Dynamic Poetry

Introductory Remarks to a Digital Medium

by

Jason E. Lewis

B.A. German Studies, Stanford University 1990
B.S. Symbolic Systems, Stanford University 1991

Submitted to the School of Design for Industry in partial fulfillment for the Requirements of the Degree of

Master of Philosophy in Design

at the
Royal College of Art

December 13, 1996

Dynamic Poetry

Introductory Remarks to a Digital Medium

by

Jason E. Lewis

Submitted to the School of Design for Industry in Partial Fulfillment for the Requirements of the Degree Master of Philosophy in Design at the Royal College of Art.

Abstract

This thesis investigates the processes by which media evolve in order to suggest future directions for the digital medium. It develops the notion of *content-lag* to describe the time-span between the introduction of a medium and the point at which it is used to produce artifacts which exploit the affordances particular to that medium to their fullest. Current difficulties in developing strong content in the digital medium are discussed in terms of content-lag. The thesis then argues that a more considered approach to interactivity will assist in decreasing content-lag in the digital medium. A framework is proposed for rediscovering the ways in which *interactivity* is deployed in the digital medium. The arguments of the thesis are embodied in project work which explores the possibilities of a computer-based poetic genre. This project work exists as a collection *dynamic poems*, which are available for interaction on a companion CD-ROM.

Thesis Supervisor: Gillian Crampton Smith

The work reported herein was supported in part by Interval Research Corporation.

Acknowledgments

I am fortunate in that many colleagues and friends have assisted me both in my studies while at the Royal College of Art and in the preparation of this thesis. The following people deserve my gratitude and more.

The staff of the Computer Related Design department:

Gillian Crampton Smith for believing that my lack of formal design training was not a bug, but a feature.

Bill Gaver for pithy commentary on all things relating to people and computers.

Colin Burns for being one of the most brilliant teachers I have ever had.

Michael Fields for teaching me, with infinite patience, that $V = IR$ and from that simple equation all good things flow.

Durrel Bishop, Jane Roderick and Geoff Smith for offering up snippets of code and coding advice which saved the day many a time.

Gail Neary and Brigitte Lievre for patience, kindness and perspective, even when faced with the lack thereof.

The members of Interval Research Corporation:

David Liddle for believing that art and design have an important role to play in the structuring of emerging digital technologies, and for believing that I had something important to contribute to this arena.

Kären Weickert for opening the door and Bonnie Johnson for bringing me into IRC, collecting together a great group of people with whom to work and encouraging me to push my limits.

Brenda Laurel for refusing to believe that this RCA thang was not going to happen, and lobbying to make sure it did.

Noel Hirst for trusting that what I say I need, I need, and making sure it gets to me whether I am 3,000 miles away on a rock-n-roll tour or 8,000 miles away on an art-n-design tour.

Bud Lassiter for performing his video voodoo at the very beginning and very end of this whole affair.

Interval Research Corporation as a whole, for sponsoring this work and for providing unbounded opportunities for realizing my ideas.

The friends and family who have suffered:

Sanny Lustig, Daniel Potter, Angela Quail and Brent Williams for not running away when this beast landed on your desk and instead taking the time to read it carefully. Ms. Quail, you deserve special thanks for phoning edits in from up in the mountains and catching all those misplaced apostrophes.

Leroy Bradford Brown, Jr. for teaching me , a long, long time ago, that the life of the mind was mine for the taking.

Elaine Brechin for listening to two years of blather about poetry, media and interactivity, and still providing considered criticism and unfailing support. There is a poem about you to be written as soon as this thesis is accepted.

My family for supporting me even when I do not make myself clear, and for *enthusiastically* supporting me when you do.

Laurel Lewis for teaching me all the basic truths about life. It's been a long haul, but, for a couple of hicks, we have managed to do alright.

TABLE OF CONTENTS

Abstract ii

Acknowledgments iii

Table of Contents v

Illustrations vii

Chapter 1 **Introduction** 8
- 1.1 The Problem 8
- 1.2 Motivation 9
- 1.3 Related Work 10
- 1.4 A Comment About Scope 10
- 1.5 The Structure of this Thesis 11
- 1.6 About the CD-ROM and the Software 11

Chapter 2 **New Media** 13
- 2.1 Paradigm Shifts 13
  - 2.1.1 Manuscript to Print 14
  - 2.1.2 The Evolution of Film 15
- 2.2 Shaping a New Medium 17
  - 2.3.1 Nativity and Multiplicity 18
  - 2.3.2 Computer Art/Digital Art 20
  - 2.3.3 Well-trodden Paths – Hypertext and Usability 20

Chapter 3 **The Word** 22
- 3.1 Poetry 22
- 3.2 Concrete Poetry 25
- 3.3 Typography 28

Chapter 4 **Interactivity** 30
- 4.1 The Need for Definition 30
- 4.2 Dynamics 32
  - 4.2.1 Constructive 32
  - 4.2.2 Reactive 33
  - 4.2.3 Active 33
  - 4.2.4 Static 33
- 4.3 Response 34
  - 4.3.1 Dependent 34
  - 4.3.3 Independent 34
  - 4.3.2 Hybrid 34
  - 4.3.4 Non-responsive 35
- 4.4 Time 35
  - 4.4.1 Cycle-time 35

4.4.2 Real-time 35
4.4.3 Interactive-time 36

Chapter 5 **Experiments** 37
5.1 Conversions 37
5.1.1 *Flash I & II* 38
5.1.2 *Scratch* 39
5.2 Concrete Poetry Redone 41
5.2.1 *WordNozzle* 41
5.2.2 *WordNozzle* – desktop version 42
5.2.3 *WordNozzle* – installation version 44
5.3 A Digital Poetry 46
5.3.1 *Breeder* 48
5.3.2 *Dying Lying Rotting* 52
5.3.3 *Telecommunication* 54
5.3.4 *Cross Purposes* 55
5.4 Beyond the Word 56
5.4.1 *Aura* 57
5.4.2 *Life is Bait* 58
5.5 Minor Reflections 60
5.5.1 New Poetic Forms 61
5.5.2 Punctuation 61

Chapter 6 **Conclusion** 63
6.1 Conclusion 63
6.2 Future Directions 64

Appendix A **Technical Discussion** 66
Aura 66
WordNozzle 67
Life is Bait 71
Breeder 76
Telecommunication 79

Appendix B **Illustrations** 82

Bibliography 120

Illustrations

Number Title Page

1	F.T. Marinetti, *Aprés a Marne, Joffre visita le front en auto*	83
2	*Club Dada* prospectus	84
3	T. Tzara, *Une Nuit d'Eches Gras*	85
4	Guillaume Apollinaire, *Lettre-Océan*	86
5	Augusto de Campo, *Here Are The Lovers*	87
6	C. Fernbach-Flarsheim, *Mirror Field Inside Random Field*	87
7	Katherine McCoy and Michael McCoy, *The New Discourse*	88
8	David Carson-edited issue of *Ray-Gun*	89
9	Phil Baines, *Can You...?*	90
10	Erik van Blokland, *Nimida*	90
11	Jonathon Barnbrook, *Burroughs* typeface	91
12	Visual Language Workshop,*Information Landscape*	92
13	Yin Yin Wong, *Little Red Riding Hood*	92
14	Jonathon Steuer, *Interactivity Matrix*	93
15	John Maeda, *Flying Letters 1*	94
16	John Maeda, *Flying Letters 8*	94
17	Amiri Baraka, *Wailers,* Poetry in Motion	95
18	*Flash I,* Initial State	96
19	*Flash I*, Cycling	96
20	*Flash II*, Initial State	97
21	*Flash II*, Progressive re-construction	98
22	Tom Berrigan, *Whitman in Black*, Poetry in Motion	99
23	*Scratch*, Initial State	100
24	*Scratch*, Composed State	101
25	*WordNozzle*, Desktop Version	102
26	*WordNozzle* Installation, Nozzle and Screen	103
27	*WordNozzle* Installation, Close-up Nozzle	103
28	*WordNozzle* Installation, User, Nozzle and Screen	104
29	*Breeder*, Sequence	105
30	*Dying Lying Rotting*, Sequence	106
31	*Dying Lying Rotting*, Zoom-in	107
32	*Telecommunications*, Sequence	108
33	*Cross Purposes*, Sequence	109
34	*Aura*, Detail	110
35	*Aura*, in Self-Storage Exhibition	111
36	*Life is Bait*	114
37	*WordNozzle* Installation, Electronics Overview	115
38	*WordNozzle* Installation, Circuit Diagram	116
39	*WordNozzle* Installation, PIC Chip Flow Diagram	117
40	*WordNozzle*,Installation, IR emitter and Potentiometer	118
41	*WordNozzle* Installation, Computational Overview	119

Chapter 1 Introduction

Poetry, in a sense, is the noise of science.
– Michel Serres (Lechte 1994)

1.1 **The Problem**Digital media force us to look at traditional media in a new light, both in terms of how works of art and design are produced and how users receive those productions. Digital media's ability to subsume the functionality of many other media means that artists and designers have an extraordinarily powerful tool with which to work; at the same time, current focus on functionality has retarded the development of both a mature aesthetic and a conceptual framework specifically suited to this new form of communication.

In the initial development of any new medium designers rely for a time on the paradigms of previous media. The time between the technical development of a medium and the development of an aesthetic native to that medium I have chosen to call the *content-lag*. Just as it took several decades for film to fully separate itself from theater and photography, and much later, for video to separate itself from film, the computer-based medium will take some time to move beyond obsession with functionality, overcome content-lag and develop a character all of its own.

I undertook the Dynamic Poetry project to explore the consequences of developing a computer-centric aesthetic while simultaneously exploring functional capabilities. Composed equally of theoretical and historical investigation and practical experimentation, the Dynamic Poetry project has investigated ways of re-designing the inscribed word for a computer-based environment. As the many attempts to make a useful electronic book have shown, simply transposing words from the printed page to the bit-mapped screen does not create an expanded reading experience. Instead, these attempts accentuate the failings of the machine and fail to leverage its strengths. Furthermore, when text appears alongside sound, video and animation, it becomes very evident that the behavioral and temporal possibilities of text have not been well explored. In well-designed computer-based work, one can see how most of the major components establish presence through movement and change. Yet, hyperlinking and deconstructive fonts aside, the text in digital media remains as inert and commonplace as it has in 450 years of printing. Part of the maturation process for the digital medium will require that text move beyond what we expect of it from its life in the printed environment. Those

who work with text in the digital environment will need to developed a more nuanced understanding of *interactivity*, particularly in the confluence of program dynamics, user responsiveness and time control.

I have chosen poetry as the textual application for this study because of the way it, in its disruption of commonplace speaking and reading patterns, provides a model for how far the structure of language can be stretched while remaining intelligible, functional and enjoyable. In the moment the reader is made aware of the difference in structure, he must also be led to not only accept that difference, but to also incorporate the significance of that difference into the overall meaning of the poem. The space between disrupting the normal communicative methods of the language and destroying that communication is a delicate one. Through the Dynamic Poetry project I have sought to develop interactions, representations and content which can inhabit that delicate space, as well as argue for various ways in which designers can inscribe and users can receive text as it changes to accommodate its latest home.

Finally, as the title suggests, my purpose in this thesis is not to define *the* digital medium, but to lay some of the groundwork for *a* digital medium. Someday there will be a vast range of digitally-enabled communication and expression; this paper does not presume to propose a basis for the entire spectrum.

1.2 **Motivation**Within the scope of this thesis project, I have sought to present several new ways of perceiving and interacting with text within the digital space. By grounding my exploration in a historical timeline that pays attention to previous changes in communication technology, I hope to illuminate both the blindness and the true feats of transcendence that accompany such change. In this way, I hope to minimize the former and maximize the latter in my own experiments to transform our use of text.

The dynamic poems themselves exist simultaneously as experiments in the state of the art and experiments in poetic expression. I hope that my commentary on both aspects will not only contribute to the discussion of what is happening to traditional literary forms as they are metamorphisized into the digital realm, but also expand the acceptable and familiar range of interactive expression.

As somebody who has been interested in and writing poetry for a decade now, I have a desire to see a halt to the digital repurposing of existing texts in favor of a writing that is explicitly, and in some essential sense, exclusively dynamic and/or interactive. Even though poetry remains as powerful a mode of expression as ever, its modern audience is minuscule in comparison to that of other forms of communication. If we do not find ways to adapt poetry to the new digital environment, I fear it will become even

more isolated. The poems I produced as part of this project suggest how such an adaptation may take place.

1.3 **Related Work**This thesis draws from a wide variety of research topics. Typography and digital media, the history of the book and the cinema, and the theory and practice of poetry all play a role in the following discussion.

The Visible Language Workshop of The Media Lab of the Massachusetts Institute of Technology developed digital typography which gave me interesting examples of how to leverage the computer's strengths to heighten type's ability to communicate emotions and ideas. Of particular use to this thesis was the work of Suguru Ishizaki, Yin Yin Wong and David Small.

John Maeda's manifesto on "metadesigning" and his subsequent efforts to create digitally authentic design-forms appeared mid-way through this project, providing both inspiration and confirmation of the concepts I explore below.

The *Fuse* series on font design represent one of the few examples of on-going experimentation in digital letter-form design. *Fuse* supports the development and publication of many fonts that either take their inspiration from the digital environment or which possess the interactive and dynamic qualities which I explored.

Various work on hypertext, particularly that of J. David Bolter and George P. Landow, offer a deep analysis of the culture of writing and how digital technology is effecting that culture. They, in turn, owe a great deal to historians such as Walther J. Ong and Elizabeth Eisenstein for the background out of which they – and I – extrapolate future literary trends. Related to this corpus are the writings of Gregory Ulmer, who does a splendid job of proposing a full theory for a new, video-based interactive media.

Just as this thesis was being completed the work of William Seaman was brought to my attention. His work explores the relationship between language and image within a digital environment, and has produced such interesting efforts as navigable poems and automatic poem-generators.

The work that more than any other triggered this project is *Poetry in Motion, vol. I* from the Voyager Press. This CD-ROM represents both the promise and the pitfalls of digital media in general, and of the next stage in poetry in particular.

1.4 **A Comment About Scope**

For the most part, this thesis does not discuss two genres within the digital medium which have been the sites of vigorous creative activity: games and virtual reality. In the case of games, a responsible treatment of the genre's goal-driven, action-oriented nature would

require a thesis of its own and would have necessarily meant a less thorough treatment of the subject at hand. Though examining virtual reality would have introduced an interesting dimension to the discussion of cinematic pursuits of realism, it would have also required a phenomenological and ontological investigation of immersive environments which, while part of the larger future of the digital medium, is not essential to my discussion of non-immersive poetic creations. I believe that the historical approach I present here would benefit the creators of games and virtual realities, and I hope that others will find the framework I introduce to be of use in examining those genres.

1.5 **The Structure of this Thesis**

The next chapter introduces content-lag as term useful for understanding some of the processes by which a medium reaches maturity. This introduction draws on the evolution of the letterpress and the cinema as historical grounding, and then connects this history to the present state of the digital medium. Chapter 3 discusses the reasons and inspirations for employing poetry as the vehicle with which to drive my experiments in overcoming content-lag and developing a framework for understanding interactive design. Chapter 4 dissects that framework and provides examples from the Dynamic Poetry experiments to illustrate it. In Chapter 5, I discuss each of the experiments in depth. For each piece, I describe its appearance and how the user is meant to interact with it[1] and the effect of the piece. The final part of Chapter 5 contains a discussion of two pieces which are not dynamic poems but which were created during the same time frame and embody some of the "native" media arguments I offer in Chapter 2. Finally, Chapter 6 concludes with a review of the arguments presented and suggestions for further work. This last chapter is followed by two appendices, the first of which contains illustrations and the second of which contains a discussion of technical issues which arose in the course of the experimentation. After the appendices is the bibliography.

1.6 **About the CD-ROM and the Software**

Dynamic Poetry, the CD-ROM which accompanies this written text, should be considered an essential component of the thesis. All of the discussed experiments can be accessed on it. The reader can either use the *Dynamic Poetry Finder* to navigate between

---

[1] I have chosen to employ the word 'user' to describe anybody who interacts with the software in question. In some case, the user may be interacting as a reader, in some places as a writer, in others as both, and in yet other cases simply as someone who is using a piece of software. Most times the intended meaning of the word can be disambiguated by context; where it cannot be I make special note.

the experiments, or he can access them directly. Both the self-launching files and the associated source-code can be accessed.

In Chapter 5, under the heading for each experiment, I have put a pointer to where that file exists on the CD-ROM. The pointer is in the format of [drive:folder:file].

I created all of the experiments with Macromedia's Director. Director is an application for interactive design which utilizes a scripting language called Lingo. In some cases, such as *Aura* and the installation version of *WordNozzle*, I augmented this software with custom-made mechanical-electronic subsystems. In other cases, such as *Life is Bait* and *WordNozzle,* I employed extensions to Director written by others in the mid-level language C and called X-Commands (XCMDs) and XObjects (XObjs). All of the work is designed to run on a Macintosh computer (the faster the machine, the better the result) in 16-bit color or better.

