



Making Do and Getting By:

Software and Design

By David Reinfurt

TABLE OF CONTENTS

- 1 Two Prototypes
- 3 Spray-Cans, Hair-Lines, Machine-Guns and Other Not-So-Soft-Wares
- 7 Plug-In Architectures

Graphic design now happens on and with a computer—we spend our working day in front of a screen where fundamental production processes are integrally bound with the features and methods of specific softwares. With the emergence of computer-aided design, the relationship between the tools and the finished product has been obscured by a black-boxing of the design process. However, software is not nearly as sealed and impenetrable as it might appear. Poking around this complex assembly, taking apart existing programs and writing new code offers a way out. The designer who also knows about software can reclaim an intimate relationship with the production of their work by modifying, creating, intentionally misusing, extending and breaking existing programs.

Two Prototypes

Anthony Froshaug (1920–1984) and Muriel Cooper (1926–1994) are two early models for an emerging designer hybrid who mixes formal strengths with an intuitive and critical relationship to their tools. Both Froshaug and Cooper began designing books; both were teachers; and both were critically engaged with how they made their work—sensitive to the daily conditions and particularities of their work environments. Both were committed to a continuous examination of their tools, forging a close bond to their work with spectacular results.



Fig. 1: Anthony Froshaug, from Royal College of Art newsletter, c.1962.



Fig. 2: Muriel Cooper using an SX-70 for self-portrait, 1977.

Anthony Froshaug practiced graphic design many ways—beginning as a graphic designer and small-press publisher in London, then moving to Cornwall and buying a press and setting up as a printer, and teaching at the [Hochschule für Gestaltung, Ulm](#), The Central School and London College. Throughout, he maintained a deep and persistent exploration into the tools and methods that surround the practice of graphic design. His trajectory took him from an intimate study of typography and typesetting, to letterpress and offset printing, to phototypesetting and not surprisingly to the computer. Froshaug passed away in 1984, the same year that the Apple Macintosh was introduced. ⁽¹⁾

“I think it was the string-and-sealing-wax approach I liked mainly; and I liked it particularly in relation to computers, which nowadays contain too much hardware for anyone outside the big organisations to afford.” (2)

Throughout his life, Froshaug was fascinated by computers. He studied their history from the first nineteenth-century attempts to Charles Babbage to the IBM 360. With the introduction of the tiny and affordable Sinclair ZX80 in 1980, Froshaug finally could get his hands into a computer to poke, prod and meticulously examine how it works. Froshaug jumped in head-first, spending hours making painstaking handwritten records and schematics detailing the contents of individual memory registers in his ZX80. Froshaug even introduced the computer in his teaching, leading London College of Printing students through exercises in designing a typeface appropriate for electronic display.

In his essay, *Two Antitypes*, Froshaug suggested, “It is sad that Sinclair has not proclaimed the identity of electronic devices and the ingenuity of his circuitry by the use of a transparent acrylic case.” Tellingly, on receipt of his ZX80, Froshaug promptly enlisted the Industrial Design department at The Central School to fashion him a clear cover for his very personal computer. (3)

Muriel Cooper also spent a career critically examining the tools of her trade. Beginning as a book designer, then as a fellow at the MIT Center for Advanced Visual Studies, serving as Design Director of MIT Press, and co-founding (with Ron MacNeil) the Visible Language Workshop in the MIT Media Lab, Cooper wasn't so much interested in computers, but rather what one could do with them. In the Visible Language Workshop, Muriel led a team of graduate students and researchers in the relentless pursuit of new forms, techniques, and methods for graphic design that were specific to the emerging context of the computer screen. The work produced there from 1975–1994 forms the foundation of contemporary interactive design practice.



Fig. 3: Muriel Cooper, Poster for MIT Center for Advanced Visual Studies, 1972.

