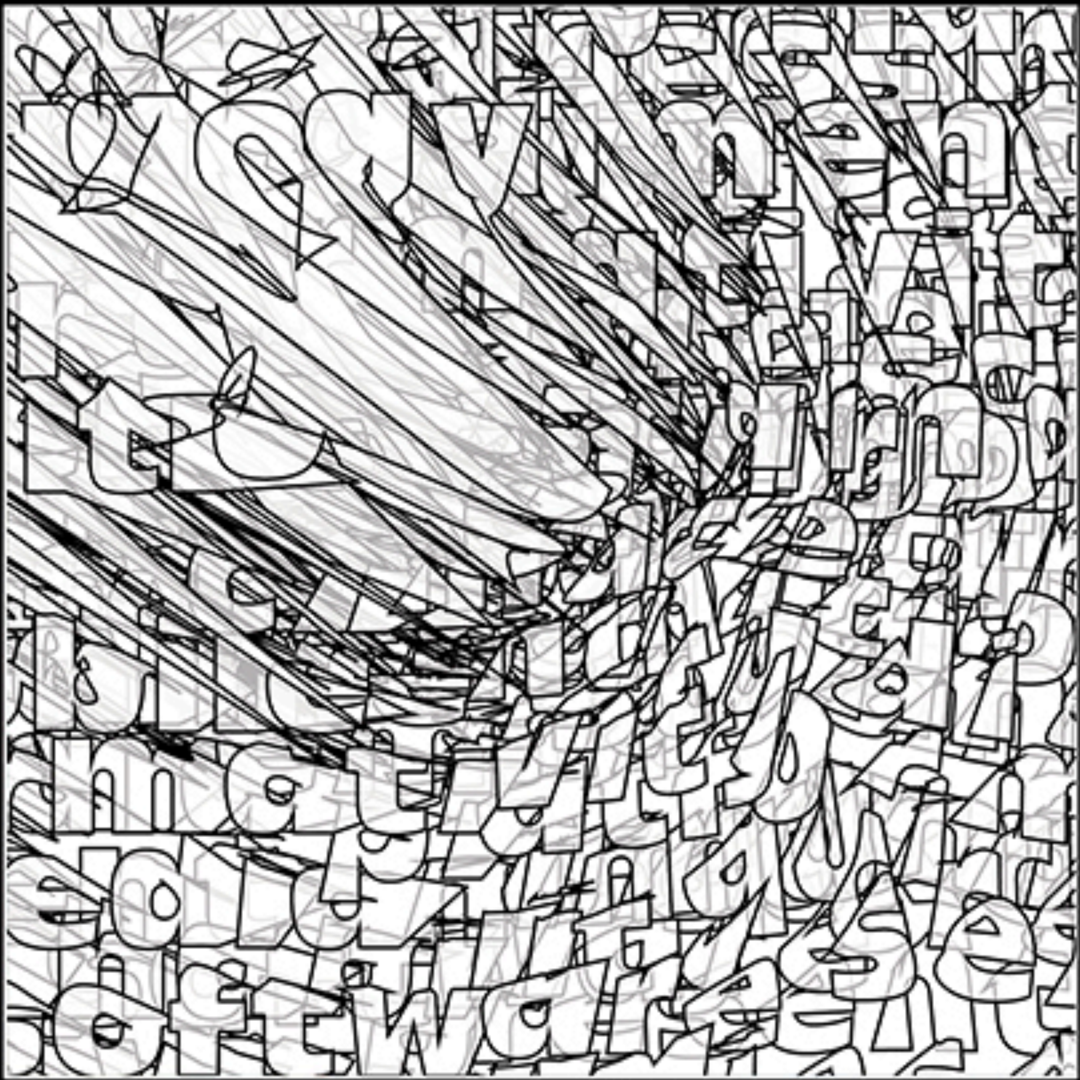
Volume 21 Number 1 January 2010 ISSN 1462-6268

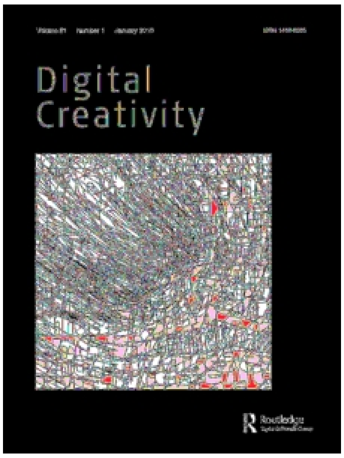
# Digital Creativity

Digital Creativity Volume 21 Number 1 January 2010



Routledge
Taylor & Francis Group

This article was downloaded by: *[Canadian Research Knowledge Network]*
On: *31 October 2010*
Access details: *Access Details: [subscription number 783016864]*
Publisher *Routledge*
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK




## Digital Creativity
Publication details, including instructions for authors and subscription information:
http://prod.informaworld.com/smpp/title~content=t714576173