Chapter 2 New Media

Some people think to make a color photograph, you just have to put color film in the camera. The result is not a color photograph.
– Harold Allen (Smith 1992)

2.1 **Paradigm Shifts**

The introduction of any significant new mass medium is often accompanied by both wildly dire predictions of how the new medium will destroy literate culture and wildly optimistic predictions about how it will supersede existing mediums in expressive capability. Several thousand years ago, Plato decried one of the earliest communication technologies:

> [Writing] will produce forgetfulness in the minds of those who learn to use it, because they will not practice their memory. Their trust in writing, produced by external characters which are no part of themselves, will discourage the use of their own memory within them. You have invented not an elixir of memory, but of reminding; and you offer your pupils the appearance of wisdom, not true wisdom. (Bolter 1991)[2]

While a much more recent voice, describing the arrival of video technology in the Manhattan arts community in the early 1960's, triumphantly declared "[a]s collage technique replaced oil paint, the cathode ray tube will replace the canvas." (Danto 1995)

These extremes of prognostication very rarely come true. What is true is that a new medium filters slowly through a culture, augmenting existing media rather than replacing them and evolving the communicative gestalt rather than revolutionizing it (Ong 1977.) The speed of this filtering is retarded or accelerated by a myriad of different factors, many of them – economic, political and very simple human factors – which are independent of the functional or evocative qualities of the medium itself, and of those working within the medium.

However, some factors affecting the artistic acceptance and maturation of a new medium reside more fully within the control of those advancing the medium. The willingness of artists, in concert with technologists, to look for affordances native to that

---

[2] Citing *Phaedrus* 275A.

medium, the degree to which they are willing to devote energy to basic experimentation, and the context they create for their audience to be able to receive such experimentation all fall within this category. Taken together, these factors all contribute to content-lag. Content-lag is the time it takes to develop content which is uniquely and powerfully suited to a new medium. Closely related to content-lag is *medium stability*, or the rate of change within the technological structure of the medium. Both phenomena influence each other.

In the following discussion of the evolution of two mediums, print and film, I will strive to illuminate the basis and usefulness of these terms for considering the current state of the art.

2.1.1 **Manuscript to Print**

For almost a hundred years after the invention of the printing press in Europe in 1450, the form of the book remained similar to that of the manuscript. (Febvre 1976) Though the technology had changed the possibilities, printers took that long to change their perception of the nature of a book, to change from seeing it as a simple mechanistically produced off-shoot to understanding it as the unique form that it is.

The dependence on the affordances particular to the manuscript effected everything from the presentation and organization of the book to the appearance and lay-out of the type. As manuscripts were created by hand, scribes resorted to abbreviations of common words and terms in order to decrease the amount of writing they had to do; this time-saving device carried-over into the print world until it sunk in that the printing press did not get tired. Vellum, the writing surface used in manuscripts at that time, was expensive, leading scribes to employ a compact, dense text-form composed out of small scripts, a minimal amount of space between lines and no space between words; these habits also continued after the advent of far less precious paper writing surface used in the movable-type printing process. It took a century for European literary culture to adapt design fully to the capabilities of the new technology. (ibid..)

It took even longer for content to reflect the change. Landow notes that it required "several hundred years of gradual change and accommodation, during which different reading practices, modes of publication, and conceptions of literature obtained" before Europe had successfully made the transition to a print-based culture. (Landow 1992) This span of time, between 1450 and approximately 1700, represents the content-lag of the move from manuscript to print. The content-lag of a communications technology is the time which it takes for the host culture(s) to adapt itself to that technology to the point of

developing new literature particular to it and according it a pervasive place in the social structure.

One might wonder why it takes any amount of time at all, or why it takes decades at any rate. As Ong and Bolter have persuasively argued, a change, or more accurately, an extension, in communication technologies both signals and predicates changes in both the individual and society which transcend instrumentality. The transition from an oral to a lettered, or chirographic society, writes Ong, "restructured consciousness, affecting men's and women's presence to the world and to themselves and creating new interior distances within the psyche." (Ong 1977) Human consciousness on the cusp of the printing-press extension of chirographic culture must accommodate different sources and quantities of information; it must tune its ear to different literature; and it must reconfigure its understanding of itself and its expressive and reflective capabilities. Very little of this, of course, is a process conscious to or controllable by the individual.

Over almost fifteen generations, however, this subtle process produced significant results. One of them was the rise of several new, major literary forms, including the pamphlet, the novel and the essay. These forms matured over several hundred years of trial-and-error during which the core technology they were based upon changed very little. Though the printing presses got larger, faster and more elaborate, the fundamental process remained essentially unchanged.

2.1.2 **The Evolution of Film**

A brief look at the development of film as a mass communications technology provides an instructive contrast to that of printing. Two particular items stand out. One is the relatively shorter period of time it took for film to become a mass medium. Two is the instability of the technology as this process was occurring, at least in contrast to the stability enjoyed by printing technology.

Dating the "beginning" of film is a difficult task, more difficult than that of printing because of the lack of a scholarly collusion similar to the one that has granted Gutenberg the honor even though little direct evidence exists to support his claim. For the purposes of this discussion, I will use 1895 as the beginning date, the year that the Lumière brothers perfected projection. At earlier dates we find plenty of Magic Lanterns and kinetiscopes, but they all lacked a quality which we have since come to identify as an essential element of film, namely that it is an audience experience, not an individual experience like those provided by these other technologies. (Sklar 1993)

Dating the other end, the point at which (like printing in the 1700's with the advent of the novel) the form came into maturation is difficult, and therein lies its interest for us and its relevance to the definition of the term content-lag.

In technological terms, film went through three distinct phases. First, there was the silent era which lasted from the late 1890's to 1920. Then, with the wide-spread application of sound beginning in the early 1930's, "talkies" became dominant. Finally, sometime around the early sixties, films shot in color began to outnumber those shot in black-and-white, leading to the present situation in which most mainstream films are expected to be in color and include sound. (ibid..) This seemingly logical progression towards greater realism was in fact hotly resisted by not only film theorists, but by actors and directors as well. As Arnheim wrote in 1935:

> The introduction of sound film must be considered as the imposition of a technical novelty that did not lie on the path the best film artists were pursuing. They were engaged in working out an explicit and pure style of silent film, using its restrictions to transform the peep show into an art. (Arnheim 1957)

It is debatable whether Arnheim's clear preference for the silent film reflects a true understanding of the essential nature of the form; what remains interesting about this comment and others like it is the very conscious realization that the destiny of the technology may not necessarily be the most desirable destiny of the art and design forms based on that technology.

Robert Sklar (Sklar 1993) echoes Arnheim's diagnosis:

> There were split-screen images, double exposures, visualizations of inner thoughts, a blend of symbolism and realism, a stress on the performance of character psychology, and, above all, careful attention to the aesthetics of image through visual contrasts, shadings, and the play of light. Many of these elements, and particularly the visual, gave way before the technological demands of early sound apparatus.

The noise-control requirements of early sound-equipped cameras halted the increasing mobility and fluidity of the camera, drained time and attention away from experiments in lighting and screen proportion. The inventiveness of the moment is exemplified in *Napoleon*, a film made in 1927. As Sklar describes it:

> Gance [the director] devised a plan utilizing three standard screens side-by-side...The complete film ran over five hours. though much of it

> appeared on the single center screen, at crucial sequences the curtains would draw back to reveal the other two screens. The screens were used in variable combinations – sometimes three separate images, sometimes a central image flanked by two parts of another scene, in climactic moments action (photographed by three separate cameras) sweeping across all three screens in a single image. (Sklar 1993)

This flexibility allowed the director to frame everything from intimate, personal moments to the vast scenery of the battlefield in an appropriate way, giving each sequence the appropriate amount of space. Yet sound re-focused such energy, side-tracking the visual maturation of the medium and drastically curtailing the diversity of the medium.

In opening up a new aesthetic dimension at the same time the medium was reaching a certain level of sophistication – evidenced by the work being done by D.W. Griffith, Sergei Eisenstein, Fritz Lang and others in the late teens and early twenties – the content-lag which had been overcome by these pioneers reasserted itself. With having to accustom themselves to the use of sound, and still struggling with the use of color, film-makers did not regain such confident artistic footing until the early forties.

From comparing these two histories, a major factor in the magnitude of the content-lag can be discerned, namely how quickly the new medium reaches structural stability. If the medium is in a constant state of flux, neither the individual artist nor his audience have enough time to absorb one wave of changes before a new one comes along.

We can also see in this comparison that content-lag is not a prescriptive term, but a descriptive one. To speak of whether it was a good or bad thing that it took several hundred years for the novel to appear after the introduction of its enabling technology is neither meaningful nor useful. It is a historical fact. The same can be said of the destabilizing effect the introduction of sound had on film. Neither should the concept be seen as advocating that the technological evolution underpinning the medium be halted in order for the art to catch up and make use of all the functionality available at any given slice in time. What the term does do is provide us with a term and a historical perspective for capturing what is presently happening with digital media.

2.2 **Shaping a New Medium**

We pitched our tents and dragged into camp our experiences in varied fields. Private activities, accidental past professions, unguessed crafts, unsuspected eruditions–all were

pooled and went into the building of something that had, as yet, no written traditions, no exact stylistic requirements, nor even formulated demands.

– Sergei Eisenstein (1957)

We find ourselves, almost five centuries since the invention of printing, and almost a hundred years since the beginnings of film, developing a new medium which may have just as large an impact as either one of those mediums. The time-period, as well as they dynamic, is similar to the one described above by Eisenstein, surveying the embryonic state of Soviet film-making in 1933. The number of people able to experiment with the medium, and thus quickly explore its possibilities, is greater than the number of those who could afford to employ early printing presses or who had early access to film-making apparatus. Yet instead of the technological change which effected printing on a century-by-century basis, and film-making on a decade-by-decade basis, we find ourselves immersed in a technology whose functional range changes almost yearly. Printing and film exhibited enough medium stability as to allow long-term, concerted experimentation in developing content *native* to those media ; digital media make such prolonged effort difficult to maintain, as the ground is constantly shifting. The content-lag is exacerbated, and an inverse relationship between the feats of technological sophistication and the artistry has arisen. Instead of change, as is often the case, being a catalyst of artistic inspiration, it seems to inhibit it.

Retarding technological acceleration is neither possible, nor particularly desirable. If we are to find leverage for rectifying the inversion, it must be sought elsewhere. As Eisenstein's "written traditions" and "stylistic demands" would at this point be premature and eventually stultifying, I will instead turn to "formulated demands." I will consider two different manifestos, one by John Maeda militating for a uniquely digital medium, and one from Rudolf Arnheim, decrying the loss of variety within the film medium, before adding my own voice.

2.3.1 **Nativity and Multiplicity**

John Maeda calls for digital media to explore the qualities native to it instead of reflexively relying on the habits and customs ordained by predecessor media. He writes:

> Yet, ironically, in the midst of all this graphical radicalism and technology, design is progressing not forward, but backward. What most seem to label as radical is simply a jumble of styles and elements recycled from the past, while the latest software upgrades merely drive this trend towards mindless copy and paste. The "digital" of digital graphic design

> refers to a mechanical process, not an original philosophy of communicative design, as it rightfully should. This philosophy will fail to materialize while designers, despite being equipped with technology befitting a rocket scientist, continue to rely on the mystical traditions of pigment on paper. (Maeda, 1995:b)

Though his rhetoric, like Arnheim's, may slightly obscure it, his is a call for designers to re-evaluate their starting position when approaching the digital medium. Maeda does not call for rejecting anything connected to existing mediums of print, film or image; he goes to great pains to acknowledge his respect for those fields and the wonders they can work. He is exhorting designers to acquire – or formulate – a grasp of the peculiarities of *digital* media. Many techniques honed through hundreds of years of practice will be useful within this new environment, but just as many will not translate well. Furthermore, by attempting to approach digital media with fresh eyes, those innovative techniques which will make the media truly sing will be discovered more quickly.

Turning to Arnheim, we see that he expounds on a similar theme when talking about the development of film, writing "[i]n order that the film artist may create a work of art it is important that he consciously stress the peculiarities of his medium." (Arnheim, 1957) His opposition to the imposition of the sound apparatus, discussed above, arises from this critical commitment to exploring the technology in artistic terms, not because of any intrinsic inappropriateness of sound.

This commitment leads Arnheim to be capable of simultaneously condemning the sound film for the way it took over film production and praise it for its own particular strengths. "By sheer good luck," he acknowledges grudgingly, "sound film is not only destructive but also offers artistic potentialities of its own." (ibid..) He expands a considerable portion of *Film as Art* engaging in this double movement, criticizing color film while discerning its possible strengths, deploring the hyper-realism brought by stereoscopic apparatus while acceding to certain uses for it. In so doing, he not only anticipates Maeda's call for a native medium, but also makes a strong argument for encouraging diversity within a medium: "In itself, the perfection of the [color, sound, and wide-screen] film need not be a catastrophe - if silent film, sound film, and colored sound film were allowed to exist alongside it." (ibid..)

Arnheim's critique of the drive towards realism goes further than a discussion of the opportunity costs. Art, in Arnheim's view, is in part a matter of picking the details out of the world and bringing them to the attention of the viewer. In his eyes, part of the fundamental power of film is in its ability to very precisely focus that attention. As the

technological apparatus of the medium became more complex, this focus became harder to attain, and as this happened, the artistic essence became diluted.

The relevance of this argument is particularly pertinent in the digital medium's present phase of development. Much of the field has been skewed toward "multimedia", focusing on the multiplicity of mediums subsumable by the technology. Much of what remains strives to the hyper-real ideal of virtual reality or *Toy Story*-like simulation. These are the sound film and wide-screen film of the present age. Speed, processing power and complexity become reasons in-and-of themselves, imposing an overwhelmingly technological imperative. Arnheim's forlorn plea to continue explorations beyond the boundaries of the monolithic, veridical "complete" film should serve as notice to the present. Instead of chasing the technology, we should lead it in the many directions our imaginations can take us.

2.3.2 **Computer Art/Digital Art**

Arthur Danto, media critic for *The Nation*, makes a distinction between "TV art" and "video art" which can be useful both productively and critically. He writes

> [L]et us distinguish 'TV art' from 'Video art.' TV connotes primarily a form of life – a room in the middle-class household, a frozen dinner – that the television set symbolizes and facilitates, so that modifying 'the tube' one is in a certain sense engaging in a gesture somewhere between social criticism and outright iconoclasm...This leaves 'video' to refer to images that owe their provenance to the same technology as television but make no internal reference to their organs. (Danto, 1995)

In the spirit of this definition, we can make a similar distinction between "computer" art and "digital" art. Computer art is that which does make "internal reference to [its] organs", which self-consciously proclaims that its existence involves, even is dependent on the computer. In the same way that Danto describes TV art as connoting an entire environment and its accompanying mind-set, computer art deals directly with the impact of computers on our society, our modes of communication, and our self-representations.

Digital art, on the other hand, is simply art made with the aid of, or presented with the aid of a central processing unit. I choose to say "CPU" because it is immaterial whether an object that looks like a computer – screen, keyboard, big box – appears anywhere. What is material is the *functionality* of the CPU, and it can be placed anywhere within the piece, usually hidden in the same way you hide the projector in a movie theater.

Contributing to current content-lag within the digital media is an emphasis on computer art, a preoccupation with the machine as machine, as powerful, intelligent machine capable of doing somersaults around traditional ways of presenting information or emotion. In itself, this is not a negative development. The space proudly and intentionally labeled "computer made" is vast and should be fully explored. But because work of this sort forms a majority of that art that is produced, computer art creates the impression that pieces produced via the computer are all bound to the same sort of self-referential proclamations about how fascinating computers are.

2.3.3 **Well-trodden Paths – Hypertext and Usability**One of the earliest proposed digital extensions of text is that of *hypertext*. A hypertext is a text in which the various references it employs – whether to bibliographic items, illustrations both static and dynamic or anything the author finds to be of possible related interest – can be accessed via an electronic link. The link may point to something which resides in the local digital environment or to something that lies half a world away and is mediated by the Internet. First conceptualized as an analog system by Vannevar Bush in 1945, hypertext has come into maturity with the current World Wide Web protocol.

Some observers of the digital evolution of text, such as Bolter and Landow have advocated hypertextuality as the defining structure of these new forms. (Bolter 1991; Landow 1992) While the ease of reference and the support it provides for non-linear narrative and information structures have become quite important, hypertext is only one among the many sorts of text-led interactions possible within the digital environment. Like multimedia, it has a place within a matrix of digital interactivity, but it should not occupy a place of prominence. As many of its finer details are presently being worked out quite capably within the World Wide Web and such authoring tools as HyperCard, I have chosen not to deal explicitly with hypertext within the Dynamic Poetry project.

