Smith, Umberto Eco, John le Carré). Non-fiction (esp. History of Human Development, History of Life). Browsing Wikipedia, Cinema, Programming.

## **PERSONAL**

Born 21st January 1959, Liverpool, U.K.

Male

Divorced with 2 children (12/2000, 1/2004)

British National

U.S. Green Card Holder

No convictions other than my own

## **REFERENCES**

Available on request

## PUBLICATIONS PRESENTATIONS AND POSTINGS

[blog] <http://www.mirandabanda.org/cogblog>

[Spur15] Eliot Miranda, Clément Béra , “A partial read barrier for efficient support of live object-oriented programming”, Proceedings of the 2015 ACM SIGPLAN International Symposium on Memory Management, 2015

[Spur14] Presentation to European Smalltalk Users Group conference in Cambridge, UK, August 2014. Video: <https://www.youtube.com/watch?v=k0nBNS1aHZ4>

Slides: <http://www.slideshare.net/esug/spur-a-new-object-representation-for-cog>.

[Sista Bytecode] ibid. Slides: <http://www.slideshare.net/esug/a-bytecode-set-for-adaptive-optimizations>

[Cog] “The Cog Smalltalk Virtual Machine. writing a JIT in a high-level dynamic language”, Workshop on Virtual Machines and Intermediate Languages, co-located with SPLASH 2011, Portland, Oregon.

Presentation on Code Generation in Cog at Smalltalks 2010 Conference, Concepción del Uruguay, Argentina, November 2010 <http://www.fast.org.ar/smalltalks2010/videos/Code+generation>

Presentation on Inline Cacheing in Cog, ibid, <http://www.fast.org.ar/smalltalks2010/videos/Inline+caching>

[Newspeak Modules] Gilad Bracha, Peter Ahe, Vassili Bykov, Yaron Kshai, William Maddox, Eliot Miranda, “Modules as Objects in Newspeak”, Proceedings of ECOOP 2010

[Newspeak] Gilad Bracha, Peter Ahe, Vassili Bykov, Yaron Kshai, Eliot Miranda, “The Newspeak Programming Platform”, <http://bracha.org/newspeak.pdf>

[Parcels] Eliot Miranda, David Leibs, Roel Wuyts, “Parcels: A fast and feature-rich binary deployment technology”, Computer languages, Systems and Structures, 31 (2005), pp 165-181. Elsevier Press.

Eliot Miranda, “Bytecode-to-bytecode adaptive optimization for Smalltalk; Compilation and execution architecture for late-bound object-oriented programming languages”, Stanford University Computer Systems Laboratory Colloquium Guest Lecture, March 2003, <http://video.google.com/videoplay?docid=-8988857822906068209>

[Contexts] E. Miranda, “Context Management in VisualWorks 5i”, presented at the OOPSLA '99 workshop on Simplicity, Performance and Portability in Virtual Machine Design, Denver, CO, November 1999. <http://www.mirandabanda.org/files/oopsla99-contexts.pdf>

[THAPI] Eliot Miranda, “VisualWorks Threaded Interconnect”, VisualWorks product documentation, ParcPlace-Digital, Inc., January, 1997.

E. Miranda, "On the Visibility and Inheritability of Smalltalk Methods.", presented at the OOPSLA '96 workshop on Extending Smalltalk, San Jose, CA, October 1996.

E. Miranda, T. Mann, "Requirements specification for parallel extensions to TDF to support Dylan - including run-time support.", ESPRIT Project 6062 OMI/GLUE. Deliverable 5.7.1, Harlequin Ltd., Cambridge, UK, 1994.

E. Miranda, T. Mann, "Initial evaluation of TDF Support for Garbage Collection", ESPRIT Project 6062 OMI/GLUE- Deliverable 4.2.2a, Harlequin Ltd., Cambridge, UK, 1994.

A. Balou, E. Miranda, "SPIRIT Multiprocessor Smalltalk", Department of Computer Science, Queen Mary Westfield College, Departmental Report no 627, 1993.



[TCODE] E. Miranda, "Portable Fast Direct Threaded Code", USENET posting to comp.compilers, published in "Selections from comp.compilers", SIGPLAN Notices, Vol. 327, No. 1, Jan. 1992, ACM

Abdullahi, S. Miranda, E. Ringwood, R. "Distributed Garbage", Proceedings of the International Workshop on Memory Management, St. Malo, INRIA, IRISIA, ACM, September 1992.