# Post PostScript please

Jason E. Lewis[a]; Bruno Nadeau[a]
[a] Concordia University, Montréal, Canada

Online publication date: 26 May 2010

**To cite this Article** Lewis, Jason E. and Nadeau, Bruno(2010) 'Post PostScript please', Digital Creativity, 21: 1, 18 — 29
**To link to this Article: DOI:** 10.1080/14626261003654632
**URL:** http://dx.doi.org/10.1080/14626261003654632

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://prod.informaworld.com/terms-and-conditions-of-access.pdf

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

Digital Creativity
2010, Vol. 21, No. 1, pp. 18–29

# Post PostScript please

**Jason E. Lewis and Bruno Nadeau**

Concordia University, Montréal, Canada

jason.lewis@concordia.ca; bruno@obxlabs.net

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010

## Abstract

Writing practices that integrate dynamic and interactive strategies into the making and reading of digital texts are proliferating as more of our reading experiences are mediated through the screen. We argue that rarely do current approaches to creating digital texts operate at the basic textual level of the letterform itself, and that this neglect is partially due to the fact that current font technology is based on print paradigms that make it difficult to work programmatically at the level of individual letters. Work produced in our lab suggests the creative possibilities in being able to easily specify behaviours at such a level, and leads us to propose that writers, typographers and programmers start thinking beyond Postscript-like formats such as OpenType or TrueType to collaboratively develop a new ComplexType format (or formats) that is designed for the twenty-first century as opposed to a simulation of the fifteenth.

Keywords: typography, electronic literature, computer graphics, textuality, ComplexType

## 1 Introduction

As people spend more time reading from screens, authors are spending more effort experimenting with methods for writing in the digital environment. Electronic literature, new media poetry and writing for programmable media are three of the many terms currently in use to describe such texts (Morris 2006). The ongoing proliferation of labels used to describe acts of creative writing that are in some way essentially digital points to the newness of the field and uncertainty about the relationship of digital texts to their print predecessors.

What all such efforts share is an interest in understanding how the qualities unique to the digital environment can be discovered, developed and employed to create engaging, perhaps even innovative, literature (Funkhouser 2007). Thus we have seen writing that, among other strategies, uses hyperlinks to create non-linear reading experiences (Joyce 1990), that combines author-composed texts with texts generated by algorithm (Seaman 1996), that integrates various other media (image, video, audio) to present a multimedia text (Amerika 1993), that incorporates textual movement and transformation in response to the reader's actions (Andrews 2003), and that explores the third dimension to create architectural reading spaces (Shaw 1989).

What very few of these explorations do, however, is experiment with the letterform itself. We argue that this is at least partially a consequence of the material available with which digital writers can work. More precisely, all of us writing in digital media are working within a paradigm for representing letters that was developed for print. The mismatch between tools and currently available

ISSN 1462-6268 © 2010 Taylor & Francis
DOI: 10.1080/14626261003654632
http://www.informaworld.com

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010

techniques makes it difficult to use letterforms in ways that, ultimately, are not that far removed from their print use. A promising way to overcome this difficulty and more fully exploit the rich opportunities that working within a digital environment can offer to authors is to reconceptualise text rendering as a computational process that assumes the screen as its primary display.

We propose the development of ComplexType, a new font format designed and programmed specifically for digital environments. The necessity for this proposal is grounded in a review of recent history of experimentation in type format. We anticipate that implementing ComplexType will open up new avenues of experimentation for the writer of digital texts.

## 2 Moving beyond print

### 2.1 Print bias

The current font formats used by most personal computing systems are TrueType, OpenType and Apple Advanced Typography (AAT), all three of which were developed in the nineties from the PostScript template developed in the eighties. PostScript is a powerful technology (Warnock and Geschke 1992), but it was developed within a print paradigm which prioritised features based on their usefulness in transferring type from the screen to paper with high fidelity.

In the move to digital letterpresses, various techniques had to be developed in order to address the gap between what the rendering engine understands is aesthetically pleasing to a reader and what humans will actually accept (Spiekermann and Ginger 1993). One set of such techniques involved kerning, or the process of adjusting the white space between letterforms. To typographers, creating a font in which all viable letter pairs appear well-spaced is centrally important to whether a text set in the font will not have any unsightly gaps or jambs and, as a whole, will be pleasing to the eye (Figure 1). Presently—almost fifty years after the development of digital fonts—an extraordinary amount of a type designer's time is spent performing this task manually. Modern formats implement various strategies


Figure 1. 'War' in the Kari typeface (top) without kerning and (bottom) with kerning. ©2009 Bruno Nadeau. Reproduced with the permission of the copyright holder.

to assist with this process, and all of these strategies are clearly engineered with a typesetting mentality. To kern a TrueType font, for instance, a table appropriately named 'kern' stores glyph pairs and a kerning value, a positive or negative value indicating the number of units that the second glyph of the pair should be moved by when the pair is typed contiguously. The font creation software employs an algorithm to provide a first-order pass at specifying these tables, and then the type designer will address problem combinations manually.