Another great concern of many working within the digital medium is the notion of *usability,* or the ease of use which a system displays. In practice, usability metrics have been applied variously to questions of functional efficiency, conceptual clarity and interactive simplicity. Despite its multiple, and sometimes overlapping, applications, usability *is* useful when discussing systems which support the exchange and analysis of information. It is also of use when discussing the viability of the designer-ly aspects of any digital system. However, it is of limited utility in a discussion of systems in which the desired response is emotional and aesthetic, and in which ease-of-use is a secondary consideration or perhaps even irrelevant to both the creator and the audience.

In an amusing reference to this distinction, Bolter writes that "the great advantage of the first printed books was *not* that you could read them in bed. Gutenberg might well have been appalled at the thought of someone taking his beautiful folio-sized Bible to bed." (Bolter 1991) He goes on further to point out how the gain in legibility offered by the uniformity of letterpress type was obtained only at the cost of, among other things, the idiosyncratic beauty of the manuscript. Books may indeed have been more usable, but this did not necessarily mean that they were more desirable.

Bolter's point should be taken to heart as the reader continues through this thesis. While I have taken great pains to ensure that the experiments I created were accessible to the user, I was far more concerned with the emotional and aesthetic impact and clarity of message than with how quickly, how efficiently and how simply the message was communicated.

Chapter 3 The Word

...we can't write sonnets any more because we no longer live in the sonnet's world. We need a form or, it is more likely, forms organic to the nature of our own world...
– Mary Ellen Solt (1971)

3.1 **Poetry**

Poetry disrupts what have become normal reading habits. Whether in the form of a sonnet proclaiming its careful structure, or a haiku boldly turning on just three lines, or a piece of free verse condensing and distilling experience into the least number of words possible, poetry demands that the reader approach it carefully and with an open mind. Poetry requires a sharpened attention to the multiplicity of meaning carried by individual words and multiplied even further by their combination. In undertaking a series of experiments designed to stretch the language of interaction and to encourage the user-reader to think carefully and consciously about the nature of interaction, poetry is a uniquely appropriate vehicle.

Formally, poetry provides a robust arena in which to observe the close exchange between form and content. The ways in which meter, rhyme, alliteration, assonance, etc., work to amplify the meaning of a poem provide a rich example for how interactive characteristics must be employed with the same care as such literary devices.

A couple of examples will illustrate this interplay between form and content. In a sonnet rhyme and meter conspire with overall structure to advance a theme, inject tension into it and then resolve that tension. Witness Shakespeare's Sonnet 130:

My mistress' eyes are nothing like the sun;
Coral is more red then her lips' red;
If snow be white, why then her breasts are dun;
If hairs be wires, black wires grow from her head.
I have seen roses damasked, red and white,
But no such roses see I in her cheeks;
And in some perfumes is there more delight
Than in the breath that from my mistress reeks.
I love to hear her speak, yet well I know
That music hath a more pleasing sound;
I grant I never saw a goddess go;
My mistress, when she walks, treads the ground.
And yet, by heaven, I think my love as rare
As any she belied with false compare.

This poem, while an adroit commentary on the sonnet-writer's usual declamation of the wonder of the love-object (cf. Shakespeare's own Sonnet 18, "Shall I compare thee to a summer's day..."), also reinforces the power of the form. The poet begins with a certain lack of praise for his mistress which soon becomes positively insulting with "my mistress reeks." As the last quatrain (four lines) unfolds, he reins in the visceral imagery of the first two, finishing it with the sensible observation that she walks upon the ground, just as does everyone else. Then the poem takes the classic turn in the final couplet, where the poet reveals that he does indeed love his mistress deeply, so deeply that he would not do what so many other poets do and besmirch his love with the hollowness of hyperbolic comparison. In sixteen lines, the poet has taken the reader from despair on behalf of the supposedly maligned mistress to acknowledgment of her (as well as the reader and his lover's) profanity to an understanding of their love which is at first perverse, but then upon reflection, becomes commonplace. This concentration of dramatic development is fundamentally reliant on the ability of the rhyme to pull the reader along by listening to the moments of completeness it provides and reaching for the satisfaction of the final couplet; on the iambic pentameter (stressed/unstressed syllables with five stress per line) provide a steady rhythm; and on the poem's invoking of the sonnet form while turning the content on an unexpected course.

An example of a far more strict form which draws power from the restriction is that of a villanelle. Not only does it require a tight rhyme scheme, but it also dictates that the poet re-use portions of the first tercet (three lines) in specific locations in the following four tercets and the final quatrain–and in a meaningful, non-repetitive manner. Duane Niatum's *The Art of Clay* demonstrates:

> The years in the blood keep us naked to the bone.
> So many hours of darkness we fail to sublimate.
> Light breaks down the days to printless stone.
>
> I sing what I sang before, it's the dream alone.
> We fall like the sun when the moon's our fate.
> The years in the blood keep us naked to the bone.
>
> I wouldn't reach your hand, if I feared the dark alone;
> My heart's a river, but is not chilled with hate.
> Light breaks down the days to printless stone.
>
> We dance for memory because it's here on loan.
> And as the music stops, nothing's lost but the date.
> The years in the blood keep us naked to the bone.

How round the sky, how the planets drink the unknown.
I gently touch: your eyes show it isn't late.
Light breaks down the days to printless stone.

What figures in this clay; gives a sharper hone?
What turns the spirit white? Wanting to abbreviate?
The years in the blood keep us naked to the bone.
Light breaks down the days to printless stone.

In this piece, the poet meditates on the spiral nature of time passing: the return of darkness each day which is yet a different darkness each day ("So many hours of darkness we fail to sublimate"); the music stopping after yet another game of existential musical chairs; and the light shining day after day as it wears down the "printless stone" of the mind. The villanelle form amplifies these themes: the rhyme works with the repeating lines to create an almost hypnotic rhythm which circles in on itself like the images contained in the poem. Returning to the beginning, the end possesses the inescapable closure of death.

Moving to the other extreme, we encounter the unfettered structure of free verse. Unlike the formal demands of the sonnet or the villanelle, free verse dispenses with rigid rhyming schemes and line lengths. Yet this does not imply that examples of the form do not possess a structure. Rather, the structure they do have is unique to any particular poem, and just as essential to its intent. Consider *Your hand full of hours...* by Paul Celan:

Your hand full of hours, you came to me–I said:
Your hair is not brown.
So you lifted it lightly on to the scales of grief; it weighed more than I...

On ships they come to you and make their cargo, then put it on sale in the market of lust–
You smile at me from the depth, I weep at you from the scale that stays light.
I weep: Your hair is not brown, they offer brine from the sea and you give them curls...
You whisper: They're filling the world with me now, in your heart I'm a hollow way still!
You say: Lay the leafage of years beside you–it's time you came closer and kissed me!

The leafage of years is brown, your hair is not brown.[3]

Like Sonnet 130, this is a poem about love, not a love poem; the negatively declarative structure of both offering up simple comparisons only to withdraw them into the true

[3] Translated from the German.

complexity of the emotions, e.g., "My mistress' eyes are nothing like the sun" and "Your hair is not brown." Like *The Art of Clay*, Celan's poem utilized repetition to diffract the unitary perception of time ("Light breaks down the days to printless stone" and "The leafage of years is brown, your hair is not brown."), and to create an adamant, fully-closed ending. Like these other poems, the structure of *Your hand full of hours...* is used to advance the imagery of the words in a way which serves the purposes of this particular poem and none other. Celan is renowned for his hyper-dense, compact verse, and he is just as well known for the inventiveness he employs in arranging structure to assist in interpreting his linguistic labyrinths. The density of his writing is no accident, or willful obtuseness, but an organic reflection of his message. As one of Celan's anthologists Michael Hamburger (Celan 1988) writes:

> The impossibility of writing poems after Auschwitz, let alone about Auschwitz, has become a critical commonplace. Celan knew that even he could not hope to do so directly, realistically, but only by an art of contrast and allusion that celebrates beauty and energy while commemorating their destruction.

The fracturing of individual psyche and collective unity engendered by the Second World War finds a reflection in his employment of a hyper-dense yet evasive style.

All three of these poets involve the reader deeply in the play of form and content. They represent poetry at its strongest, carefully orchestrating the presentation of material which is mundane in its ingredients–they employ words that anybody can use–to create artifacts which are concisely evocative in their effect. It is these characteristics which drew me to employ poetry as the vehicle for exploring the limits of digital text.

3.2 **Concrete Poetry**

Shakespeare, Niatum and Celan's poetry sits within the mainstream tradition which considers the appearance of the words as a neutral factor, concentrating on the concepts behind the words, and to a certain extent, the sound of the words. Yet, by definition, the inscribed word has a visible aspect, which has consequences for the entirety of what the reader takes from the poem. In the case of traditional poetry, a conventional range of typefaces and presentation allow the reader to ignore the visual appearance, to factor it out of the poetic equation.

Yet when the visible appearance of the words is treated as a principal dimension in the production of poetic meaning, a new field of creative play is opened up. Letterform and layout operate semantically within the poem, or within any inscribed

material so treated. In the history of the word after the invention of the printing press, such visual manipulation of type has been an ongoing effort, but until the advent of the modern period these efforts remained fairly circumscribed. Beginning in the late teens and early twenties of this century, the activities of the Futurists and Dada, and the work of Guillaume Apollinaire and Tristan Tzara among others, broke down those boundaries in every direction.

The members of these movements felt played freely with both letterforms and layout, disdaining the industrialized identity of traditional type and the rigidity of rectilinear layout. Italian Futurist F.T. Marinetti spoke for many of his compatriots when he said:

> I am against what is known as the harmony of setting. When necessary, we shall use three or four columns to a page and twenty different type faces. We shall represent *hasty perceptions* in *italic* and express **scream** in **bold** type...a new, painterly, typographic representation will be born on the printed page. (Spencer 1969)

One of Marinetti's pieces, *Apres la Marne, Joffre visita le front en auto* clearly exhibits this determination to employ the visual element of typography to its fullest effect. (Figure 1) It shows the crazy path taken while driving to the front after the First World War Battle of the Marne. Words follow the course taken, dipping in and out of French valleys (the big M's), passing through the fields of the dead (the X's which are also crosses) and overtaking regiments of soldiers (inside the U at the bottom). *Club Dada* and *Une Nuit d'Eches Gras* further exemplify the movement's chaotic jumbling of type orientation, size and face. (Figures 2 and 3)

An even more powerful embodiment of the "dynamic visualization" of type (a term taken from *Apres la Marne*) can be seen in *Lettre-Océan* by Guillaume Apollinaire. (Figure 4) Apollinaire wrote many calligrammes, or figurative poems, in which he sought to give form as well as expression to common experience as part of a theory that such experience was as much of the body and the environment as of the mind. *Lettre-Océan* is one of his most well-known pieces, bringing images of the ocean, the sun and the pyramids of the Mayas into a tight exchange with the printed word. These sorts of typographic and poetic experiments combined with a growing advertising industry constantly searching for novel ways in which to entice consumers subjected both typography and page layout to radical re-imagining. (Drucker 1994)

The Concrete Poetry movement, which began in the mid-50's and gained critical mass in the mid-60's, is a direct descendant of such early pioneers. To the poets involved

in this movement, the sonnet was a form to be cherished, but not sustained in a world of technological change, political upheaval and sexual revolution. Within such a context, calling up the sonnets calm rhythms and carefully ordered rhymes was an act of willful denial. The world was stretching into unrecognizable, chaotic shapes, and poetry, they felt, needed to stretch with it.

These poets, beginning with the de Campos brothers in Brazil and Eugene Gomringer in Switzerland, embarked on their experiments with a fervor similar to that of the Dadaists seeking to make sense of the changes brought about at the advent of modern warfare. Mary E. Solt (1971), in her survey of the field, *Concrete Poetry: A World View*, notes their attitude that traditional poetry could not keep pace with the way life was being lived, in that time and in those places. Their reaction, then, was almost spiritual, a need for an expressive form which would distill the *Zeitgeist*. They aspired to an expressivity which would capture the sudden expansion in humanity's environment offered by the exploration of space and the "concentration and simplification" offered by the on-rush of telecommunications technologies.

The range of experimentation carried out by this movement serves as a reminder of how limited currently accepted poetic forms are. In *Here are the Lovers*, Augusto de Campo uses visual oppositions to reflect the semantic oppositions of "above*i"* and "below*she*", "lovers" and "parents". (Figure 5) Carl Fernback-Flarsheim's *Mirror Field Inside Random Field* has the appearance of a sub-atomic collision, where the core particles retain their stability as the surrounding particles begin to destabilize and then fly outward with the force of the impact. (Figure 6) These poems explore space as an extension of semantics and use words to create objects and objects to traffic in the meaningfulness of words:

> ...the non-linguistic objects...function in a manner related to the semantic character of words. In addition to his preoccupation with the reduction of language, the concrete poet is concerned with establishing his linguistic materials in a new relationship to space (the page or its equivalent) and/or time (abandoning the old linear measure)...[T]he concrete poet is concerned with making an object to be perceived rather than read. The visual poem is intended to be seen like a painting; the sound poem is composed be listened to like music... (ibid.)

Thus we see that the Concrete Poetry movement focused on the material out of which the poem is constructed. The medium no longer disappears into neutrality, nor is it ignored. Instead, it becomes an integral part of the expressive form.

The influence of the Concrete Poetry can be seen in modern advertising and design. However, their demand for new ways of communicating given new ways of seeing and mutating ways of being has fallen by the wayside. The role of any art is to craft anew the apparatus by which we struggle to apprehend a changing world. Along with a re-evaluation of the ways in which words can be presented and received, that spirit is what the present project takes from those who began – and continue with – Concrete Poetry.

The Dynamic Poetry project used each of these forms as models. The attention to the power of language found in traditional poetry, the "dynamic visualization" found in the work of the Futurists and their fellow travelers, the conflation of medium and language found in the Concrete poets–the experiments described in this thesis employ these approaches to explore the space of interactive, evocative text. It is clear that the impact of a digital media on poetry will be no less significant than the impact it is already having on narrative and image. More than a decade after the advent of the personal computer, the time is ripe for suggestions about how the poetic form might migrate into this new environment and how the pioneering spirit of Apollinaire, de Campos and others can be infused into new territories of textual experimentation.

3.3 **Typography**

Carl Dair, at the end of his book *Design with Type*, muses about the opening frontiers of "typography in motion", pointing towards television, film and computers as possible sources of advancement in this direction. (Dair, 1967) This was almost 30 years ago. Since then, while experimentation within the field of moving pictures has continued apace, much of the work within the digital realm has been directed at either improving resolution or radically re-imaging letterforms. The Cranbrook theorists' drive to deconstruct the visual aspect of type, David Carson's magazine work and the font experiments of the designers associated with FUSE, to name but a few, challenge basic notions of typography and textual presentation. (figures 7, 8, and 9) Their work has been inspired in some part by the manipulations available to designers working within a digital environment. Some even attempt to go beyond the visual dimension. Erik van Blokland created *Nimida* which randomly degenerates its letterforms, and Jonathan Barnbrook designed *Burroughs* which transforms the user's typing into gibberish. (Figures 10 and 11) They represent two of the very few efforts being made to design type and type-usage which possesses time-based characteristics and/or responds to user input. Otherwise, the potential of text within the digital environment remains largely unexplored.

This untapped potential is partially the result of technical difficulties. The amount of processing power required to intelligently animate hundreds of words, and thus thousands of letterforms, is beyond that which is available in personal computers. In addition, most programming environments which are sufficiently sophisticated to provide an appropriate level of control over appearance and movement over time for so many objects require a level of programming expertise beyond that which most designers or artists possess.

These conditions make it is no surprise that a significant majority of the work that *is* done in this direction takes place at Massachusetts Institute of Technology's (MIT) Media Laboratory. In MIT's Visible Language Workshop (VLW), access to high-levels of computing power and programming talent have allowed innovative exploration of this area. In general VLW has pioneered the three-dimensional display and navigation of text-based information spaces. (Figure 12) In particular, Suguru Ishizaki, David Small and Yin Yin Wong have contributed. Ishizaki (1996) has created an email environment in which the dynamic appearance of email headers convey information about the urgency, source and age. He has also proposed an agent-based approach to textual behavior which is a more computationally refined version of the *Breeder* experiment described below. Small (1996; MIT VLW 1994) has developed a notion of "expressive typography" which involves using physical models of object dynamics to drive the movement of type and employing "wet" typography as way of recapturing the rich visual interaction between pigment and paper. Wong (1995; 1996) has developed the concept of "temporal typography", or the dynamic visual treatment of written language, as a means of extending the range of typographic expression. (Figure 13)

The efforts of the VLW group have gone far in expanding the ways in which text can be treated in the digital environment. Their work focuses on articulating the time-dimension and enriching the visual play of text. For most part, however, their work has been in creating functional extensions of the typographic form, ones which are in service of software to support electronic mail, financial analysis and literary reference. The poetic context–whether actually for poetry or some other expressive purpose–have not had the benefit of such treatment.