Cooper's desire to confront the conditions (daily, mundane and technical) of graphic design production was consistent throughout her career. Her 1972 resume sounds as if it was written by William Morris in 1872: “Interests and Goals: The significance of participatory and non-authoritarian communication forms in relation to specialization and professionalism... Direct, responsive means of reproduction.” Designing books for MIT Press, including the first edition of *Learning from Las Vegas* and the MIT Press logo, she experimented consistently with the forms and methods of book design and production. As she described, “So I had a little support for this R&D unit at MIT Press... We did some stuff with rubber stamping, cut and paste—it was the Whole Earth Catalogue era. There was a lot of Method Acting in what I was doing.” (4)

Describing the graphic design of Herbert Muschamp's first book, *File Under Architecture*, Muriel Cooper said: "This was my favorite book. Very innovative, in the mid-70s. It was done on brown wrapping paper and set on an IBM composer, which was a typewriter (designed by Eliot Noyes) that has a head with a type ball on it. The ball let you change typefaces. It was very tedious of course—change the ball to get bold type." (5)

It is this sensitivity to the daily tools and circumstances of graphic design practice which connects Anthony Froshaug to Muriel Cooper and to all of the designers in this article. Cooper used a typewriter, rubber stamps, cut and paste, and (eventually) computer code, while Froshaug learned letterpress, metal type, typography and cheap computers; this mode of practice continues today. And since contemporary graphic design necessarily happens on and with the computer, these designers are confronting the computer on its own terms and in its own language.

Spray-Cans, Hair-Lines, Machine-Guns and Other Not-So-Soft-Wares

Jürg Lehni is a young Swiss designer / programmer and a prime model for this emerging hybrid practice. Through small interventions and ambitious projects, he is prying open the closed doors of commercial graphic design software and making room for others. One such project is *Scriptographer*, a small program that Jürg created as a Plug-In for Adobe Illustrator.

Jürg describes the intention concisely on his website (www.scratchdisk.com): "Scriptographer gives the tool back in the hand of the user, it confronts a closed product with the open source philosophy. And the best thing is: it doesn't cost anything." Scriptographer is a Plug-in that allows Illustrator to be programmatically controlled (scripted) through a simple interface and existing JavaScript syntax. This lightweight software allows the development of completely new drawing tools integrated into designers' existing daily design work. Jürg claims, "by automating certain parts of the workflow, new ways of working can be discovered, and new aesthetics will result." Free scripts are posted by designers on www.scriptographer.com for download, use, misuse and modification. Jürg is currently working on an all-new version 2.0 of his Plug-in.

Scriptographer led Jürg to Hektor, a large-format printer which is scaleable, mobile and spectacularly particular. Together with engineer Uli Franke, Jürg designed and built Hektor while completing his studies at the École Cantonale d'Art de Lausanne. While requiring formidable technical skills to design and build a spray-painting vector format printer and print driver software, the result is satisfyingly ad-hoc. Hektor is composed of a laptop computer running Illustrator with the Scriptographer Plug-in, micro-controlled stepper motors, some cable, pulleys and a can of spray paint.

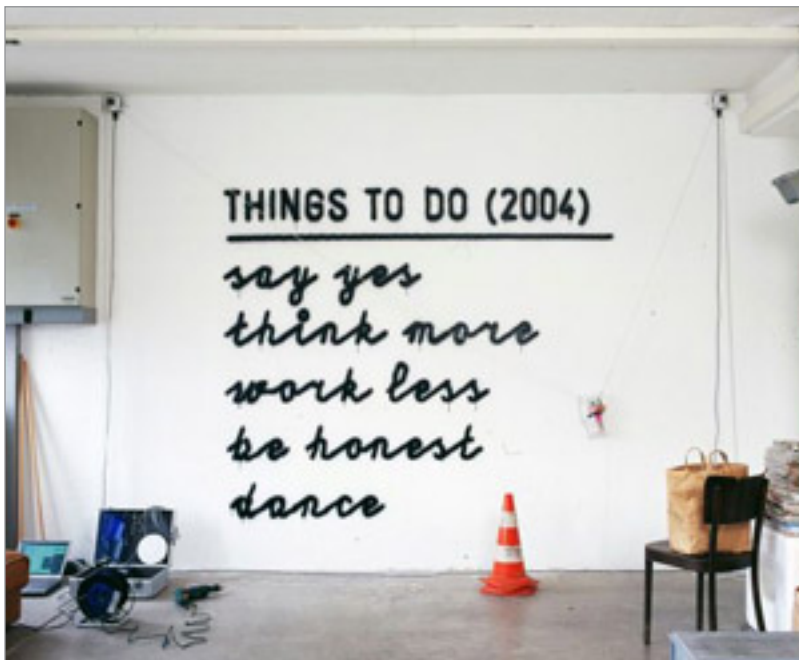


Fig. 4: Hektor with Cornel Windlin and Joel Nordström, *Things To Do* 2004, 2003.