The kerning table provides a series of flags, one of which indicates the direction, horizontal or vertical, in which rendering engines should adjust the glyph. The binary option is evidence of how the format is optimised for the linearity of typeset text, favouring the grid-like aesthetic of centuries of material printed within the movable type paradigm.

### 2.2 Wrestling with PostScript

In the late eighties, some computer scientists and typographers experimented with PostScript to create dynamic fonts that exploited seldom-used properties of the language to produce differences every time a character was generated. André and Borghi's interpretation of Knuth's Punk font (Knuth 1988, André and Ostromoukhov 1989), Scrabble (André 1990) and Beowolf (Van Blokland and Van Rossum 1990) are the pioneering exemplars of dynamic fonts that harness the

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010


Figure 2. Beowolf. © 1990 Erik Van Blokland and Just Van Rossum. Reproduced with the permission of the copyright holder.

programmatic possibilities of digital media by exploiting features of the PostScript Type 3 format (Figure 2), a version of PostScript Type 1 that Adobe made available to other type foundries. However, the Type 3 format was generally neglected in favour of TrueType, which was considerably better at the task of high-fidelity reproduction but did not incorporate the computationally promising features of the Type 3 format.

The Punk, Scrabble and Beowolf fonts offer an early, tantalising glimpse of what might have been the future of digital typography. Though they focused on randomness, André and Borghi (1989) in particular were already envisioning a more advanced notion of dynamic type aware of and capable of altering context.

Another example that pushed standard typographic technology beyond the assumptions of its engineers was Lucas de Groot's typeface Move Me MM (1994). This font exploited the characteristics of Adobe's PostScript-based Multiple Master (MM) format, which allowed typographers to design the extremes of a typeface between which the system could perform interpolations to generate an infinite number of styles. Move me MM used this feature, which was ordinarily used to quickly produce a wide range of weights (e.g. light, regular, bold), to create glyphs that could be animated by manipulating the sliders in MM-supporting font design software (Figure 3). While this exploit did not allow authors to write with the animated typeface, it did demonstrate what might be possible given the right support. Unfortunately, the MM technology suffered the same fate as the Type 3 format, left behind in favour of the resolutely print-oriented TrueType.

The experiments of André, Van Blokland and Van Rossum's, and De Groot's are three of the few explorations made into using extensions of PostScript as a dynamic language for drawing letterforms. Their work hinted at avenues of development that never fully matured.

Though not focused on the computational infrastructure behind Postscript technology, another research trajectory worth discussing is embodied by the Neville Brody-directed *Fuse* CD-ROM journal. *Fuse* encouraged those interested in type design to push, hard, on the limits of the still relatively new Type 1 format using standard font design tools. Every issue of the journal featured several fonts, each one custom designed for *Fuse* by a leading typographer or designer. Over eighteen issues the project accumulated a diverse array of experiments that envisioned a much greater range of font use than that which finally settled into the status quo. Tobias Frere-Jones's Reactor (1993), in which each successive character slowly fuzzes up earlier characters with visual noise as you continue typing, is a prototypical example of how the *Fuse* work anticipated some of our concerns with the letterform exceeding its standard printerly constraints by altering its context (Figure 4).

A third avenue of exploration was opened up in the late nineties, when Lewis began working with standard PostScript fonts as the basis for SoftTypes (Lewis and Weyers 1999). SoftTypes were behaviours that controlled the dynamic and interactive visualisation of PostScript-based letterforms, and could be used in combination to create arbitrarily complex composite behaviours (Figure 5). SoftTypes were designed to be applied by an author without programming intervention, using menus

Figure 3. Letter 'A' from Move Me MM. © 1994 Lucas de Groot. Reproduced with the permission of the copyright holder.


Figure 4. Reactor. © 1993 Tobias Frere-Jones. Reproduced with the permission of the copyright holder.

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010

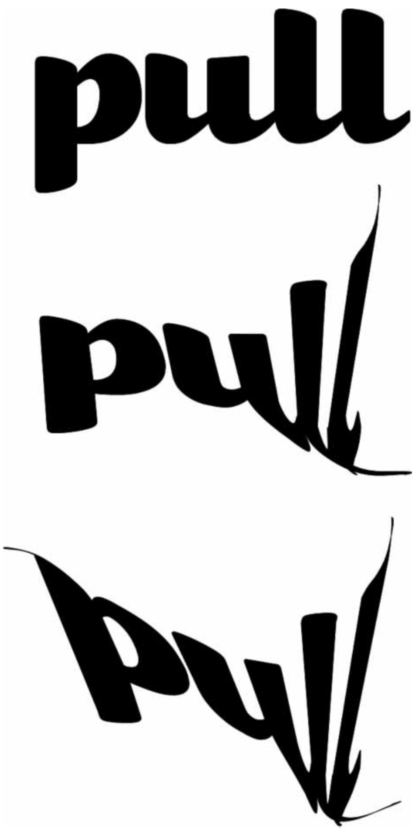



Figure 5. 'Pull' deformed with SoftTypes in It's Alive! © 2001 Jason E. Lewis. Reproduced with the permission of the copyright holder.

in the same way that she might change fonts or apply a bold style. Several years later Lee *et al.* (2002), proposed a kinetic typography engine that took a similar approach, establishing a primitive set of behaviours that could be combined together to form complex behaviours. Both these efforts established interesting software architectures for treating such fonts dynamically, but ultimately, even though both relied on standard Open/TrueType fonts, they proved to be dead-ends because their architectures could not easily be imported into standard word processing or visual effects software.