Chapter 4 Interactivity

4.1 **The Need for Definition**Interactivity is often identified as the defining quality of digital media. Yet its meaning is not well articulated. In a field where the simple "yes/no" button clicks required of the home-shopper are described with the same term as the intricate structure and sophisticated interfaces of complex narrative pieces, interactivity is simultaneously everywhere and nowhere to be found. The elusiveness of the term extends content-lag by obscuring more issues than it clarifies. To be useful as an analytical term, both for those who wish to study the medium and for those who wish to create within it, the term must be better defined.

Consider the "Interactivity Matrix" which appeared in the premiere issue of *Wired* magazine, one of the bellwethers of the digital form. (Figure 14) Developed in the process of researching the telepresence potential of virtual media, it provides an instructive example of just how broadly interactivity can be, and is, construed. Books are on one end of the spectrum, the Star Trek holodeck on another. The distance between them covers an enormous amount of functional, aesthetic and experiential ground. Furthermore, the change that takes place over that ground is not additive, i.e., it is not a matter of adding more functionality to a book until one ends up with the holodeck. What, then, is it that makes the holodeck "more" interactive than a book?

In Chapter 2 I introduced John Maeda's manifesto on digital media. He has also done concrete work towards defining interactivity and establishing the media on its own terms. He argues that designers must, among other things, fully employ the time dimension and explore "reactive graphics, visual experiences that respond to user input in realtime in a way that defies physics (not virtual reality) and are devoid of content (not interactive media in the ordinary sense)." (Maeda, 1995:a) Reactive graphics, then, extend the dynamic realm by exploring actions which cannot be performed in the physical world, and which respond immediately to the user's own actions. Maeda's efforts are provocative arguments for this approach. The textual playspace of his *Flying Letters* collection is of particular interest within the context of this paper. In one segment the alphabet plays out wherever the user's cursor is located, each letter fading into the background with exquisite slowness. (Figure 15) The cursor moves with a sly delay, giving the user the feeling of leading a slightly recalcitrant cat around by the leash. In another segment, two different sentences are spelled out, word by word, depending on whether the user has the cursor in the top half of the screen or the bottom half. (Figure

16) In the top half, “to the heavens” comes spilling out; in the bottom half, “to the earth.” The same sort of fade as in the previous example lends this piece a certain softness. Both pieces contain interactions which are elegant and simple. They *react* in realtime and their actions make their own physical sense. Neither are reproducible outside of the digital medium, nor are any of the other segments on the *Flying Letters*. Instead they affirm the new media’s need for their own particular philosophy and approach, and provide convincing attempts to work out such a philosophy via concrete prototypes.

One drawback to Maeda’s work is that the interactions he showcases are extremely abstract. They point the way to a new vocabulary, but his reluctance to simultaneously pursue content leaves the user without any context in which to evaluate them. His reasons for this abstraction – to focus on the form – are laudable, yet one yearns to see those forms developed in concert with content that would be equal to them, to see reactivity *mean* something. If an aesthetic native to the digital medium does exist, it must prove itself capable of communicating content of substance.

Jim Campbell, a San Francisco-based installation artist, has proposed a different way of looking at interactive work. He sees it as existing on a spectrum with controllable systems on one end and responsive systems on the other. The first endpoint emphasizes the user’s control over the system, providing him with an easily definable and one-to-one synchronicity between his actions and the reaction of the system. The “conversation” between the user and the piece can be characterized as being command-and-comply. Campbell cites most CD-ROM-based pieces and games as examples which inhabit.

At the other end lies work which the user’s actions within the environment (not just a single point, such as a mouse) are interpreted by the system in a surprising yet meaningful manner–surprising in that the user cannot predict the exact consequences of an action, meaningful in that the system is not responding in a completely random manner but one which exhibits an overall direction and rationale. This type of conversation can be characterizes as being much more like a dialogue, with the user and the system both posing questions to one another. The user’s attention is not on manipulating the system, but in having an interesting experience with it. (Campbell 1996)

An experiment of Dynamic Poetry project, *Aura* (section 5.4.1), provides a good example of the experiential end of the spectrum. On approaching *Aura*, thought it is clear to the user that his actions are causing a reaction by the system, the mapping between the two is quite ambiguous. However, instead of being frustrating, this ambiguity is integral to the piece’s cloak of ritual mystery and the careful way one should approach ritualistic objects.

Like Maeda, Campbell addresses time. "If the new element to film was time," he says, "then...the new element to interactivity is the present." (ibid.) Many interactive pieces react at their own pace, not that of the user's. Sometimes that reaction is too slow, such as with CD-ROMs, but it can be too fast as well. Truly responsive systems, however, interact with the user right here, right now, at a speed which neither puts the user to sleep nor baffles him.

Campbell's interest is in transcending the traditional computer science command-and-control paradigm which leave users frustrated and bored. Maeda's interest is in transcending the tradition-bound presentational methods of analog media which fail to exploit the particular strengths of the digital environment. In the course of the Dynamic Poetry project I have sought to revise my own definition of interactivity in ways similar to that of Maeda and Campbell, and in ways which make the interactive distance between a book and the Star Trek holodeck meaningful and operational.

Using the experimental work I undertook as a concrete grounding, and using Campbell's characterization of interactivity as a dialogue, I have come to a definition which deconstructs interactivity into a dual perspective. *Dynamics* refers to the way in which the system "sees" the world, i.e. which objects activate under what conditions. *Response* refers to the way the user sees the world, i.e. what kind of activity occurs when he undertakes actions. As it is integral to all activity at the interface, I will also propose a definition of *time* which is articulated specifically for the digital environment.

4.2 **Dynamics**

Dynamics describes an interaction from the viewpoint of the program. For the purposes of this discussion, the program can be thought of as a set of instructions which dictate what happens when to which objects. The term can be used either globally to describe the general character of a piece or locally to describe the character of particular components of a piece.

The dynamics of an object (again, whether a whole piece or element) can be *constructive, reactive, active* or *static*. To use the dialogue analogy, a static object does not contribute to the conversation in any way other than its presence. An active object is one which holds forth in a monologue. A reactive object does what its conversational partner asks it to do. At the other end, a constructive object engages in a full-blown conversation from which both partners take away new thoughts.

4.2.1 **Constructive**

An element which possesses constructive dynamics is one which allows the user to add to or subtract from it. Such alterations, where the user changes the "dataspace" of the piece by typing in text or excising out an image or recording in video, become integrated into the piece. The interaction between the user and the piece is in the form of an exchange.

WordNozzle (section 5.2) is an example of a piece which is highly constructive. The user creates a text-pouch which then serves as the source for the text. He then has full control over the appearance and placement of that text as it sprays out of the nozzle. The constructive option is open to the user at all times, allowing him to fully determine the final product.

4.2.2 **Reactive**

A reactive element responds to the user's actions, though not to the extent that a user can change its composition. Most multimedia, and indeed, most digital media is composed of elements which exhibit a reactive dynamic. Though similar to Maeda's use of the term, I define the term more extensively. The reaction may or may not follow a virtual physics; in line with the main thesis of this paper, it definitely possesses a semantic which supports the content of the piece.

Most of the experiments in the Dynamic Poetry project contain substantial reactive elements. An example are the stanzas in *Dying Lying Rotting* (section 5.3.2) which require the user to zoom into them to read further. Another example in that poem are the gliding stanzas, which can be moved by the user and which divulge more lines when the user repositions them.

4.2.3 **Active**

An active element changes its physical appearance over time, independent of the user's actions. In the simplest example, an animation, once begun, undergoes change in a purely active manner. Active elements–and pieces which are composed of active elements–function much like a movie does once it is started. The action unfolds at a predetermined rate and in a predetermined direction. *Telecommunication* (section 5.3.5) is an active piece, with the various stanzas in constant motion.

4.2.4 **Static**

A static piece exhibits no change. I have included mainly to establish one endpoint on the dynamic range, and in order to cover products such as fractal painting

and hyper-layered images which are particularly digital but which do not change appearance or composition.

4.3 **Response**

Response describes the means by which the dynamics of an element are initiated or maintained. This perspective contains *dependent, independent* , *hybrid* and *non-responsive* elements, where "dependent" means "dependent on the user's input." As with dynamics, response is applied to both individual elements and to a piece in its entirety.

4.3.1 **Dependent**

A dependent element relies solely on the user's input to determine the action it exhibits. This term covers a significant portion of current digital pieces, in which the user clicks his way through a series of scenes.

Most of the experiments done for the Dynamic Poetry project contain dependent elements. *Telecommunications* (section 5.3.3) consists entirely of dependent elements: each stanza only moves when activated by the mouse passing over it, and the speed of movement is directly related to the position of the mouse relative to original point of intersection.

4.3.3 **Independent**

An element which is independent acts without any input from the user. *Breeder* (section 5.3.1) is a good example of piece whose response in completely independent. The individual words already "know" how to interact with one another, and initiate that interaction on their own. Their syntactic communication and visual assemblage takes place regardless of the active involvement of the user.

4.3.2 **Hybrid**

Hybrid refers to elements which act both independently of and dependently on the user's input. Hybrid elements often seem the most life-like, or natural, as they reflect an organisms ability to take independent action and to respond to action that it is taken on it. It also captures longer term complex dynamics, in which an element ages or transforms itself according to an internal clock all the while accepting external input which materially effects it.

The candle in *Aura* (section 5.4.1) responds in a hybrid manner. When no user interacts with it, it burns merrily away, randomly switching between different video clips in order to simulate the erratic movement of a flame caught in subtle cross-drafts. When a

user enters the environment, the candle switches modes and responds to the user's movement. Extinguishing the candle and relighting it are completely dependent on the user's actions.

4.3.4 **Non-Responsive**

This category is substantially identical with static. It establishes the point against which other forms of response are measured, i.e., no response.

4.4 **Time**

Time in this context refers to the clock which determines the rate at which action takes place, i.e., how often an element advances from any state *t* to any other state *t + 1*. Time can be characterized as *cycle-time* to *real-time* to *interaction time*.

4.4.1 **Cycle-time**

Cycle-time is the speed at which the Central Processing Unit (CPU) itself runs. In the present generation of personal computers, this rate can be anywhere from 60 MHz to 200 MHz-plus (MHz = thousands of cycles per minute). Technicalities aside, the main effect of increasing cycle-time is that any given set of actions performed by the computer will be completed within a smaller amount of real-time. An event programmed to occur at the beginning of the main event loop of a program will happen approximately twice more often on a 200 MHz machine than on a 100 MHz machine. *Telecommunication* (section 5.3.3) is an example of piece which operates exclusively in cycle-time; this mode was chosen to reflect the poems concern about the effect of technology, and the speed of technology, on interpersonal communication.

4.4.2 **Real-time**

Real-time is the time is the time in which we, as people, operate, and is effectively the base to which the other times are compared. Depending on the action, real-time may be perceived to be faster or slower than cycle-time. Rotating a half-bitten apple through 180° will often take longer in cycle-time (as a rendered object) than in real-time (as a material object) because of the processing power needed to perform the math involved; copying a document in real-time will almost be always be slower than copying it in cycle-time. *Dying Lying Rotting* (section 5.3.2) operates in real-time, as the lifetime of the gliding stanzas are tied, code-wise, to the real-time clock of the machine. Regardless

of whatever speed at which the host CPU is running, those stanzas should always live for the same amount of real-time.

4.4.3 **Interactive-time**

Finally, there is *interaction-time*, which equates a cycle to every interaction undertaken by the user. In such a sense, time does not move forward until the user takes some action. Each time the user acts, time passes. Though not implemented, I considered using interaction-time in *Cross Purposes* (section 5.3.4.) Each time the user interacted with a particular stanza, its complementary stanza would age by one increment. Eventually, if one particular stanza was viewed quite often, its complement would disappear from screen. The poem's commentary about point-of-view would be reinforced, as the user concentrated on one viewpoint to the extent that the other one disappeared.

These three sorts of time are not exclusive. A piece could make use of all three, for varying reasons. Exploring the semantics of the different times, both singly and together, should provide further avenues of fruitful experimentation.

Like Maeda and Campbell's efforts, this characterization is not meant to be exclusive nor definitive. Rather it is an attempt to give greater definition to an amorphous term in the hopes that others will be able to use it as inspiration for thinking about and creating in the digital environment.

Chapter 5 Experiments

This section describes the experiments conducted for the Dynamic Poetry project work. Each experiment's visual appearance and interactive characteristics are described, and the result is discussed.[4] Except for *Aura*, which I completed before any of the rest, *Life is Bait*, I created between *WordNozzle: desktop version* and *Dying Lying Rotting*, and *WordNozzle: installation version*, which I made last, the pieces are presented in chronological order. This ordering shows the progression of my ideas about and understanding of the notion of a digital poetry based on interactive text.

Within the chronological ordering is a thematic grouping. "Conversions" refers to my first explorations of the design space. The two pieces discussed reflect an attempt at converting existing poems into a form more suitable to a digital environment. I came to regard this repurposing approach, i.e., taking a creation which was made for one medium and recasting it into another, as inappropriate. The remaining experiments are poems composed specifically for the digital medium. The next section, "Concrete Poetry Revisited," deals with the two manifestations of the *WordNozzle* concept. Among other things, *WordNozzle* was an attempt to update the Concrete Poetry approach to visual language in a way sympathetic to the new medium. The section following "Concrete Poetry Revisited," entitled "A Digital Poetry," deals with the final series of Dynamic Poems. These pieces are the fruit of all the other experimentation and the theoretical framework of interaction developed in the process.

Finally, in "Beyond the Word," I present *Aura* and *Life is Bait*, two experiments which are not based around interactive type but which do explore some of the space delineated by an interest in developing pieces native to the digital medium.

5.1 **Conversions**

Flash and *Scratch* were directly inspired by *Poetry in Motion,* a CD-ROM anthology of poetry put out by the Voyager Company. The inspiration was positive, in that I had been impressed by Voyager's willingness to explore the presentation of poetry in the digital realm; and it was negative, as the various media used in the presentation of the poems –

[4] As has been mentioned previously, a discussion of the technical details of the various pieces can be found in Appendix A.

text, video, image and sound – are not well integrated with one another, and the design of the textual element in particular is negligible. *Flash* and *Scratch* are based on two different poems from the *Poetry in Motion* volume, Amiri Baraka's "Wailers" and John Berrigan's "Whitman in Black." They are designed to develop a rich textual presentation. In the end, though they bring to light many issues which fueled subsequent experiments, in and of themselves they are not completely successful. The difficulties I encountered in trying to graft interactivity onto these print-native poems led the project away from repurposing efforts and further motivated the argument for native media argument advanced in this thesis.

5.1.1 ***Flash I & II***

[Dynamic Poetry Experiments:DP Direct Access:Flash I/Flash II]

DESCRIPTION

Flash I & II were aimed at increasing integration between media elements and imbuing the text with interactivity. To the first end, I moved away from the original discreetly blocked layout find in the *Poetry in Motion* original. (Figure 17) Instead, I used a single large still for the background to create an all-encompassing visual atmosphere. The QuickTime movie of Baraka performing "Wailers" is brushed into the left-hand side of the screen in an attempt to negate the strong border effect of the standard QuickTime window.

In *Flash I* the text is not bound to any particular place on the screen. (Figure 18) In fact, the text itself is the cursor and can roam anywhere. If the user moves the cursor up or to the right, the text displayed at the pen moves forward through the poem in one line chunks; if the user moves it down or to the left, the poem moves backwards in the same increments. (Figure 19)

In *Flash II* the text is again unanchored, but instead of scrolling through the poem a word at a time, individual lines of the poem are scattered around the screen. (Figure 20) By moving the cursor over a line, the user activates the subsequent line, causing it to move underneath its predecessor. This can be done until the poem is fully reconfigured into its original linear form. (Figure 21)

DISCUSSION

Flash I & II present the user with an interface which is visually more integrated and interactively more dynamic and responsive than the original version. The video is no longer a separate or alternative means of experiencing the poem, but of a piece with the

text. In *Flash I*, the text only reveals itself as the user interacts with the poem, forcing him to take an active role in the reading of it. In *Flash II*, the user is again asked to play a role in constructing his experience of the poem, but here the reading takes on a more playful feel. The user composes two-line segments, then three-line segments, onward until the entire poem is reconfigured. This action creates a montage-like impression, with the sensible-but-not-quite-meaningful parts slowly resolving into a coherent whole. In a sense, the user "discovered" the poem instead of simply reading it. In both pieces, the combination of active and reactive, dependent and independent elements created a rich encounter with the piece.

However, this increase of activity brought along its own problems. In *Flash I*, the user never saw the complete text of the poem, so the notion of forward and backward did not mean much. Being exposed to one line at a time – especially when not in a single sequential direction – hindered the user's ability to maintain the relationships between non-adjacent lines. With *Flash II*, the playfulness of the interface became its defining feature, leaving the user feeling like he was having to jump through hoops simply to read it. In both versions the audio accompanying the video was so strong that it led the experience, and when users found that not only did the cursor-text not necessarily relate to what was being spoken currently, but that it also had no effect on the progression of the video component, they became frustrated by the illusion of more control than actually existed.