Hektor has been used to produce wall-murals in studios, galleries and for magazines—all from industry-standard Illustrator vector graphics. Prowling along the wall in loops and starts, the spraycan writes large and precise graphics in refreshingly lo-fi spray-paint. Further, the mix of a low and high technologies in this oversized plotter combine in a potent mix where both the process and result are equally satisfying.

Yet, Hektor met its match in Amsterdam-based designer Will Holder. For Tourette's II, a week-long program of performances, lectures, films and music organized with designer Stuart Bailey in Amsterdam (2004), Will challenged Hektor, conspiring with William Morris and Adobe Illustrator to test the good printer's patience.



Fig. 5: Hektor with Will Holder, Hektor Meets William Morris, for Tourette's II, Amsterdam, 2003. [Click to see movie](#)

Walking for several days in Corsica and reading William Morris' Utopian novel, *News from Nowhere*, Will found Compton, an intricate wallpaper pattern designed by John Henry Dearle for Morris & Co. in 1896, specifically to be made by machine. Originally requiring sixteen colors of ink, printed in meticulous succession and intense registration that only a machine and master printer could achieve, the pattern was a perfect test of Hektor's limits. He quickly contacted Jürg to make his proposal: "Frantically sms'ing with Jürg from Corsica (he was bluetoothing his answers, so fast!)" (6) Hektor had never painted that large of a surface and couldn't handle this detailed and colorful image, but Will insisted. A compromise was reached and Will prepared an Illustrator drawing.

Three weeks later, Hektor arrived in Amsterdam to begin printing while negotiations continued between what Will wanted and what Hektor could produce. Throughout, Jürg played negotiator, the harried technician attempting to solve technical issues and negotiate the form. Eventually, the Compton pattern was simplified and broken into a repeating pattern to be painted each of four nights. After each night, Jürg was rewriting (slightly) Hektor's software and biking around Amsterdam looking for better-smelling paint. The resulting man versus machine printing and painting performance recalls John Henry and feels absolutely contemporary. Over these four evenings, Hektor covered the wall in drips and runs, precise and messy simultaneously, neither winning nor losing the battle and producing a beautiful wallpaper which couldn't be made at any other time or in any other way.

Another artist / designer working in New York City used one printing accident to produce a stunning series of wallpaper installations. At the computer in her Red Hook studio, Kara was trying to make a simple line drawing with Adobe Illustrator, a software she knows but perhaps has not mastered. On screen, she was looking for the finest line weight and switched to an older version. Trying Illustrator 3.2, Kara selected the Hairline weight for all of her lines, which simply provides the finest line possible rather than a specific point width. On finishing the drawing and making a laserprint, she was surprised and pleased. With the outdated software, a current print driver and a fussy laserprinter, the resulting drawing was almost invisible on the page — ephemeral and shining. Eventually, Kara expanded these hairlines to the environmental scale of wallpaper, realizing an effect that is both tentative and encompassing. The wallpaper was not a result of the technical mastery of complex softwares, nor of intense design, but rather an everyday mistake and the sensitivity to honor it.

Min Choi, a designer in Seoul, Korea, is equally sensitive to the repetitive, always the same, always slightly different, quality of days spent in front of a computer making graphic design. One day sitting in front of his computer, Min realized that every time you type a word / sentence on the keyboard, the resulting sound of key-clicking is unique to that word, to that moment, to that instance. Min used this idea to design and program a custom software application which generates the performative typeface, Type-Machine-Gun. (www.minch.org/tmgdemo.html)

Min describes Type-Machine-Gun as a “sound-sensitive ‘word processor’ software by which one can create dynamic word-images. Connected to the computer’s microphone, it generates and distorts letters on screen according to how hard you stroke the keyboard.” The louder/ harder/ and quicker that a user types the keys, the more distorted the forms of the letters become. The result is a concrete typography which is a record of its own making. Each time, a particular word will necessarily yield a unique typesetting. His typeface program exploits several characteristics of its software context: each letter is continuously different, words are generated on the fly with the performance and software smoothly modulating the form.