### 2.3 Post PostScript

At the same time that Lewis was developing SoftTypes, Cho was taking a different approach to addressing the issue of typographic interactivity and dynamics (Cho 1999). His work followed in the footsteps of researchers at the MIT Media Lab—such as Cooper (1994), Maeda *et al.* (1996) and Soo (1997)—who explored kinetic typography starting in the mid-eighties. Cho put aside PostScript and its descendents, abandoning the whole concept of a universal system for digital type. He offered instead multiple custom computational models open for adaptation in static, dynamic and interactive media (Figure 6).

Ten years later, Hillner continued this 'clean room' approach with his experiments in 'virtual typography' (Hillner 2007). Cho and Hillner's low-level, build-it-from-scratch approach towards letterforms added greatly to our understanding of how diverse the notion of a 'font' could be in the digital context while severely highlighting the limitations of standard PostScript-like formats in supporting such diversity.

## 3 From complex surfaces to complex type

### 3.1 Start with the screen

In his paper 'Writing on Complex Surfaces', John Cayley (2005) asks how we could develop an approach to text-based work that is 'faithful to graphics, typography, visuality and textuality all at once'. A central concern of his paper is to grasp the creative and phenomenological consequences of texts that are constructed to have an active, real-time interplay between text, visual aesthetics and structure. He focuses his answer on the increasing richness of reading surfaces and the opportunities their spatial and computational complexity afford for creative exploitation. We are interested in answering the same question by focusing instead on the letterform itself, in particular by moving away from a three-decade-old type technology designed for print and towards a born-digital format. We call such a format ComplexType.

ComplexType will allow a common means of employing letterforms that can exploit several unique characteristics of digital media. Examples include spatial letterforms that function as virtual

Digital Creativity, Vol. 21, No. 1

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010

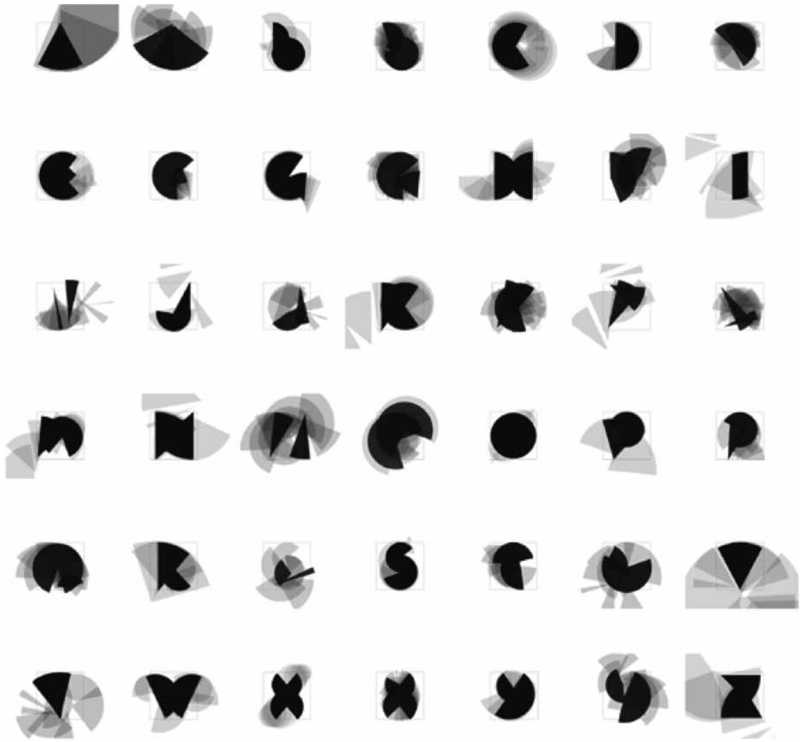


Figure 6. Type me, Type me not. © 1999 Peter Cho. Reproduced with the permission of the copyright holder.

objects with volume and/or which can move in three-dimensional space such as those proposed by Miller (1996); interactive letterforms that can respond to the actions of the reader, other glyphs and other media within the same display environment, such as with Lewis and Weyers' ActiveText (1999); variable letterforms that change, evolve and mutate (Cho 1999); and letterforms designed to take advantage of network connectivity like Twin's use of meteorological data to specify appearance (Littlejohn 2004).