5.1.2 ***Scratch***

[Dynamic Poetry Experiments:DP Direct Access:Scratch]

DESCRIPTION

Like the two versions of *Flash, Scratch* is an experiment in improving the visual presentation and interactive level of an existing piece, the *Poetry in Motion* segment on Tom Berrigan's *Whitman in Black*. (Figure 22) I paid particular attention to two areas in which *Flash* is weak: giving the user access to an overview of the poem and providing mouse-based control over the progression of the audio and video. The latter concern had a particularly strong impact on the design, and led to the "discovery" of using the random access capabilities of the digital video to "scratch" the audio track in a way similar to that used by DJ's.

Scratch begins with an introductory screen which allows the user to select a view of the poem as it was originally written, in standard linear form. The user can also choose to see an interview with the poet. The third choice takes the user to the piece proper.

Compared to the clean, minimal *Flash* , the visual chaos of *Scratch* is quite striking. (Figure 23) The video has been almost completely obscured beneath accreted words, with only brief glimpses of movement to give away its presence. In contrast to *Flash*, I chose to concentrate on the sound in *Scratch*. because the visual aspect of Berrigan performing is uninteresting compared to Baraka's. The visual component of Berrigan's video remains visible only to an extent sufficient to evoke its presence, leaving the sound to occupy the most prominent place in the user's perception.

The user can perform any one of three actions. He can move the cursor over the video area and treat it like a miniature *Flash*; if he stays within that area, movement up and to the right spools through the lines of the poem in a forward direction, the opposite movements would have the opposite effect. If he moves out of the video spot and clicks on the textured surface, the first line of the poem pops out. Successive clicks spit out each successive lines. If he clicks on a line that has been laid down already, the audio/video would "scratch", or relocate, to the line's position and commence playing from that spot. If he lays down all the lines of the poem, he can access any line at any moment. (Figure 24)

Finally, after the user lays down however many lines he wishes, he can click on the replay button. The video then plays in the order that the lines appear from the top of the screen to the bottom. The result is a re-sequencing of the entire piece.

DISCUSSION

Scratch is more successful than *Flash*, though not ultimately successful in the sense of being poetically self-sufficient piece. The access to "Whitman in Black" as originally written helps enormously in understanding the entire piece, and the complex behavior connected to direct control of the audio/video stream moves the user from being a passive observer of the spoken version of the poem to being an active participant in its flow. On the other hand, the opportunity for this interaction are not made clear to the user. Once they are, however, the people who used it seemed quite fascinated by the ability to scratch with the words and reassemble the piece.

Several elements of *Scratch's* interactive abilities are not obvious. Why move the mouse up or right to spool *forward* through the lines of the poem (while over the video clip)? Why does the cursor spit out different-sized type, and what is the interactive meaning of the size change? Why does the vertical relationship between the lines' orientation effect the order of re-sequencing but their horizontal relationship does not? Why is it not possible to perform a natural extension of the resequencing function by rearranging the lines once they have been laid on the page?

The scratching reaction powerfully involved users in the structure and sound of the poem, and is a device which deserves further refinement. Such refinement, though, must be in the context of a poem in which that sort of deconstruction and (re)assemblage of lines is integrated into its thrust. "Whitman in Black" is, if anything, about a man–Walt Whitman–at one with his New York environment. The scratching interferes with this sense to the point of actively working against it. Resequencing has similar unwanted poetic consequences, breaking down the ordering of lines which were crafted carefully into a specified position within the poem. It is either with a more sympathetic type of poetry, such as Burrough's cut-ups or various forms of found poetry, or within the context of a poem created from the first moment with such a component in mind that scratching will prove its usefulness.

5.2 **Concrete Poetry Redone**

As discussed in Chapter 3, a significant influence on the entire Dynamic Poetry project is the work of the Concrete Poets, and by heritage, the Futurists and Dada. Both the spirit of revolution and artistic timeliness invoked in these movements, and the dedication to equalizing the visual and linguistic dimensions provide inspiration for *WordNozzle*. The starting point for this project was when I found myself increasingly frustrated by the cumbersome methods for experimenting with typography available in most commercial graphics programs. In such software, the user must often type in the words, highlight them, pull one menu to select the font, another to select the size, yet another to select the style, and yet a fourth to select a color. This is a slow and enervating process, and not conducive to rapid, free-from experimentation with type.

Simultaneously, I was researching the Concrete poets. The techniques which they employed were yet more time-consuming and intolerant of revision. Realizing this not only granted some perspective on the current tools and they might be improved, but also engendered a curiosity about what sort of digital environment would be appealing to a Concrete poet. *WordNozzle* represents an attempt to improve the tools and create such an environment.

5.2.1 ***WordNozzle***

WordNozzle is equal parts digital graffiti and concrete poetry, an experiment in "painting" with letters, words and paragraphs. It was designed to let the user spit out characters, words, paragraphs, entire manifestos, even, in a continuous stream, while also allowing him continuous control over the appearance. Two main versions of *WordNozzle*

exit, one suitable for desktop use and distributed as shareware via floppy or on-line download, and the other as an installation piece presented in 1996 at the Royal College of Art.

Both versions allow the user to manipulate standard parameters such as font, size and color. The desktop version possesses more extensive file input-output capabilities, including saving the canvas and the ability to up-load text files into the nozzle. The installation piece replaced the standard desktop-plus-mouse environment with a large-screen-plus-firehose interface. Each version is described and discussed individually below.

5.2.2 ***WordNozzle –*** desktop version

[Dynamic Poetry Experiments:DP Direct Access:WordNozzle Desktop]

DESCRIPTION

Upon beginning, the user is shown a title screen while the program loads the fonts particular to the user's computer. The program then fades onto the main screen. (Figure 25) As soon as the user clicks anywhere, he is prompted to choose a text as the source for the nozzle. This text, or more appropriately, text-stream, is "injected" into the hose and the first word appears at the nozzle. At this point, the writer can change the appearance of the word by using the characteristics palette to the left of the writing surface.

By moving the word over the area marked "font" and letting it "hover" there, he can cycle through all the fonts available on the system, applying in turn each font to the word. To select a particular font, he removes the word from that area. By letting the word hover over the area marked "size", he can increase the point-size by moving slightly above "size" and decrease it by moving slightly below "size". Selection occurs in the same way as with choosing a font, i.e., by moving out of the "size" area. To change the greyscale shade of the current word, he moves it over the greyscale spectrum at the top of the screen, pulling down and out of the spectrum when the word exhibits the desired shade.

File input-output is handled in the lower part of the left-hand side. Positioning the current word over "wipe canvas" and then clicking clears the screen. The current position in the text-stream is maintained in this action. Doing the same over "load text" allows the user to select another text document as the text-stream. Clicking on "save image" lets the user name and save the canvas as a PICT file. This is an image-file format common to most graphic and word-processing applications on the Macintosh and allows the user to use what he has created as a graphic element.

DISCUSSION

The basic functionality of WordNozzle – manipulating the font, size, color and location of type – can be found in most desktop publishing and graphics packages. Yet the ability to treat the text-stream more like an ink or paint flow, while maintaining control over discrete words, circumvents the cumbersome, modal manipulation and lay-out methods common to such packages. This flow successfully takes the focus off the tools and their manipulation and places it on the words themselves.

The visual quality of the interface works well in creating a seamless space within which both the writing surface, the parameter controls and the design can all exist. The parameter controls dissolve into the grain of the background, the disappearance of the cursor removes the usual windows-based intermediary between the user and the object being used, and the grainy, layered, off-kilter typography embedded in the background encourages experimentation.[5]

WordNozzle successfully handles both the issue of creating a flexible experimental tool and providing an environment in which one can create new forms of Concrete Poetry. In the latter context the piece raises the interesting issue of composition and authorial control.

As designed, *WordNozzle* requires two distinct authorial stages. The first is in the composition of text-streams to feed the nozzle. At this stage the author chooses the basic ingredients of a poem–words–but does not necessarily have control over their final sequencing or relative layout. Even though they are sprayed from the nozzle in a linear order, the compositor's control over location gives him the final say in how the poem is actually read. Similarly, the compositor does not necessarily control the ingredients with which he is working. In some cases, the author and the compositor are one and the same, and *WordNozzle* becomes akin to a performative medium. The author uses it as a means of displaying his words; he can choose to do this in the same way every time, or vary this aspect at each performance. The situation is analogous to a poet who reads his poetry for an audience and changes the reading each time he does so.

---

[5] Even though not accessible in the present interface, the *WordNozzle* software engine is capable of handling a spectrum of flow. That is to say, it is possible to switch between a linear text-flow, in which the words from the text-stream are accessed in their linear order; or random text-flow, in which words are picked at random from the text-stream; or random, exclusive text-flow, in which words are again picked randomly, but with no repeats. The software also supports switching between letter-sized, word-sized, sentence-sized, paragraph-sized and entire text-sized chunks.

However, in the case where the two roles are taken by different people, a situation occurs which is much more similar to a mixing d.j.. who utilizes samples of music already composed by others. Such an artist works at a level one-stepped removed from that of the musicians, but is capable of creating new and meaningful compositions out of the music he finds. The compositor is in a similar position, to the point that the text-streams he mixes do not even have to be composed specifically for this purpose. *WordNozzle* does not care whether the text-stream is composed of verse or the READ ME file from the compositor's word processing program, or the contents of an electronic mail box, or any other text document.

One of the several causes for the lack of an audience interested in the postmodern conceit of non-linear narratives and works which are continually open to authorial input from a multitude of voices is the desire for a clear voice or point-of-view which leads the reader through a carefully constructed linguistic edifice. *WordNozzle's* compositing stage runs the danger of succumbing to the same weakness. When I go searching for a poem by Walt Whitman, I want the poem written by Walt Whitman, not somebody's "remixing" of Walt Whitman. But there are times which I am actually interested in something which takes Whitman's words and builds upon them, amplifies them or distorts them in a way that may shed light on the original, or the subject matter of the original or some completely different matter.

Yet there is still the uncomfortable feeling that I am doing violence to Whitman's text by re-using it in such a matter. Beyond that discomfort is the critical issue of whether his text is the type of text best suited to such treatment, i.e., whether the particular content embodied in the words and Whitman's poetic structuring of the words is amiable to the re-forming which *WordNozzle* supports and encourages. For these reasons I see the writing of new text-streams to feed *WordNozzle* as of paramount importance, for they can be created specifically with *WordNozzle* in mind. Users–and readers–will always be free to locate authorship where they will, but if they do not have to struggle against the ghost of an author who has created a tight structure in service of a specific range of meaning, that freedom will only be enlarged.

5.2.3 ***WordNozzle –*** installation version [Dynamic Poetry Experiments:DP Direct Access:WordNozzle Installation]

DESCRIPTION

The most significant difference between the installation version and the desktop version is that the installation expands the software into a body-sized version with an input device

much more in keeping with the piece as a whole. The personal computer monitor of the previous version is transformed into a rear-projected large-screen measuring 1m x .75 m. The mouse is replaced by a fire-hose and nozzle procured from the supplier to the London Fire Brigade. (Figure 26)[6] The user controls the movement of the words by pointing the nozzle; he controls the flow of text by using the top mounted handle on the nozzle as a valve. (Figure 27) Pulling back on the handle increases the flow of text, while pushing forward decreases it to the point where the flow eventually stops and the user can simply moving a single word around the screen. (Figure 28)

The screen itself looks very similar to that for the desktop version, with slight modifications to the left-side input-output area. As this is designed as a public installation with short, 5-minute interactions in mind, there was no need – and no way to facilitate downloading files for users to take with them. Thus the "save image" functionality has been removed. In order to simplify the interaction for the short time-frame, all the control interaction consists of hovering, i.e., like with the "font" and "size" controls, the user hovers over "wipe canvas" and "load text" to activate them (instead of clicking on them as with the desktop version.) If he hovers over "wipe canvas" for more than a second, it activates. If he hovers over "load text" for more than a second, then a window pops up in the middle of the canvas. Within the window, the names of the files available for uploading into the nozzle appear in succession. To select one, the user simply moves off of "load text" and the text whose title was the last to show is selected.

DISCUSSION

I transformed *WordNozzle* into this form for several reasons. I wished to explore its performative aspects, to see how satisfying it is for users to work with words which are selected for them. I wished to investigate the marriage of whole-body based input devices with the screen-based virtuality, and in so doing extend my understanding of the mechanics and electronics required to pull control away from the keyboard and mouse. Finally, I wanted to see if very large text had the emotional impact and pull that I suspected it had. Generally, user interaction tended to take three phases. The first phase consisted of the user accustoming himself to the notion that the fire-hose was functional and to the slight reaction lag which using it entailed. The next phase consisted of him exploring the interface – changing font and size, selecting colors, wiping the canvas, etc. The final phase consisted of him actively playing with the piece and the feel

[6] The photos depict the color version, which I initially installed for the opening night. Afterwards, however, I re-installed the standard greyscale version.

of the interaction. All of this happened relatively quickly (though in some cases people did get stuck at the first or second stage.) Judging from the number of people who interacted with the installation and their willingness to learn the peculiarities of the interface, the piece was successful as a means of letting people "perform" with a given set of text-streams. Furthermore, users were intrigued by the use of the nozzle and the connection between its physicality and the virtuality on the screen.

On the large-screen, the words were quite powerful. This could be seen in the way that companions of the nozzle-holder would often urge them on, to see what was next to come out of the nozzle, to see what the results of large-scale adjustments in size would be, etc. In addition, the scale brought forth the richness in texture obtained in the simple layering of words of different greyscale. It encouraged me to think about recasting the piece on an even large scale, say that of an entire wall.

5.3 **A Digital Poetry**

With *Flash* and *Scratch*, I took poetry created by others for a print environment and repurposed it for the digital environment. With *WordNozzle*, I used poetry written by myself, originally for a print environment as well. In all three cases, the fragmentation caused by these new ways of presentation failed to do the poems themselves justice. The need to write poetry which, from the first creative moment, is destined for a digital environment had been made increasingly clear.

Breeders represents the first move in that direction, and the rest of the poems described in this section represent the further working out of issues involved in creating poetry specifically for and within a digital environment. In *Breeders* I designed the basic software engine to handle the chunks of text and then improved it with each subsequent piece. This engine makes use of object-oriented programming (OOP) practices to handle the complexity of action within the environment, and led me to adopt a behavioral approach to creating the interaction within the poems. Thus, individual words are treated as individual entities with a certain kind of "life" in the environment which is determined by the behavior which I assign to them. Once they come into being, I no longer need to worry about controlling their actions–they "know" how to act. In effect, I set the starting conditions and yell "Go!" The words-as-objects take care of themselves from then on.

In order to craft these environments, I developed a process for approaching the simultaneous demands of interactive form and poetic content. Though I have embodied the process in four steps, the reader should be aware that they often do not follow one another in a clean progression but reflect and modify each other at all times.

1. *What behaviors are interesting in the context?* Given the thematic arc of the poem, what should the text do? Which parts should be dynamic, which parts should be responsive, and in what ways?

2. *What metaphor best describes the behavior?* If chunks of text are going to change capability and appearance over time, should we think of that as "aging" (organic process on animate subjects)? Or "weathering" (organic process on inanimate objects)? Or "fatiguing" (in/organic process on both)?

3. *What visual appearance and changes in visual appearance correspond to a particular behavior?* If a chunk of text "ages", how does the user see that? how much of its history (and perhaps its future) should be made apparent?

4. *How does one embody such qualitative behaviors in code?* It is one thing to talk about "weathering"; it is yet another to find away to do that in computer code.

At different moments during the process, a different one of these questions would come to the foreground and demand attention. For instance, I initially spent substantial time in the first question, brainstorming about what I might want text to do. As part of that, I was constantly considering question two, brainstorming metaphors to keep myself clear on what I was thinking. For example, when I decided to implement the narrative-place behaviour in *Breeder: No Discussion,* I moved on to the third question. As I became more familiar with the capabilities of Director's implementation of OOP, I returned to the first and second categories to update my thoughts on the behaviors. All the while, I was thinking about the visual appearance but delayed expanding on those details until the piece was working well enough to see what the simple dynamics of the environment were. At that point, the third question came heavily to the fore and was one of the catalysts for collaborating with Neil Wilson on *Articulate*.

It was also necessary to develop a vocabulary which would succinctly capture the behavior of these texts. In some cases, I used descriptive terms from traditional poetics; in others, new terms were employed. Below is a brief glossary:

Chunk Descriptors – describe discrete entities within the environment

line: an ordered set of words

stanza: any collection of lines which make up a discreet, though not independent, unit of meaning; the lines which compose a stanza may be sequentially or randomly displayed.

anchor: a stanza or set of stanzas always present on the screen.

in-betweens: lines which appear interleaved between other lines, where the in-betweens are of a significantly smaller type size than the dominant lines.

text-pouch: any text object that serves as a source or is accessed by these poems.

Dynamic Behavior – describe the actions which chunks can take

march: when the words of a line or stanza appear one at a time, one after another.

stream: quick movement of a line or stanza across the screen in an uneven, though directed, manner.

glide: slow movement of a line or stanza in an even, directed manner.

scramble: a line or stanza which moves about the screen in a highly random fashion

gather: when several lines or stanzas flock to a common location; that location may be spatially fixed or moving.

vector: the direction and speed of movement possessed by an object.