Detroit-based designer Danielle Aubert has also completed a series of software-driven, daily drawings using Microsoft Excel. This spreadsheet program, workhorse of accountants, project directors and middle managers, seems like a particularly unlikely place to begin a free-ranging graphic design exploration. However, in this work completed over the first 5 months of 2005 while a graduate student at Yale University School of Art, Danielle uncovered the specific graphic possibilities in color coding, patterning and using the excel spreadsheet as a canvas. By undertaking a free but consistent exploration and intentionally misusing a particular software, Danielle reveals how a relentless and undirected approach can yield unexpected results. 58 Days Worth of Drawing Exercises in Microsoft Excel required no particular programming skills nor technical flair, but rather an open idea, a specific agenda and a critical relationship to the software designers use daily.

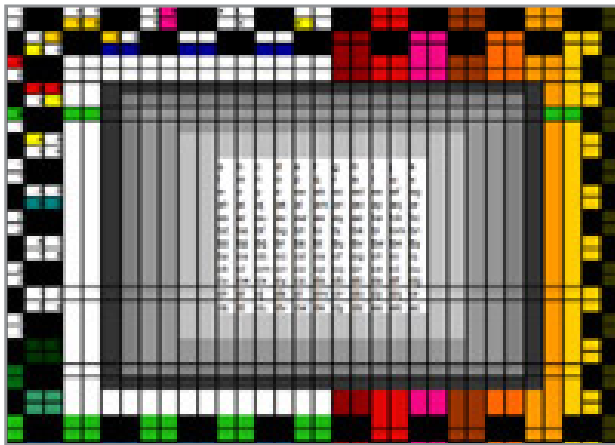


Fig. 8: Danielle Aubert, 58 Days Worth of Drawing Exercises in Microsoft Excel, April 2004.

Finally, a graphic practice based in The Hague, Netherlands seems to encompass many of these new ideas, while mapping out an entirely new terrain for graphic design, for software, and for new audiences. Lust is a five-person graphic design practice established by Thomas Castro, Jeroen Barendse and Dimitri Nieuwenhuizen. They work in a variety of media including printed materials, interactive installations and architectural graphics, and describe their practice as revolving around process-based design and coincidence.

Recently, Lust was commissioned by The Hague art cinema (Filmhuis) to transform their public facade into a map and information screen. Without the budget for an LCD or LED installation nor the desire to use the window as an opaque screen (blocking the view of the town square below), they came up with an ad-hoc solution that mixes low technology and high concept to maximum effect. By applying semi-opaque vinyl film squares in a grid pattern that represents The Hague, and using a standard high lumen projector, the facade could be animated simply. The resulting animations are easy and direct as well, since nothing needs to move, as only the individual vinyl squares light up, tracing current activity in the city.