J. Dollimore, E. Miranda, Wang Xu, "The Design of a System for Distributing Shared Objects", Computer Journal Special Issue on Distributed Systems, Vol. 34, No. 6, Dec 1991

M. Slater, K. Drake, A. Davison, E. Kordakis, A. Billyard, E. Miranda, "A Statistical Comparison of Two Hidden Surface Techniques: the scan-line and z-buffer algorithms", accepted for 1991 EUROGRAPHICS conference sponsored by the European Association for Computer Graphics

M. Slater, K. Drake, A. Davison, E. Kordakis, E. Miranda, "The Graphics Subsystem of the Spirit Workstation" Proceedings of the Graphics and Interaction in ESPRIT sessions, Eurographics 89, Hamburg FRG, 1989.

E. Miranda, S. Cook, "Smalltalk on Transputers", Department of Computer Science, Queen Mary Westfield College, Departmental Report no 490, 1989.

[BrouHaHa] E. Miranda, "BrouHaHa - A Portable Smalltalk Interpreter", pp. 354-365, Proceedings of OOPSLA '87, vol. 22, no 12, SIGPLAN Notices, ACM, December 1987

A.J. Dix, M.D. Harrison, E.E. Miranda, "Using Principles to Design Features of a Small Programming Environment", Proceedings of a Software Engineering Environments Conference, Lancaster University, I. Sommerville, Ed., 1986

H. Bjelkhagen, F. Pouyat, P. Kasper, E.E. Miranda, R.L. Sekulin, W. Venus, L. Walton, "Tests of High Resolution Two-Beam Holography in a model of the Big European Bubble Chamber, BEBC", pp. 300-308, Nuclear Instruments and Methods in Physics Research, no. 200, Elsevier Holland, 1984

C.M. Fischer, G. Homer, J.D. Lewin, E.E. Miranda, R.L. Sekulin, P.F. Smith, "Tests of In-Line Holography for Use in a Small High Resolution Bubble Chamber", pp. 49-67, Proceedings of a Meeting on the Application of Holographic Techniques to Bubble Chamber Physics, Rutherford Appleton Laboratories, RL-81-042, R.L. Sekulin, Ed., May 1981

C.M. Fischer, E.E. Miranda, V. Peterson, R.L. Sekulin, "Laboratory Tests of Holography for Large Bubble Chambers", pp. 181-200, *ibid.*

C.M. Fischer, E.E. Miranda, R.L. Sekulin, "Remarks Concerning Holography for RCBC", pp. 203-214, *ibid.*

## **OTHER PAPERS REFERENCED**

[1] Gilad Bracha. Executable Grammars in Newspeak. Electronic Notes on Theoretical Computer Science, Volume 193C, November 2007 pp. 3-18. (<http://bracha.org/executableGrammars.pdf>)

[2] Gilad Bracha. On the Interaction of Method Lookup and Scope with Inheritance and Nesting. 3rd ECOOP Workshop on Dynamic Languages and Applications (2007). <http://dyla2007.unibe.ch/>

[Sista] Clément Bera, Presentation to European Smalltalk Users Group conference in Cambridge, UK, August 2014. Video: [https://www.youtube.com/watch?v=X4E\\_FoLysJg](https://www.youtube.com/watch?v=X4E_FoLysJg)  
Slides: <http://www.slideshare.net/esug/sista-talkesug2>.

*Talk, it's only talk,  
Arguments, agreements, advice, answers, articulations, announcements.  
It's only talk.*

*Talk, it's only talk,  
Babble, burble, banter, bicker, bicker, bicker,  
Brouhaha, balderdash, ballyhoo.  
It's only talk  
Backtalk!*

*Talk, talk, talk, it's only talk,  
Comments, cliches, commentary, chatter, chit chat,  
Conversation, contradiction, criticism,  
It's only talk.  
Cheap talk!*

*Talk, it's only talk  
Debates, discussions, these are words with a "d" this time...  
Dialogue, duologue, diatribe, dissension, declamation!  
Double talk,... double talk.*

*Talk, talk, it's all talk  
Too much talk, small talk, talk about trash  
Expressions, editorials, explanations, exclamations, exaggerations  
It's all talk!*

*[Elephant talk, elephant talk, elephant talk!](#)*

*Adrian Belew  
Discipline, King Crimson, EG Records, 1981*