The reader will find it helpful to refer to this list while reading through the project reviews which follow.

5.3.1 ***Breeder***

[Dynamic Poetry Experiments:DP Direct Access:Breeder]

DESCRIPTION

The goal of *Breeder* was to create a whole canvas composed of pieces of text interacting with each other, and with the user, in different ways and in an ongoing manner. It attempts to approach the intersection of text and interactivity through an overarching biological metaphor. Employing such a metaphor to bridge the gap between the

inorganic determinacy of silicon and the phenomenological vagaries of the flesh is nothing new; the world of computers has seen "genetic breeding" of algorithms, of visual stimuli (like Kai's Power Tools' *Convolver*), and of intelligent software agents. The main ideas that I wish to pull from the metaphor are those of *behavior* (the actions exhibited by an entity in response to stimuli) and *inheritance* (the transmission of behavior to new entities.) Several prototypes using the basic *Breeder* engine were made; I have included *Breeder: No Discussion* on the CD-ROM.

Breeder: *Glide*. The first iteration focused on creating a basic environment. Working with a text-pouch created from three poems I wrote as a series a few years ago, I wanted to dynamically juxtapose individual words. On initialization, *Glide* randomly samples the text-pouch, chooses words to seed the environment, places them at a randomly chosen edge of the bounding box and then glides each word into the environment. Each word is exists as an object which knows its text-content, its speed, its direction and its age. When a word falls outside the boundaries of an invisible box, the it and the object associated with it are forgotten and a new one is created. Upon creation, it randomly samples the text-pouch,

Breeder: *No Discussion*. The second iteration attempted to implement what I have chosen to call *narrative-place* behavior. Narrative-place describes a word which "knows" that it is part of a larger piece, knows its location within that piece and searches for the other entities that come before and after it in that piece. When it finds them, it combines with them to form longer and longer phrases until the entire piece is reconstructed. The text used in *No Discussion* is a fragment that runs "the between popping with a loud, empty bang of disconnection, no direction."

In *No Discussion*, as any particular word is sampled from the (linearly ordered) text-pouch, the word that immediately precedes it and the word that immediately follows it are recorded as part of its knowledge. As a word-object glides across the environment, it inspects all the other words with which it comes into contact. If the contact-word is one of its neighbors, the word-object grabs its contents to create a phrase. If the contact-word precedes the word-object, the subsequent phrase regards the previous word of the contact-word as its new previous word. If the contact-word follows the word-object, the subsequent phrase regards the following word of the contact-word as its new following word.

As time passes, the previous and following word of each original word and subsequently built-up phrase migrates further and further away, until eventually the full text is reassembled. The number of words which would complete this progression is dependent on the course of the random sampling function and the randomly chosen course of words through the environment.

To help speed things along if the environment becomes saturated with incomplete phrases which cannot find any additional points of linkage, the environment is seeded from time to time with random words.

Breeder: *Glass* The third iteration experiments, briefly, with putting sound to the environment. Fellow Computer Related Design student Lukas Girling pointed out that the rhythms of the text movement reminded him of music from Philip Glass' *Einstein on the Beach*. After listening to the opera, I agreed with Mr. Girling. The rhythm created by the combination of steady movement and random speed and direction contained distinct echoes of Glass' play between almost droning repetition and rhythmic variance. I combined the two, culling two samples from the opera which were then continuously looped as the environment evolved. The visual and aural dimensions complement each other, pointing the way to more complex interactions between the text, voice and music.

DISCUSSION

A quick technical note: the software-hardware combination I used to implement these pieces is not capable of simultaneously handling more than a dozen moving chunks with any degree of visual elegance. As more chunks are added above that level, the movement slows down to the point that it is not only unappealing but uninteresting as well. For this reason, *Breeder* only operates on a portion of a text-pouch. The piece that is on the CD-ROM serves as a miniature study of what an entire, complete piece would be like.

Glide is akin to found poetry or William S. Burroughs cut-ups. The sense of the piece lies not in the denotations of the words themselves but in the connotation of their constantly shifting amalgamation and the efforts of the user to synthesize this. The duty of the poet in this case is to provide raw material which is highly suggestive and to choose the range of visual realization which these words possess to aid in that. *Glide* does not follow one particular thematic arc, but rather many, and with a visual dynamism which keeps the user engaged.

No Discussion possesses more of a direction, though one which is necessarily kept brief because of the technical constraints discussed above. The poem deals with a

family in a continual process of breaking apart and coming back together, and the difficulty discussing the reasons for this happening. The dynamics of the poem work in tension with the movement direction of the words, which capture a feeling of not being able to connect, not being able to reassemble the familial cohesiveness: the words assemble themselves in a way that the family cannot. The contrast is pointed and poignant.

Glass served to show the possibilities of adding sound to a piece. As such, it was successful but does not warrant further discussion in and of itself.

The biological metaphor used in the *Breeder* series proved to be a powerful shorthand for conceptualizing text-based interactivity. I quickly came to the conclusion that I wanted the text to have personality, down to the word level at least and perhaps to the character level if possible. That personality could be a literal reflection of the denotative content of the word(s) in question, or it could be connected to the connotative suggestion, or it could be in some ways at odd with these senses. It interested me in pushing the grammatical control over words we constantly have to exercise *down* into the physical representation of the word itself.

Some of the other behaviors which are inspired by this work and which warrant further exploration are:

semantic evolution: if/how the character-content of the word changed over time.

lifespan: if/how the word "weathered" subsequent viewing/interactions.

spatial stability: if/how the word migrated in two dimensions.

pathology: if/how the word interacted with other words, either with hostility, neutrality or affection.

knowledge acquisition: if/how a text chunk increases what it knows about itself (etymology, other instantiations, etc.) and about its environment (what other text chunks coexist with it, what other "species" there are and how to talk to them...) A text-chunk could search a network for occurrences of itself, collecting pointers as it goes. As it collects them, it visibly swells. When it's poked – either by another chunk of text or by the user – it can disgorge that information, getting thinner as it does so.

hierarchy: if/how some text chunks would be privileged over others. Privilege could take the form of network access, or what sorts of other info-bits it could communicate with, or what its capacity is,

> or etymological age, or frequency of use...This behavior would be one of those most interesting to implement within the breeding metaphor, as an offspring could inherent knowledge, but not privilege, or privilege, but not knowledge. Or both, if we want to create some spoiled-rotten text.

After these initial prototypes had been completed, I spent time with fellow Computer Related Design student Neil Wilson, who has a background in typography, conceptualizing a font that would support the unorthodox behavior I was imagining. The result, *Articulate*, is a based on single characters having an articulation that allows them to hook up with other characters to create words, and from there to create phrases, and from there to create sentences. They also have a texture which makes it possible to color them in various ways, ways which can be used to signal the presence of various characteristics. As connections are made, these colorings will change to reflect the combined characteristics of the new phrase. The combination of *Breeders'* non-visually mature but interactive prototype and *Articulate's* visually mature but non-interactive images illustrate what could be done given more development and a powerful computing environment.

5.3.2 ***Dying Lying Rotting***

[Dynamic Poetry Experiments:DP Direct Access:Dying Lying Rotting]

DESCRIPTION

Dying Lying Rotting consists of an introductory title sequence and a main screen. The main screen possesses two main elements. Three stanzas are positioned in the center of the screen, reading in the standard top-to-bottom, right-to-left format. These form the anchor of the poem. Each of the anchors, in a black, medium-sized font, has a rhyming line picked out in bold. In between each line of the stanzas can be seen, too small to legible but large enough to be recognized as type, more lines of text.

Around the anchors glide a phrase from each of four additional stanzas. Their movement, unlike the completely random movement found in *Breeder,* is axial, consisting of either purely vertical or horizontal travel. Each one moves from one side of the screen to the other, and then reappears at some other edge and makes the journey again at a slightly different speed. Each of these stanzas is set in a different font, and, as the user begins the piece, they each possess a hard, bright red color.

As time passes, one notices something peculiar about these gliding stanzas. If an individual glider does not encounter any of the other gliders, it dims. If it does come into

contact with another one, one of the two becomes slightly brighter while the other becomes slightly dimmer. Eventually, it becomes evident that in particular pairings, one phrase always becomes brighter while the other phrase always becomes dimmer. If the poem is left running for long enough, then three of the gliding stanzas will get to the point where they are so dim they're invisible, while one has increased to maximum brightness. (Figure 30)

The user can grab any one of these gliders and move it about the stage. When he drops it, more text is displayed. This new text remains fixed as the glider resumes its movement.

If the user grabs one of the anchor stanzas, he finds that it zooms in and out. If he zooms into it, then the small lines in the white space in-between the lines of the large text grow larger and larger, to the point where they can be read. The text of the anchor stanza also grows larger. If the reader zooms out the opposite occurs, to the point where the major anchor lines become so small that they themselves become unreadable. (Figure 31)

Finally, a set of controls is located on the bottom of the screen. "Restart" begins the poem again; "Age" increases the rate at which the gliding stanzas move and the rate at which they fade; "Save" saves the current state of the poem; "Pause" freezes all movement and halts the poem clock.

DISCUSSION

Dying Lying Rotting deals with a young man “talking” to his grandfather at his grandfather’s funeral. The anchor stanzas make it clear that the young man possesses a fair amount of resentment towards the older man; the reader finds out more about the nature and causes of that resentment through interacting with the gliders and zooming into the anchors.

As more is read, it becomes clear that the conflict between the two was based just as much in straightforward generational differences as personality differences. The younger man feels that the older man has strangled the family for years, insisting on imposing his traditional views on everybody. The older man’s deflection of the younger man’s enthusiasm for change is summed up in the line “you are nothing more than your years, and even they, in comparison to mine, are nothing.”

The behavior of the gliders echo this conflict. These stanzas each have an age associated with it, which is based on the chronological order of the events they represent. So, for instance, the gliding stanza “December 13, 1988” is the “youngest”, as it takes place earliest, and “open coffin” is the oldest, as it takes place at the latest point in time.

The younger stanzas grab "life" from the older ones whenever they encounter each other, keeping them youthful (and bright) and making the older ones yet older (and dim.)

As a first attempt at creating a fully realized piece based on the *Breeder* paradigm, *Dying Lying Rotting* succeeds in some ways and does not in others. The two-fold aging dynamic that exists between the gliding stanzas, where they all get older in general yet the ones which are younger can steal time from those which are older, subtly supports the themes of aging, dying and the rise of youth to replace that which has died. The stillness of the anchor stanzas in contrast to the constant movement of the gliding stanzas reflect as well the frenetic-ness of youth compared to the stillness of great age. The user's moving of the gliders and zooming of the anchors serves to draw him into the poem and to encourage him to look "behind" what is immediately there.

On reflection, the interaction between the gliding stanzas should be made more visually explicit. I considered having the letterforms decay with age, but not only did this quickly render them difficult to read, but the visual effect was actually too strong, suggesting that to age is to decay. The sense I wanted, in which a young man attempts to thwart and drain away the power of the older man, required something less drastic. Nonetheless, the fading behavior employed, though more subtle, opens up an interpretive space which is too wide.

The zooming of the in-betweens in the anchor stanzas, while an interaction which seemed to please readers, ultimately failed to fit into the thematic content of the piece. I have retained them in this latest version in order to create a contrast between a dynamic movement that does work (albeit with flaws) as an integral part of the poem, e.g., the aging, and one that does not. The piece may benefit by being separated into two pieces, one of which deals with the anger and utilizes the anchors and the other of which deals with the inter-generational exchange and utilizes the gliding stanzas.

5.3.3 ***Telecommunication***

[Dynamic Poetry Experiments:DP Direct Access:Telecommunication]

DESCRIPTION

After moving through the title screen and entering *Telecommunication*, the user is presented with an empty black screen. After a moment, the first stanza streams on-screen from the right, traveling quickly. In fairly short order, another seven stanzas appear, one after another, all streaming from right-to-left. As they stream across the screen, they also march through their lines. All based on the same font, each stanza can be distinguished by variations in gray-scale and slight degrees of letterform deconstruction. (Figure 32)

After all eight of the stanzas have made their entrance, "SILENCE", the sole anchor of this piece, written in large type and colored blood-red, fades completely into view. It moves slowly across the screen, much more slowly then the stanzas, and does not march.

If a stanza comes into proximity to the anchor, the rate of its march decelerates. The closer the stanza gets, the slower the march gets. If one of these actually intersects "SILENCE", its march slows to the point of one word per two seconds.

After a period of approximately five minutes, the stanzas repeat the opening movement, but in reverse. One by one, they stop streaming and marching. "SILENCE" continues its movement, until, finally, the reader is left with the black background with only the anchor moving, slowly, on it. This will continue until the reader quits the poem.

DISCUSSION

Telecommunication deals with the how the speed and responsiveness of communication affects a friendship. The anchor acts like a black-hole, attracting the streaming stanzas to it whenever they get within a certain range. And, like a black-hole, time "slows" down and these streams move and march more slowly the closer they get to the anchor. This singularity-effect is intended to echo the arc of the poem, which concerns a friendship, once co-present, now long-distance, which is slowly dissolving into the silence of distance.

The individual stanzas are snatches of the correspondence between these two people, one male and one female, as it progresses from face-to-face to telephone to email and, finally, to letter, as the geographic distance between them widens with time. The slowing-down of the stanzas, both speed- and march-wise, reflects the way in which these people, in the lulls between (and, finally, the absence of) communicative acts, replay previous conversations in their heads – an act of remembrance as well as an act of prognostication, as each attempts to figure out what is going wrong. In the silence, all they have are what has been said, and they linger over those words.

The global temporal arc reflects the diminishment of communication. The piece begins with a flurry of conversation, then SILENCE enters, and, slowly, the quantity of that communication diminishes, until it ceases together, leaving the reader in visual silence.

5.3.4 ***Cross Purposes***

[Dynamic Poetry Experiments:DP Direct Access:Cross Purposes]

DESCRIPTION

Cross Purposes presents the user with a grid of text, with lines reading from right to left crossing, at right angles, lines reading top to bottom by character. Each line is a stanza. If the user wipes the cursor across the screen quickly, small red dots will appear adjacent to each stanza and then disappear. If the user settles on a particular stanza, the red dot appears and remains. If the user then moves the cursor along the dominant axis of that stanza, it moves in that direction and an arrow appears in place of the dot, pointing appropriately. The further away the cursor gets from the arrow, the faster the stanza moves. Reversing direction slows the movement until, on returning to the arrow, movement stops and the arrow turns back into a dot. The cursor has returned to its point of contact with the stanza, and thus its relative origin. Moving the cursor in the opposite direction moves the stanza in the opposite direction, with the same acceleration/ deceleration relative to the point of intersection. (Figure 33)

In this way, the poem is read. The user shuffles the stanza backward and forward, downward and upward, sometimes singly, sometimes in axial pairs.

DISCUSSION

Like *Telecommunication*, *Cross Purposes* is a poem with two voices. But where the other poem is a 'conversation' between two people, and deals with the break-down of that conversation, this one is much more akin to a double monologue in which the individuals describe the selfsame events from very different perspectives.

The voices are distinguished from one another in traditional ways – tone, diction, vocabulary – as well as by the strong contrast in the rectilinear visual presence of the poem. One voice exists in the vertically-oriented text, while the other exists in the horizontally-oriented text. Furthermore, the font for each is slightly different (two sans serif fonts, *Geometric* and *Arbitrary*.) Each stanza in one voice has a twin stanza in the other voice, both of them talking about the same event or idea, but in very different ways. The difference in perspectives is presented as a very strong visual conflict between the vertical and the horizontal.

This representation creates a problem, however. Because of training which teaches us to read in a horizontal manner first and in a vertical manner second, the horizontal voice is privileged over the vertical movement. As asking the user to move the lines around is akin to asking him to get involved in the exchange, the layout of those lines is tantamount to asking him to “listen” to one voice more than another. Though this was not the intent of this poem, one can see how it might be used to strong effect in a different sort of piece.

5.4 **Beyond the Word**

The following two pieces, *Aura* and *Life is Bait*, do not fit strictly within the purview of Dynamic Poetry, yet they both exemplify different aspects of digital media which this thesis has sought to explicate.

Aura represents an extreme example of "digital art", a piece which makes no reference to its reliance on the computer-as-machine. The complex real-time interaction between the user, its sensors and its display is only possible within the digital medium, yet my collaborators and I had no desire to draw attention to this fact. Instead, we aimed to create a piece which possessed an atmosphere and "weight" of interaction distinctly at odds with the clunky and complicated interaction often found in digital work. Furthermore, we removed all the input-devices we have come to associated with the computer – keyboard, mouse, cathode ray tube – and which narrow the point of physical interaction down to the fingertips and the eye and replaced them with the user's entire body. Instead of a hunt for manipulative leverage, the interaction was meant to be one of reverent encounter with an interesting presence.