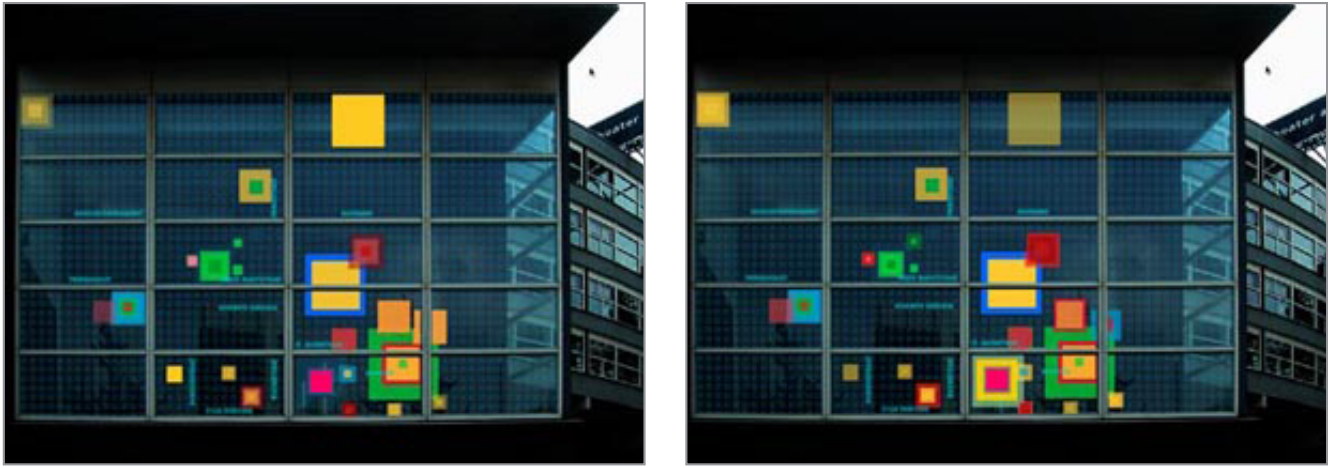


Fig. 9: Lust (Thomas Castro, Jeroen Barendse, Dimitri Nieuwenhuizen), facade installation for Filmhuis, The Hague, 2005.

Turning to the front page of their website (www.lust.nl) suggests which direction Lust is headed. In the place of a welcoming graphic or descriptive text, we see simply the current computer desktops of the five studio members, combined, sliced and mediated through an intense pixelation process. Installed as a complex but lo-fi studio monitoring system, a software program called EvoCam watches, records, and distorts the desktops every five minutes as Lust members continue with their daily work. These screenshots are then processed using a custom Macromedia Shockwave software program that Lust developed to slice, block and create the mixed desktop images on-the-fly. Logging onto their website right now, five live desktops combine to make one composite image—what we see is the current work, the current software, and an always-changing, software-specific picture of a new graphic design practice.

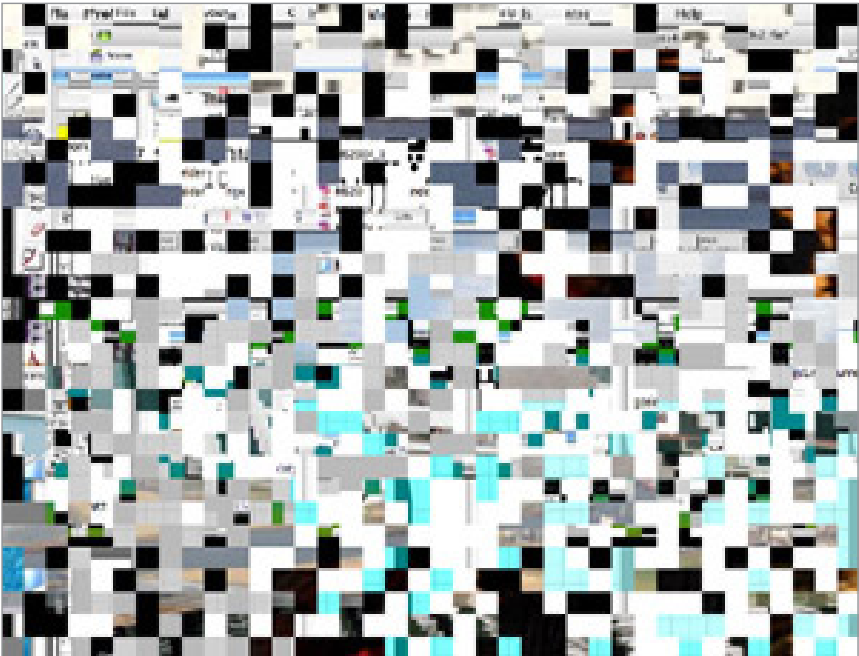


Fig. 10: Lust (Thomas Castro, Jeroen Barendse, Dimitri Nieuwenhuizen), Desktops The Hague, 2005.