## 3.2 Possible frameworks

The complexity and diversity of typographic spaces offered by computational media make it unlikely that we can devise a universal font format that would allow writing on any complex surface with the same facility and quality. Standard typographic technology like Open/TrueType are universal in that they facilitate the distribution and exact reproduction of type on any devices for which the typographic space is limited to a simulation of print. ComplexType would use a modular approach, taking inspiration from more flexible formats like Unified Font Object (UFO) (Leming *et al.* 2003) and Scalable Vector Graphics (SVG) fonts (Eisenberg 2002), where basic modules necessary for all fonts would be combined with modules that target writing in specific environments.

Typically, a ComplexType font would include a general font information module and a character map, two modules required by all fonts. If the typeface was designed for a 3-D environment, it could include a module of glyph definitions represented as 3-D geometry (e.g. list of triangles). The font could also include a module indicating the dynamic properties affecting the glyphs or their constituents. Furthermore, a separate interactive module could be included to indicate actions triggered by different input mechanisms. An application that supports ComplexType fonts would

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010

check which modules were present in each installed font and make the ones it supports available to the user. It is clear that a certain level of standardisation is required if ComplexType fonts are to be supported in a range of applications that advance beyond the experimental stage, but this process should take place through collaboration with authors and designers to address the various typographic spaces of interest to these groups.

### 3.3 Potential ComplexType fonts

Looking at the history of recent experimentation in this area suggests, even at this early stage, several possible ComplexType font designs.

A notable genre of typeface, which was a significant part of the early experiments conducted by Maeda and later Cho, is type with letters constituted of multiple graphical elements or particles that arrange together to form the glyphs of a typeface. The flexibility inherent in the fragmentation of the glyph provides interesting options to an author. Cho's Type me, Type me not (Figure 6) exploits the adaptability of glyphs made of similar circle sections to create type that smoothly transforms from one letter to another. In the same way that kerning is an activity that targets the flat surface of typeset text, in a ComplexType font it would be possible

Figure 7. Hydre. © 2009 Bruno Nadeau. Reproduced with the permission of the copyright holder.

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010

to define the 'spanning', or the transitioning from one glyph state to another.

Typefaces made of particles also offer interesting possibilities for interactivity. Nadeau's Hydre typeface (2009a) combines a common outline of lines and curves with a large number of small dynamic circular particles to form each letter. The particles, which can appear anywhere in the typographic space, are attracted by certain letters and move continuously once trapped in an outline. The particles are always subject to dislodging forces applied by interacting readers who have a certain control over the formation and breaking apart of the text (Figure 7).

ComplexType fonts would also include features that provide a fine degree of control over the rendering process. Nadeau's Origin typeface (2009b) combines several particles that appear differently over the formation of each glyph. Starting as animated spermatozoa, the particles reach the glyph to which they belong and become specks leaving subtle traces around the letter, slowly revealing it in the negative space (Figure 8).

Another approach towards type design, which was central to the experiments of Miller (1996) and also present in Cho's work, is type built for 3-D virtual environments (Figure 9). Although it is possible to integrate standard outline fonts into the third dimension, the expensive process of converting the glyphs of these fonts to a format supported by 3-D virtual environments adds considerable friction to writing practices that do not assume the immutability of the letterform. In addition, a glyph definition that specifically targets these environments would lead to innovative typeface designs as type designers adapt to the new material from which to build letterforms.

Stroke-based fonts are another approach, one that defines the glyphs of a typeface as a series of strokes, simulating a pen or brush applying ink on paper. This technique dates back to Donald Knuth's pioneering mathematical typography research of the mid-seventies, which led to the implementation of the METAFONT system (Knuth 1999). In addition to METAFONT, stroke-based fonts are commonly used to minimise the number of vertices required to represent the complex ideograms of East Asian languages. A stroke-based ComplexType font would extend Knuth's method in order to incorporate properties specific to dynamic and interactive environments. This, in turn, would facilitate


Figure 8. Origin, based on the Haettenschweiler typeface. © 2009 Bruno Nadeau. Reproduced with the permission of the copyright holder.

Figure 9. Polymorphous font 'f'. J. Abbot Miller.

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010

the manipulation of the letterform over time and by the reader.

A ComplexType specification would involve a core font specification that would allow the use of ComplexType fonts in various types of applications. In the same way that you now specify a typeface (Helvetica, Times, etc.), a style (bold, italic, etc.) and a point size, a ComplexType font would allow you to specify how its glyphs behaved (movement, lifespan, interaction, etc.) The individual glyphs in such a font would integrate capabilities for recognising and using temporal change, handling interaction with the user as well as other letterforms and media elements, processing external data sources and communicating across the network. All these characteristics can be used to determine, in real time, how glyphs appear and evolve over time. The integration of dynamic and interactive behaviours into the ComplexType format would provide writers (and typographers and designers) with a high degree of control over reasonably complex behaviours.

## 4 Writing with complex type

In order for ComplexType to be useful to writers, writers need to be involved in the process of developing the specification. Previous experiments with active glyphs rarely engaged writers centrally, more often being the products of investigations into design and typographic form (cf. Maeda, Cho, LettError) or the computational properties of PostScript (cf. André *et al.*). They have not been investigations into creative language use.