Life is Bait is a showcase for both formal and content explorations. Content-wise, my collaborator and I were interested in exploring how well the medium could encourage exploration of a complex, highly emotional socio-political issue. Formally, we wanted to deconstruct the screen environment, fragmenting the singular, television-like presentation and desktop-lockout common to multimedia in particular. Combined, these goals led us to a design which supported an age-old debate format – pro versus con, with moderator – within an innovative aesthetic format.

Taken together, these two pieces – even if they do not revolve around poetry – explore parts of the digital media terrain similar to the other pieces discussed in this thesis.

5.4.1 ***Aura***

[Dynamic Poetry Experiments:DP Direct Access:Aura]

Aura represents a collaboration between Elaine Brechin, Robert Strong and myself, all Computer Related Design students. The piece was originally created for the *Rituality* CRD first-year show, and was then re-designed on commission from the London-based arts sponsor, ArtAngel, for its *Self-Storage* installation exhibit. I discuss the latter incarnation below.

DESCRIPTION

Aura consists of a small platform on which rests a plinth. At the top of the plinth is a stylized match in a small cradle straddling a striking block. Hanging above the plinth is a piece of translucent paper upon which a video image of a burning candle is rear-projected. From the user's perspective, *Aura's* main visual presence is that of a burning candle floating in space, surrounded by a simple alter a few steps high.

Upon approaching the candle, the user can see his body movements subtly effecting the flame, as if the wind of his passing was blowing towards it. Once on top of the platform and directly in front of the candle, he can blow on it and thereby extinguish it. If he then takes up the match and strikes it on the block, the candle re-lights. (Figure 34)

Aura was located in the lowest, farthest corner from the entrance to the *Self-Storage* exhibit. This distance, and its relative isolation from the other installations, gave the approach to it a pilgrimage-type effect. One at a time, visitors were let into a very tall, narrow corridor which was dark except for the red glow. When they came to the end of this corridor, they could turn right. They then saw another tall, thin corridor, except that this one was much longer than the original one and had huge tubing running down the left-hand side. At the end of this corridor, they could see the candle flickering. As they got closer, they could see that it "sat" atop an alter that could be reached by climbing four short steps. Once up there, they could interact with it as described above. (Figure 35)

DISCUSSION

Aura plays with the tensions between a computer's complexity and a candle's simplicity, between the materiality of the machine and the ephemerality of fire and light, between the developing rituals of silicon culture and the aging ceremonies of previous times. As a whole, the piece suggests an altar or some other place of focused movement, reverence and supplication.

Aura was designed with several goals in mind. First and foremost my colleagues and I wished to provide two types of interactions simultaneously. One is the focused, intentional interaction of traditional human computer interfaces. You have a goal, blowing out or lighting the candle, and the interface gives you the capability to achieve that goal and visually suggests how to go about it. The other type of interaction is one is not often embodied in the digital environment, a result of the machinery's will to complexity and the binary coldness of numbers. This is the casual, unmeant interaction one gets with delicate objects in the natural world–the slight movement of leaves as you hurry passed, the head-movement of a bird in a tree as it watches you walk underneath its

perch. Combining both of these forms of interacting allowed the piece to be both actor and object, and the observer/user to be both actor and audience.

Secondly we wanted to explore alternative means of input and output. Part of the original brief which spawned the piece was to create an interaction which in no way relied on the user utilizing a keyboard or mouse. We took it one step further by also removing the ubiquitous monitor as well. The user approaches *Aura* with the entire body, uses his breath to trigger the main event (blowing out the candle) and a very physical object to complete the cycle (the match on the striking block to relight the candle.)

Finally, we wished to design a piece which illustrated the concept of digital art as opposed to computer art. (Section 2.2 to defines this distinction.) Many of the people who interacted with the piece in the *Self-Storage* exhibit expressed surprise when it was pointed out to them that it was driven by a computer. Both the design of the environment and the feel of the interaction led them to postulate some other mechanism, from holograms to trick videos. We took that reaction as a sign that not only had we succeeded in removing an oppressive sense of "computerness" from the interaction but had also succeeded in our other goals as well.

5.4.2 ***Life is Bait***

[Dynamic Poetry Experiments:DP Direct Access:Life is Bait]

Life is Bait represents a collaboration between Paul Trevor, a London-based professional photographer, and myself. The piece was commissioned by the British Arts Council for its "Emotional Computing" series. Content requirements were to produce a piece which explored the ability of the medium to evoke an emotional response. Technical requirements were to deliver a program which fit within 1.4 Mb of space and was fully self-running, so that the Arts Council could easily distribute it on a floppy or over the Internet.

DESCRIPTION

Life is Bait opens with the transcript of a news broadcast summarizing the issues at hand in the McLibel trial taking place here in London, in which two activists are being sued by McDonald's for libel. The text scrolls from bottom-to-top as a very distorted voice reads out the contents. After the text finishes, two simple screens appear, one after another, providing some more background information and details about how to use the software. After the user has clicked through these, the screen then shrinks down to about 1/3 screen size, and goes through a series of child-like scripts, with people yelling out different

things that “Life is...” on the soundtrack, until, finally, it gets to “Life is... Bait.” Then the screen goes blank, and the user is launched into the core screen(s).

The heart of *Life is Bait* consists of four windows, the one large one and three more which are about 2/3 its size. (Figure 36) Initially, the smaller windows, labeled “McDonald’s”, “McLibel Two”, and “Neutral”, are empty. The large window, labeled “Bait”, has a multitude of visual elements moving around inside of it. When the user clicks on any of these elements, he gets a proxy of it attached to his cursor. If he then drags-and-drops it into one of the smaller window, a text-field appears in the smaller window. Dragging the same object into all three different windows produces three different texts, each of which represent the views expressed by that particular party to the action.

Finally, at the upper right-hand corner of the Bait window are two buttons. One, “quit”, is for obvious purpose. The other one, “Ref”, when pressed, calls up a window in which reference information – contact addresses, the text of documents enter into court, etc. – can be viewed.

DISCUSSION

The title, *Life is Bait*, is an attempt to capture the consumer culture environment, in which we are bombarded from all sides with temptations and lures to buy something. Often, this “bait “ is cast in terms that make the purchase appear beneficial to an individual’s life, and, furthermore, make the producer of the product out as deeply concerned for the individual’s well-being. Mr. Trevor and I feel that, more often than not, this concern is hypocritical and short-lived, lasting only long enough to extract money and, in the end, resulting in events that are actually detrimental to the health of the individual, the culture and the ecology.

As a visual metaphor of all this bait floating about, we created the main box as a “collage of temptation”. In the same way the consumer is tempted to buy and buy, we wanted to tempt the user to fully explore the issues we had to present. I had worked out the basic dynamics of the collage in two pervious pieces, making it relatively straightforward to insert the appropriate content once we had done the research.

Life is Bait served as a vehicle for commenting on two design issues which I feel contribute heavily to the current state of content-lag. The first is what I call the “Television Assumption.” Users and creators screen-based digital media tend to assume that the spatial and surface coverage constraints of television should be carried over to the environment simply because they both use screens. Thus, most new media fills up every available inch of a 640 x 480 dpi screen, regardless of content or interaction. But, new

media users can ignore the technological imperatives television was subject too, such as the need for a standard format for capture, transmission and reception. Furthermore, "filling the frame" is merely an accepted aesthetic norm, not a necessary one.

Yet the size of the windows, their number and shape should be related to the amount and kind of information being presented. This conviction led directly to us using multiple and smaller windows in *Life is Bait*. The number of digital objects we could use in the piece was severely limited by the 1.4 Mb storage limitation. This, along with the "bait" theme, led us to design the piece as a teaser, a small bite of the much larger issues. The size of the different windows in the piece is a direct result.

The other issue is what I term the "Narrative Assumption." New media users and creators commonly use a single space for presentation, even for the presentation of fairly complex and multifaceted issues. They combine with a need to hijack the screen and demand the user's undivided attention. Such methods are well-suited to the television screen and the film-theater, but ill-fit the much more flexible and multi-tasking medium of digital media. This conviction led us to separate each "voice" into its own window, encouraging the user to really see them as different points-of-view telling different stories. This also encouraged us to not hide the desktop, leaving our piece as just one among the many events happening in the computer space.

5.5 **Minor Reflections**

The text-based experiments described above show the large space which exists in which poets can explore new ways of writing. They also point the way towards particular forms which can be developed, certain ways of approaching text itself, and some of the general problems involved.

5.5.1 **New Poetic Forms**

Interesting transformations on classical forms of poetry can be achieved by using time instead of space to support repetition. For instance, instead of a villanelle with its use of repeated lines and strict numbering of lines, I can imagine a form which dictates that a particular word or phrase disappear and reappear at certain intervals, or move across the screen in a certain direction or of a certain speed. Or the overall time of a poem could be used in place of number of lines (such as in a haiku or sonnet) to define the form. A piece like *Telecommunication* could be the template of such a form.

A communal form could develop in which writer 1 begins with a blank writing surface and builds up an original poem. Writers 2, 3, 4,... n then take up the poem, either

sequentially or in parallel, and re-write it, subject to the constraint that they cannot add any more words to the text-pouch. All the writers' poems could exist simultaneously as layers of the same "meta-poem."

In general, the formal structure of beginnings, middles and endings is a prime arena of experimentation. The poems I have written here have beginnings, but no real middles or endings other than, in some cases, that determined by the clock. But I can imagine a piece with no beginning. Every time the reader boots it up, he confronts a slightly different screen. Yet the ending would always be the same. A parallel case can be made for a poem with only one middle, but countless beginnings and endings. The middle acts as a plateau in the interactive flow, like a caesura, to which the poem constantly returns. The timing of the cycle could be based on any of the three times discussed in Chapter 4.

5.5.2 **Punctuation**

Creating punctuation that works with the various dynamic styles developed so far has proven to be extremely difficult. In *Cross Purposes*, I chose to do without punctuation because the vertical orientation of half the stanzas tended to isolate periods, commas and question marks from the phrases they were meant to punctuate. Instead, I chose to use spacing to delineate phrases from each other and relied on the writing to make the points usually made by question marks.

With *WordNozzle*, similar problems surfaced. Since words come out singly and the composition of any phrase could be altered at any time by the visual layout of the piece, punctuation at first seemed superfluous. But while experimenting with the tool, it became clear that punctuation marks, by themselves, were quite powerful and useful to retain. Suddenly, a period no longer meant "this sentence ends here. stop." It simply meant "stop", regardless of the sentence to which it belonged originally. A comma no longer meant "this clause ends here, take a moment's pause." Like the period, it was shorn of its context and became a rather imperative "take a breath."

One avenue which I did not have time to explore fully is that of using time very precisely to replace rhythmic indicators such as periods and commas. For instance, in *Telecommunications*, instead of displaying these marks within the text, they could be used as invisible indicators which tell the program how long to pause the march at a particular word, or how quickly to march through a particular phrase.

Finally, the various pieces within the project show clearly that the horizontal, linear imperative of written text makes using punctuation in any other way either

confusing or comical. The creative vocabulary developed for dynamic poetry will have to include provisions for innovative forms of punctuation, whether it be the use of visual space or timing or some other method. As punctuation, like letterforms, relies on widespread convention, developing such forms which will be robust enough to survive into common usage will prove to be challenging.

Chapter 6 Conclusion

6.1 **Conclusion**

The growth of new media is an unruly phenomenon. This thesis has sought to show how an understanding of past media shifts involve *content-lag* as creators struggle to assimilate new affordances. One of the main factors effecting content-lag is the retention of familiar ways of seeing and making which cause participants to overlook fresh possibilities. In the present state of evolution in the digital medium, one particular area that has suffered from such a lack of attention is text.

This thesis drew on the experiments performed in the visual realm by pioneers such as the Futurists and the Concrete poets as inspiration for what can be done within a dynamic and interactive context. The experiments in digital poetry I conducted as part of this investigation form a body of suggestions about how to infuse digital text with the same energy and creativity already being expended on digital image and sound. This energy is needed if we are to progress towards pulling text in general and poetry in particular into the digital age.

By identifying content-lag as an analytical category and encouraging others to apply it to digital media, those who work within it can reflect on their work and the direction in which it is leading the media. Hopefully, this reflection will inspire them to develop robust expressive forms which are in some essential aspects *native* to the media.

Within the framework of content-lag, I argue that the concept of interactivity possesses a perverse combination of ubiquity and senselessness which interferes with the development of a rich understanding of what is possible in the digital environment. Maeda's reactivity and Campbell dialogue models serve as examples of how to re-think interactivity, and led me to extend their approaches. My model takes a dual perspective on the issue, where *dynamics* refers to the way in which the system sees the world and *response* refers to the way in which the user sees the system. Each perspective has several aspects. Dynamics can be thought of as *constructive, reactive, active* or *static*. Response can be thought of as *dependent, independent* or a hybrid of both.

In addition to dynamics and response, I re-evaluate the *time* dimension within the digital environment. I propose that time possesses three aspects: *cycle-time, real-time* and *interactive-time*. These terms describe the different speeds at which the user experiences his interaction with the computer-based piece.

Taken together, these approaches will hopefully catalyze other designers and artists working with digital media to strive for ways in which to transcend the present hollow obsession with interactivity. In the end, though content-lag is a not an avoidable phenomenon, we can work in ways which may shorten its duration. By choosing, at least some of the time, to forego the most obvious uses of the technology and push, hard, on the characteristics unique to digital media, we will find not only those points at which it breaks. Sometimes we will find those points at which it speaks like nothing has ever spoken before.

6.2 **Future Directions**

More work needs to be done in developing dynamic poetry into a viable form of artistic expression. One of the strengths of the digital medium – which I have not approached – is that it is agnostic towards form. Bits are bits, and whether they come out as sound, image, moving image or text is the choice of the artist/designer. Future directions to explore include integrating those elements into dynamic poems.

Many of the experiments undertaken in the current project exist in a two-dimensional screen space. By combining the mastery of the third-dimension represented by the work such as that done at the Visual Language Workshop with the novel applications of the Dynamic Poetry project, the writing space available to the poet becomes much larger and, in some ways, more complex. Giving the poetry spatial depth will certainly be a fruitful avenue of exploration.

Another dimension not presently investigated involves modulating the level at which the poet exerts control. The Dynamic Poetry project concentrates largely on the level of the phrase, and to a somewhat lesser extent the word. Yet letterforms themselves are ripe for a remaking into interactive objects, as are entire documents.

In a similar vein, the agent-like behavioral approach taken in *Breeders* warrants a more sustained and computationally intense treatment. A scheme of character- and word-based personality could be developed which supported a radical form of text-based dynamic cut-up and collage, and which would have benefits for more practically-oriented work on handling large scale information spaces. A Dynamic Text Markup Language (DTML) could serve as a meta-language for describing and reproducing textual behavior. In essence DTML could follow the lines of Standard Generalized Markup Language (SGML), Hypertext Markup Language (HTML) and Virtual Reality Markup Language (VRML), all of which have proven enormously powerful as protocols for creating and maintaining the visual appearance of data as it flows from platform to platform, network to network and medium to medium.

Finally, the application of additional poetic talent to the form will greatly advance it. Though I have made a sincere and partially successful attempt to write pieces which have all the power and grace of a skilled poet, my abilities in this area pale compared to some. The participation of others is crucial to creating a viable poetic genre within the new media. As Eisenstein might say - the world awaits its first digital poet!

Appendix A Technical Discussion

This appendix presents some of the interesting technical issues encountered in the course of the Dynamic Poetry project. As Computer Related Design is not about developing fundamentally new algorithms, "interesting" in this case means code and circuit design which can be modified and re-used by others working within a Director-like scripting environment and with a need for some basic structures for alternative input devices.

This section is necessarily of a technical nature. Code examples have been modified to make them more readable to the non-coder and circuit examples have been simplified. I have tried to make these illustrations as clear as possible but have not endeavored to provide a basic course in programming or circuit design.

Aura

Aura used a combination of household-grade motion detectors and simple contact switches to support user interaction. Both kinds of input were run through a keyboard whose keys had been modified by interposing an external switch. When one of these external switches was closed, the keyboard sent a signal to the computer that the key corresponding to that switch had been pressed. Director is designed to capture keypresses, so the software waited for certain keypresses to occur and took appropriate action.

Seven key-switches were used. Two were wired to the motion detectors so that whenever they detected movement instead of sounding an alarm they closed the circuit. The Director script mapped these two key-switches to two video segments which showed the burning candle wafting either to the right or the left. For example, when the user moved towards the left-hand side of the space, the motion detector on that side detected the movement. This resulted in the "L" key-switch being activated. This in turn resulted in Director swapping the current video segment out for the segment which showed the candle moving to the left.

Four key-switches were wired on one side to the thin antenna which hung down from the horizontal poles which supported the rear-projection paper. On the other side they were wired to the metal ringlets in each corner of the paper. In this way the wire not only suspended the paper, but when the paper was pushed gently backwards, the ringlets made contact with the antenna and closed the circuit. A user blowing on the paper would accomplish this. These key-switches were mapped to a segment of video which depicted

the burning candle being blown heavily and eventually going out. To add a slightly more realistic feel, approximately every 1 in 8 times it would map to a video segment which showed the candle being heavily blown and not quite going out.