Plug-In Architectures

Perhaps a new hybrid, impure and resistant graphic practice is emerging between what was writing, programming and performance. Still, we continue to use the same softwares every day—a range that might be so varied as to include the Adobe Photoshop-InDesign-Illustrator trinity, Quark XPress, and Macromedia Flash. As Jürg tells it, the same applications are used daily “by nearly all the graphic designers all over the world.” Function sets, software paradigms, and user scenarios are mapped out for each software project to ensure the widest possible usability, resulting in an averaged tool which skips the highs, lows, errors, and quirks which might make such a complex software more productive.

Previously, the tools we used as graphic designers were either functionally specific and simple (a pencil), or complex and generic (a camera), but not both specific and complex at the same time. This is the problem with our current software tools: an overdetermined functionality is combined with Byzantine complexity to produce industry-standard graphic design software for users who rarely feel emboldened to proceed beyond the standard tools and techniques. And, of course, this yields graphic design work which rarely exceeds the mannerisms of its software environment.

However, recall that all software is made from modular and miscellaneous pieces and in these soft fissures between code libraries, applications, and versions, opportunity remains. The major software companies realize this. This modular and flexible system implicit in software reveals itself as the Plug-In Architecture standard in graphics programs, web browsers, code compilers, database applications, and even games. A Plug-in is a small computer program that works within another software to extend or enable a specific function. They can be written to add new features (a logorhythmic twirl filter for Photoshop), to extend existing functionality (a print-driver for a spray-can enabled inkjet printer), or even to change the interface (three-dimensional articulated window-frames in Quark XPress).

The first Plug-Ins appeared around 1988 in SuperPaint 1.0 from Silicon Beach software. Shortly afterwards, with version 0.87, Photoshop added the Plug-In Architecture. (7) Michigan and California-based photographers and programmers Thomas and John Knoll wanted to be able to make small and fast changes to their new image manipulation software without changing the considerably more complex application code. Soon, this robust and truly new tool included a wide range of third-party filter effects that contributed tremendously to its success. Of course now produced by Adobe, Photoshop controls the market and, like Xerox or Kleenex before, has even entered our contemporary lexicon.



By making room for hackers and hobbyists in commercial programs via a Plug-In architecture or even by releasing the application source code wholesale (as with Mozilla Firefox or Mac OS X), large software companies have acknowledged that software might actually be made better by large groups of distributed tinkerers. Linux, an Open Source operating system, began with a lonely post by Finnish programmer and student Linus Torvalds: “I’m doing a (free) operating system (just a hobby, won’t be big and professional like gnu)”. Fourteen years later, the phenomenal success of this Free Software project offers concrete proof that substantial software can be effectively developed, used and revised in an ad-hoc manner by a dispersed group of interested individuals. And this is the opportunity for contemporary designers who currently rely on industry-standard commercial softwares. Individually, we can commit to using software critically, engage the mechanics of production, ask questions, use old software, share, write new software and refuse to passively consume the latest innovations. Like Muriel, Anthony, Will and Hektor have suggested, this might actually lead us back—closer to our work.

ABOUT THE AUTHOR

David Reinfurt runs O R G inc., a graphic design practice that works for cultural and educational institutions in a range of media. David also teaches in the Interactive Telecommunications Program at New York University and at Yale University. Visit: www.o-r-g.com

FOR MORE INFORMATION

Visit: www.adobe.com/designcenter/main.html

Notes

- (1) Robin Kinross, Anthony Froshaug—Typography and Texts (London, Hyphen Press, 2000).
- (2) Ibid.
- (3) Ibid.
- (4) Ellen Lupton, Muriel Cooper: Conversation with Ellen Lupton, <http://designwritingresearch.org/essays/cooper.html>, 1994.
- (5) Janet Abrams, Muriel Cooper's Visible Wisdom, www.aiga.org/content.cfm?ContentID=655, 1997.
- (6) Will Holder, from an email conversation, 2005.
- (7) Marc Pawlinger, from an email conversation, 2005.

Better by Adobe.™

Adobe Systems Incorporated
345 Park Avenue, San Jose, CA 95110-2704 USA
www.adobe.com

Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Mac and Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. PowerPC is a registered trademark of IBM Corporation in the United States. Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

© 2005 Adobe Systems Incorporated. All rights reserved.
Printed in the USA 10/05