In the NextText project we proposed a tight integration of writing, designing and programming in order to more deeply explore the production of innovative digital texts (Lewis 2007). Among the conclusions we reached three years into the project was that making work which provides a reading experience as rich as its dynamic and interactive experience (and vice versa) requires that the author be able to directly engage temporal change, interactivity, network connectivity, etc. as part of a conscious strategy of meaning making *within* the writing process (as opposed to the post-writing process.) Such *complex writing* will benefit enormously if every aspect of the technology through which it is realised has been rethought in terms of the digital.

The ongoing development of our Mr. Softie application has served as a test-bed to experiment with rudimentary forms of complex writing. In the SoftSketches series (Lewis and Nadeau 2006), we

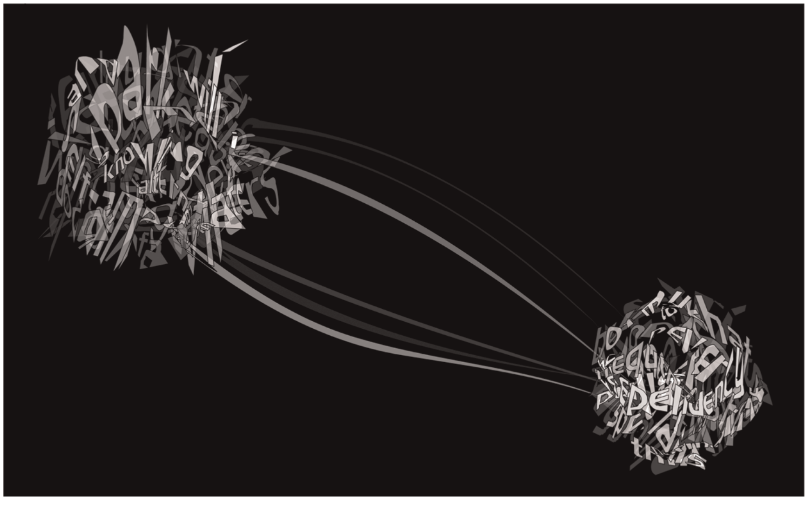


Figure 10. Dependency. Bruno Nadeau.

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010

Figure 11. Dependency (detail). Bruno Nadeau.

wrote poems specifically with Mr. Softie in mind and then used the application to manipulate the appearance and further edit the texts. The resulting works take part in the concrete poetry tradition of intensive focus on the presentation of the text. The text of Dependency, which can be summed up in the line 'an / electric thread ties and bonds the / agglomerated mind', evolved into a mass of letterforms intertwining in two knotted masses grasping at each other across the canvas (Figures 10 and 11).

In *History*, a poem about how the past becomes abstract and flattened the further away we get, we worked the letterforms to the point where they became a collection of abstract geometries layered one on top of another (Figure 12). In both cases, the ease with which Mr. Softie allowed us to apply complex behaviours promoted an ability to work intensely and deeply with the visual representation to the poems, allowing us to shape them however the text demanded.

In 2008, we invited poet David Jhave Johnston to collaborate on the Mr. Softie development process by creating text works with the application and providing us with his critique of the tool. He produced *Softies*, a series of media and motion works for the web that employed Mr. Softie to create the texts (Johnston 2009). One of the series, *Stand Under*, contains the lines 'Develop an understanding/Stand under/Humility understands', and is represented (at first) by a strange, elongated tower of text in which parts of 'understand' are crushing the rest of the poem (Figure 13).

As the piece progresses, the remaining text reasserts itself, pushing up against the 'understand' and causing the whole text to wobble from side-to-side and distort drastically as it expands to take

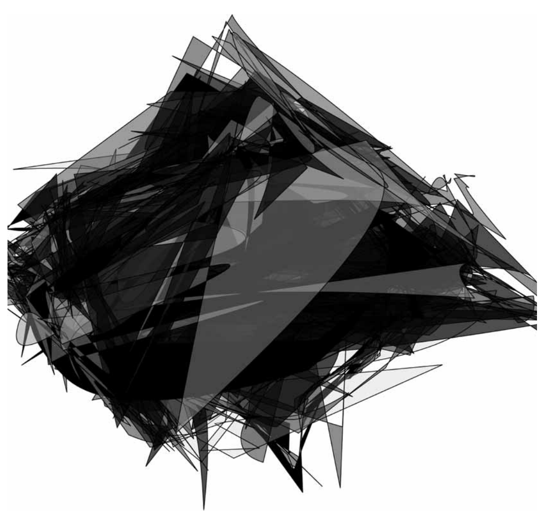
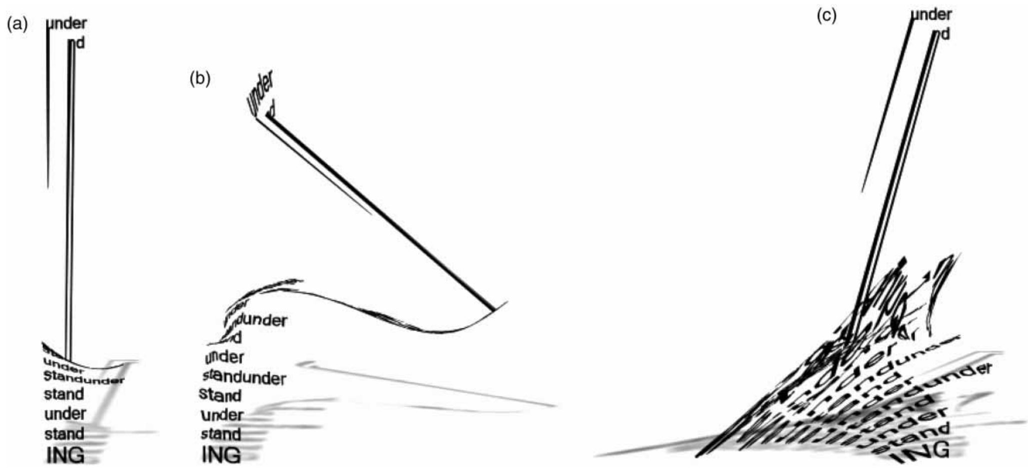


Figure 12. History. Jason Lewis.

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010




Figures 13a, 13b and 13c Stand Under. David Jhave Johnston.

up a normal amount of space. The text, manipulated algorithmically and by hand, possesses a liveliness that would be difficult—and significantly more time-consuming—to achieve using standard tools for word processing or motion typography.

We find it encouraging that Johnston and our works provide initial indication that supporting ComplexType-like capabilities as part of a complex writing process can produce texts that begin to tap into the full power of the computational environment.

## 5 Conclusion

In this paper we have shown how the print bias of current technology impedes our ability to exploit the full potential of digital text. We have discussed various early efforts to get beyond the print paradigm, and have used our current research as the basis for articulating several trajectories of further exploration into ComplexType technology that should prove fruitful in producing tools and techniques useful to writers of any kind of digital texts.

We believe that such efforts will help answer some of the more significant questions involved in the evolution of writing in the digital environment: can digital technology foster new, substantive modes of writing in the same way that the printing press provided a technical foundation for the rise of the novel, the essay and the pamphlet? What techniques can be developed for allowing creators to write a text, design its appearance and program its behaviour in an integrated manner? How can these techniques be configured as tools that allow creators to easily draft, sketch and prototype without losing focus on the content of the work? And, finally, how might access to such tools expand the pool of individuals writing innovative digital texts, and consequently extend the role such texts can play in everyday life?

We are generally interested in considering how qualities unique to media presented via computing machinery can be articulated—both conceptually and technically—in support of new techniques for incorporating meaning into the presentation of texts. The more time people spend reading text on one form of screen or another, the less sense it makes to rely on a technology developed in many core aspects to address challenges related to displaying text on the printed page. Correspondingly, it also makes sense for the writers of those texts to have available to them technology developed specifically for the screen in order to fully exploit the creative possibilities for displaying text on it.

Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010

## Acknowledgements

We thank the Social Sciences and Humanities Research Council, the Fonds de recherche sur la société et la culture and the Hexagram Institute for Research/Creation in Media Arts and Technologies for their support of this research.

## References

Amerika, M., 1993. Grammatron. *Web-based interactive work* [online]. Available from: http://www.grammatron.com [Accessed 10 August 2009].

André, J., 1990. The Scrabble font. *The PostScript Journal*, 3 (1), 53–55.

André, J. and Borghi, B., 1989. Dynamic fonts, André, J., Hersch, R., eds., *Raster imaging and digital typography: proceedings of the international conference*, October 1989 Ecole Polytechnique Fédérale Lausanne. Cambridge, New York: Cambridge University Press, 198.

André, J. and Ostromoukhov, V., 1989. PUNK: de Metafont à PostScript. *Cahiers GUTenberg*, 4, 23–28.

Andrews, J., 2003. Arteroids 2.5. *poemsthatgo*. 14 [online]. Available from: http://www.poemsthatgo.com/gallery/fall2003/arteroids/article.htm [Accessed 14 August 2009].

Cayley, J., 2005. Writing on complex surfaces. *Dichtung-Digital* [online]. Available from: http://www.brown.edu/Research/dichtung-digital/2005/2/Cayley/index.htm [Accessed 10 February 2010].

Cho, P.S., 1999. *Computational models for expressive dimensional typography*, Dissertation. Massachusetts Institute of Technology.

Cooper, M., 1994. Information landscapes, Technical note. Massachusetts Institute of Technology.

De Groot, L., 1994. Move Me MM [typeface]. *Fuse*, [CD-ROM], 11.

Eisenberg, J.D., 2002. *SVG essentials*. Sebastopol, CA: O'Reilly and Associates.

Frere-Jones, T., 1993. Reactor [typeface]. *Fuse*, [CD-ROM], 7.

Funkhauser, C.T., 2007. *Prehistoric digital poetry*. Tuscaloosa, AL: University of Alabama Press.

Hillner, M., 2007. *The poetics of transition: the ambiguous characteristics of virtual typography*. London: Royal College of Art.

Johnston, D., 2009. Glia. *Web gallery of work* [online]. Available from: http://www.glia.ca [Accessed 20 January 2010].

Joyce, M., 1990. *Afternoon, a story*. Cambridge, MA: Eastgate Systems.

Knuth, D.E., 1988. A punk meta-font. *TUGboat*, 9 (2), 152–168.

Knuth, D.E., 1999. *Digital typography*. Stanford, CA: Center for the Study of Language and Information.

Lee, J.C., Forlizzi, J., and Hudson, S.E., 2002. The kinetic typography engine: an extensible system for animating expressive text. *In: Proceedings of the 15th annual ACM symposium on user interface software and technology*, 27–30 October 2002 Paris, France. New York: ACM Press, 31–90.

Leming, T., Van Blokland, E., and Van Rossum, J., 2003. *Unified Font Object (UFO)* [online]. Available from: http://www.unifiedfontobject.org/ [Accessed 20 January 2010].

Lewis, J.E., 2007. Writing-design-programming. *In*: Andrew Hutchinson, ed. *Digital Arts and Culture Conference 2007 proceedings*, 15–18 December 2007 Perth, Australia. Perth: Curtin University of Technology, 204–213.

Lewis, J.E. and Nadeau, B., 2006. *Soft sketches* [online]. Available from: http://www.brunonadeau.com/softsketches/ [Accessed 20 January 2010].

Lewis, J.E. and Weyers, A., 1999. ActiveText: a method for creating dynamic and interactive texts. *In: Proceedings of the 12th annual ACM symposium on user interface software and technology*, 7–10 November 1999 Ashville, USA. New York: ACM Press.

Littlejohn, D., ed., 2004. *MetroLetters: a typeface for the twin cities*, Minneapolis, MN: University of Minnesota Press.

Maeda, J., 1996. *Flying letters*. *Reactive Book Series*, no 2, Tokyo: Digitalogue.

Miller, J.A., 1996. *Dimensional typography*. New York: Princeton Architectural Press.

Morris, A., 2006. New media poetics: as we may think/how to write. *In:* A. Morris and T. Swiss, eds. *New media poetics: contexts, technotexts and theories*. Cambridge, MA: MIT Press, 1–46.

Nadeau, B., 2009a. *Hydre* [online]. Available from: http://www.brunonadeau.com/complextype/hydre/ [Accessed 20 January 2010].

Nadeau, B., 2009b. *Origin* [online]. Available from: http://www.brunonadeau.com/complextype/origin/ [Accessed 20 January 2010].

Seaman, B., 1996. *Emergent meaning as examined and explored within a specific generative virtual environment*, Dissertation. Centre for Advanced Inquiry in the Interactive Arts.

Shaw, J., 1989. *The legible city* [online]. Available from: http://www.jeffrey-shaw.net/html_main/show_work.php3?record_id=83 [Accessed 20 January 2010].

Soo, D., 1997. *Implementation of temporal typography system*, Dissertation. Massachusetts Institute of Technology.

Spiekermann, E. and Ginger, E.M., 1993. *Stop stealing sheep*. Berkeley, CA: Adobe Press.

Van Blokland, E. and Van Rossum, J., 1990. Random code—the Beowolf random font. *The PostScript Journal*, 3 (1), 8–11.

Warnock, J. and Geschke, C., 1992. *PostScript language reference manual*. Menlo Park, CA: Adobe Systems.

**Jason E. Lewis** is an associate professor of Computation Arts at Concordia University. His research/creation practice revolves around experiments in visual language, text and typography, with a core interest in how the deep structure of digital media can be used to create innovative forms of expression. He founded Obx Laboratory for Experimental Media (http://www.obxlabs.net), where he directs projects devising new means of creating and reading digital texts, developing systems for creative use of mobile technology, designing alternative interfaces for live performance, and using game and virtual environments to assist aboriginal communities in preserving, interpreting and communicating cultural histories.

**Bruno Nadeau** is a Research Associate at the Obx Laboratory for Experimental Media at Concordia University. He is a computation artist with a special interest in technologies to thinker with, unusual interactive interfaces and typography. His interactive artwork was shown in new media art galleries like Oboro in Montréal and the Beall Center for Arts and Technology in Irvine. His publications were presented as part of international conferences such as Digital Arts and Culture (DAC) and Tangible and Embedded Interactions (TEI). Bruno received a MS in Information and Computer Sciences with a concentration in Arts, Computation and Engineering (ACE) from the University of California, Irvine, and holds a BSc with a major in Computations Arts from Concordia University in Montréal.



Downloaded By: [Canadian Research Knowledge Network] At: 14:22 31 October 2010