The final key-switch was attached to a pressure switch hidden under the metal strip on the top of the striking block. When the user struck the match, it pushed down on this switch, closed the circuit and activated the key-switch. These action was mapped to a segment of the video which depicted the candle relighting.

In between action on the part of the user the program randomly switched between five different segments showing the candle burning.

WordNozzle

This discussion will be in two parts. First I will present three portions of code from the basic software engine. Then I will discuss the circuit designed to enable the nozzle in the installation version.

The software contains two segments of code which might be of interest to others working with Lingo. One series of handlers deals with uploading text files and storing them for random and linear access. The difficulties here were 1) storing the text in such a way as to provide very fast access to any point in the stream, and 2) finding a way around Director's in-built limit of 48 sprite channels in order to allow unlimited text flow.

```
-- TEXT LOADING AND SPRAYING HANDLERS
--
-- getTextStream prompts the user to choose a text file. If the user does
-- choose one, the contents are put into a global variable called gText.
-- All subseqent calls for text extract content from gText. Using a list
-- allows access which is fast than if stored within a field cast member
-- and far fast than if repeatedly accessed from the file itself.
on getTextStream
	-- gText : global variable which holds text
	-- newFileChosen : global variable which flags whether a new file has been
	-- chosen by the user
	global gText, newFileChosen

	-- FileIO is an extension of Lingo which handles file input/output operations.
		-- This call presents the user with a dialogue box asking him to choose
		-- a new text file. If a new file is chosen, a pointer to it is put into
		-- TextSourceObj, the file contents are put into gText and the flag
		-- newFileChosen is set to true.
	put FileIO(mnew, "?read", "TEXT") into TextSourceObj
	if objectP(TextSourceObj) then
	put SourceObj(mReadFile) into gText
	put 1 into newFileChosen
	else
	put 0 into newFileChosen
	end if
	if objectP(sourceObj) then SourceObj(mDispose)
end getTextStream
```

```
-- sprayCanvas calls sprayNozzle and primeNozzle to handle spraying a word on the
-- canvas
on sprayCanvas
	sprayNozzle
	primeNozzle
end sprayCanvas

-- sprayNozzle actually places the current word in the correct place
on sprayNozzle
-- NozzOffset : the number of the sprite channel currently being used
-- AnchorH, AnchorV : cooridnates of where the mouse was clicked
-- (the mouseLoc won't do as it may have moved slightly in the time
-- it has taken to process all of this.)
	global NozzOffset, AnchorH, AnchorV

	set the loch of sprite NozzOffset to AnchorH
	set the locv of sprite NozzOffset to AnchorV
	put nozzOffset + 1 into NozzOffset
	updateStage
 end sprayNozzle

-- primeNozzle handles advancing through the text-stream and setting
-- the appearance of the words. It does this after checking to see
-- if the amount of sprite channels reserved for words has been
-- exceeded. If so, takes a picture of the canvas, swaps that
-- picture into the background and resets the sprite counter to the
-- first available sprite channel. It then performs a recursive call
-- to actually prime the next word.
on primeNozzle
	-- NozzOffset : position within the text-stream gText.
	-- fontSize, gColor, gStyle, gFont: determine the visual characteristics
	-- of the current word
	-- gString : the current word
	-- gNextText : the next word in the text-stream
	global NozzOffset, fontSize, gColor, gStyle, gFont, NumTextChunks
	global gString,  gNextText

	-- skipNextText advances the pointer into the text-stream by one chunk
	-- and sets the global variables gString and gNextText
	skipToNext
	-- if there are sprite channels remaining, then set the new word at
	-- the end of the nozzle and adjust its appearance
	if NozzOffset <= NumTextChunks then
	put gString into cast NozzOffset
	set the textFont of cast NozzOffset to gFont
	set the textSize of field NozzOffset to fontSize
	set the textAlign of field NozzOffset to "left"
	set the foreColor of cast NozzOffset to gColor
	set the textStyle of cast NozzOffset to gStyle
	set the visible of sprite nozzOffset to true
		-- setTextWidth adjust the length of the text field to accomodate
		-- the changes in appearance
	setTextWidth(NozzOffset)
	updateStage
	else
		-- SavePage takes an image of the canvas and makes that image the new
		-- canvas
	SavePage
		-- 4 is the lowest sprite channel available for use
	put 4 into NozzOffset
		-- moveOffStage moves all the sprites from channel 5 above off of the
		-- screen
```

```
        moveOffStage((nozzOffset + 1), numTextChunks)
        updatestage
        PrimeNozzle
        end if
        put gNextText into field "nextTextField"
end primeNozzle
```

This second code segment is a routine built to circumvent a Lingo bug. If the length of the text in a text-field is increased during run-time or the point size of the text is increased by more than 10 points, Lingo will oftentimes not increase the size of the field. This results in words which are visibly truncated. Every time the appearance of a word is adjusted, the code below is called to make sure that it is correctly displayed.

```
-- setTextWidth sets the length of a text-field to one which fits the
-- word which is stored in whichCast and which is represented by
-- gString. This is required because Director often does not update
-- the text field correctly.
on setTextWidth whichCast
        global gString,gFont,fontsize,gStyle

        -- strWidth is an XCMD which takes takes the content, font, size and style of a
        -- string and returns the width in pixels of that word
        put strWidth(gString,gFont,fontsize,gStyle) into newwidth
        -- if strWidth is passed incorrect parameters it retuns ERROR
        if word 1 of newwidth = "Error" then
        put "ERROR at newwidth switch in ms4:utilities 3:"&&newwidth
         else
                -- an additional 4 pixels is added just to be sure
        put value(newwidth) + 4 into newwidth
        put the rect of cast whichCast into rectList
                -- this sets the third value of rectList to newwidth; the third value
                -- represents right horizontal coordinate
        setAt rectList, 3, newwidth
        set the rect of cast whichCast = rectList
                -- the registration point of text fields cannot be changed to anything
                -- other than the topleft; all the rest of this code is to
                -- find the proper offset so that the field appears in the
                -- correct place. Otherwise it appears shifted down and
                -- to the right
        put the loch of sprite whichCast into oregh
        put the locv of sprite whichCast into oregv

        put (the height of cast whichCast)/2 into halfheight
        put (the width of cast whichCast)/2 into halfwidth

        put oregh + halfwidth into centerh
        put oregv + halfheight into centerv

        put (the width of cast whichCast)/2 into newhalfwidth
        set the loch of sprite whichCast to (centerh - newhalfwidth)
        set the locv of sprite whichCast to (centerv - halfheight)

        end if
end setTextWidth
```

The third segment is a generic handler, *traverseList*, which deals with traversing a list. *WordNozzle* makes extensive use of lists to handle information about the fonts and styles available on the host system. The following code allows the programmer to move up or down any list by passing the list itself, the direction of movement desired, and a starting point. Lists are treated as non-terminating, i.e., when either end of the list is reached, the code 'wraps' around to the other end.

```
-- traverseList allows a programmer to move up or down a
-- list from any given starting point. It's chief utility
-- lies in its ability to treat the list as non-terminating,
-- i.e., it appears that it has no beginning or end. Could
-- be easily modified into a more general case in which
-- the caller can choose whether it is terminating or
-- non-terminating and the caller can choose how many
-- positions within the list he wants to move.
-- whichList : a list
-- whichDirection : "up" or "down". These terms are
-- relative and the programmer must keep track of what
-- they mean for his particular case.
-- startingPoint : the point in the list from which
-- direction is determined
on traverseList whichList, whichDirection, startingPoint

 -- going down list
      if whichDirection = "down" then
      -- check if at head of list; move to tail if true
      if (getPos(whichList, startingPoint)) = 1 then
      put getAt(whichList, count(whichList)) into newValue
   else
      -- move down list by 1
                  put getAt(whichList, ((getPos(whichList, startingPoint)) - 1)) into newValue
      end if

-- going up list
      else if whichDirection = "up" then
       -- check if at tail of list; move to head if true
      if startingPoint= getLast(whichList) then
      put getAt(whichList, 1) into newValue
   else
      -- move up list by 1
                  put getAt(whichList, ((getPos(whichList, startingPoint)) + 1)) into newValue
      end if

-- if whichDirectin is neither "up" nor "down" report an error
      else
      put "error with whichDirection switch in
                  traverseList:"&&whichDirection
      end if
      return newValue
end traverseList
```

The greatest challenge in creating the installation version lay in designing a system which would track the nozzle in two-dimensional space and not be effected by movement in the third dimension. Various ready-built systems exist on the market but all

were either too expensive or required hardware incompatible with the Macintosh. Michael Field of the Computer Related Design research staff volunteered to lead the design and implementation of a tracker which relied on infra-red sensing.

The electronics were designed around a programmable integrated circuit (PIC) which was programmed by Mr. Field and myself. Figure 37 shows a overview of the electronics. Figure 38 shows the circuit diagram.

For an overview of the program cycle, see Figure 39. Briefly, a high-powered infra-red transmitter is placed in the end of the nozzle. (Figure 40) It emits a continuous stream of pulses. Four infra-red receptors placed around the edge of the screen read and store the background light levels. The emitter is powered up and emits a pulse. The receptors are synchronized to take a "live" reading as this happens. The background levels are then subtracted from the live reading to remove any noise created by the ambient light in the room. The result is passed back to the Macintosh along a serial cable where it is used by the *WordNozzle* software.

Once received by the *WordNozzle* software, a series of mathematical manipulations are done to interpret the values received. (Figure 41) The difference between the two horizontal values will become the x value, while the difference between the two vertical values will become the y value. Normalization is performed first to compensate for scale and then again to translate the four values into two values. Next comes a geometry filter to compensate for the fall-off which occurs in the far-corners of the receptor matrix. Finally, the x and y values are translated from Cartesian space to Macintosh graphics world space, i.e., the origin is moved from 0,0 resting in the middle of the screen to 0,0 resting in the top-left.

Simultaneously a potentiometer set into the handle of the nozzle is also monitored. The further forward the handle, the more resistance possessed by the potentiometer and vice versa. These readings are then mapped such that if the handle was all the way forward and the current was near zero, no spraying occurred. As the handle is moved backwards, the resistance dropped, the current increased and this is mapped to higher and higher rates of spray.

Life is Bait

Life is Bait uses the core *Breeder* software engine to handle the display, movement and selection of images which appear in the main window. As that code is presented in the *Breeder* implementation section, I will focus here a group of handlers which were

developed specifically to support the multi-window, drag-and-drop environment of *Life is Bait*.

Though Director supports the opening of multiple movies at once, and thus the simultaneous existence of multiple windows, it requires that the programmer use the standard Macintosh window styles. Because we were interested in avoiding the static, television-framed style so common to windowing environments, we decided that it was unacceptable to use the standard window styles. This was possible as Director does allow the programmer to change the position of any window from within the code.

First attempts to use an update loop to track the user's mouse position as he dragged and redraw the window at each new position were not succesful because 1) window redraw is too slow and 2) system redraw was locked out, e.g., as the window is being redrawn the background of its previous location would not be redrawn. A visually inelegant and interactionally confusing method.

A custom XObject, dragRect, written by Jeff James for a project I did several years ago provided the kernal of a solution. This XObject, when triggered by a mousedown, will draw a marquee at any specified rect, move that rect with the mouse as it is being dragged, and return the coordinates at which the marquee was dropped. The majority of the following code is used to translate between the global coordinates used by the general windowing system and the local coordinates which were employed in order to make the whole environment independant of screen-size.

```
-- WINDOW DRAGGING HANDLERS
--
-- dragWindow emulates the MacOS's window moving functionality
-- within the Director environment. When the move spot is
-- grabbed, a marquee outline of the window is made. As the
-- user drags the mouse, the marquee moves along with it.
-- When the user drops the mouse the window is redrawn in
-- the new location
-- whichWindow : the name of the window being moved
on dragWindow whichWindow
	-- baitRect : global containing the rect of the
	-- main window. All calculations are done using
	-- the origin point of the main window point.
	-- This method is used to ensure that the
	-- calculations are screen-size indedpendent.
	global baitRect

	put the rect of window whichWindow into windowRect
	put baitRect into dragger
	-- calculate the offset of the window from the
	-- reference window.
	put stageOffset(dragger, windowRect) into dragger
	-- forces the cursor to the center of the drag icon
	put cursorCenter(dragger) into dragger
	put getAt(dragger,1) into lH
	put getAt(dragger,2) into lV
```

```
        put getAt(dragger,3) into rH
        put getAt(dragger,4) into rV
        put lh&","&lv&","&rh&","&rv into dRect
        put lH into oH
        put lV into oV
        -- dragRect is an XObject which handles the display
        -- and movement of the marquee. Returns the number
        -- of pixels moved in the horizontal and vertical
        -- directions.
        put dragRect(dRect, "noclipping") into locOffset
        -- calculates the new rect of the moved window from the
        -- amount of horizontal and vertical displacement
        -- undergone while in dragRect. Returns a new rect
        -- in 'local' terms, i.e., relevant to the main
        -- window.
        put localFromOffset(dragger, locOffset, oH, oV) into localRect
        -- local coordinates have to be converted into global
        -- coordinates, as this is the framework that Director
        -- understands when setting the rect of a window.
        put localToGlobalRect(localRect, windowRect) into globalRect
        set the rect of window whichWindow to globalRect

end dragWindow


on stageOffset babyRect, windowRect
  put getAt(babyRect, 1) into hDiff
  put getAt(babyRect, 2)  into vDiff

  put getAt(babyRect, 1) - hDiff into lH
  put getAt(babyRect, 2) - vDiff into lV
  put getAt(babyRect, 3) - hDiff into rH
  put getAt(babyRect, 4) - vDiff into rV

  set productRect = rect(lH, lV, rH, rV)
  return productRect

end stageOffset

on cursorCenter rectList
  put baitRect into outlineRect

  put 10 into dragoffH
  put 10 into dragoffV

  put getAt(rectList, 1) - dragoffH into lH
  put getAt(rectList, 2) - dragoffV into lV

  put getAt(rectList, 3) - dragoffH into rH
  put getAt(rectList, 4) - dragoffV into rV

  put getAt(rectList, 4) into rV

  set productRect = rect(lH, lV, rH, rV)
  return productRect
end cursorCenter


on localFromOffset whichRect, addition, oH, oV
  put getAt(whichRect, 1) into lH
  put getAt(whichRect, 2) into lV
  put getAt(whichRect, 3) into rH
```

```
  put getAt(whichRect, 4) into rV

  put rH - lH into widthH

  put  value(item 1 of addition) into cH
  put  value(item 2 of addition) into cV

  put cH - oH into mH
  put cV - oV into mV

  put (rH + mH) - dragoffH into rH
  put (rV + mV) - dragoffV into rV
  put (lH + mH) - dragoffH into lH
  put (lV + mV) - dragoffV into lV

  set productRect = rect(lH, lV, rH, rV)
  return productRect
end addToRect

on localToGlobalRect babyRect, parentRect
  put getAt(parentRect, 1) + getAt(babyRect, 1) into lH
  put getAt(parentRect, 2) + getAt(babyRect, 2) into lV
  put getAt(parentRect, 1) + getAt(babyRect, 3) into rH
  put getAt(parentRect, 2) + getAt(babyRect, 4) into rV

  set productRect = rect(lH, lV, rH, rV)
  return productRect
end localToGlobalRect
```

The next set of handlers support drag-and-drop between windows. This functionality is in no way supported from within Director and required the use of another XObject, called dragPicture and written by Geoff Smith, also of Interval Research. DragPicture allows a programmer to designate a PICT image which will follow the cursor and return the coordinates at which it is dropped. By combining this XObject with code which translates between local-and-global coordinates and code which communicates between the various windows, one can build a reasonable facsimile of the standard Macintosh OS drag-and-drop functionality within Director.

```
-- DRAG-AND-DROP HANDLERS
--
-- dragThing tracks which sprite is being dragged and determines if its
-- release point is within a window. If it is, it tells that window to display
-- the dragged object at the point where it was dropped.
-- Though Life is Bait only displays text in the target windows, the handler
-- will handle either text or PICT images.
on dragThing whichSprite
  put 8 into ink
  if whichSprite <> 3  then
       -- dragPicture is an XObject which displays the image as it is being
       -- dragged and returns the point, in global coordinates, at which
       -- it is dropped.
       put dragPicture(mnew) into dragObj
       put the castnum of sprite whichSprite into whichCast
       put ((the width of cast whichCast) / 2 * -1) into offH
       put ((the height of cast whichCast) / 2 * -1) into offV
       put the name of cast whichCast into whichName
```

```
        put dragObj(mDrag, the picture of cast whichCast, offH, offV, ink) into releasePoint
        put item 1 of releasePoint
      put item 2 of releasePoint
      put point(value(item 1 of releasePoint), value(item 2 of releasepoint)) into newPoint
      -- withinAWindow determines if the release point is within a window boundary
      put withinAWindow(newPoint) into hit
      if hit then
            -- talkToMovie tells the window to display the appropriate text/image
            talkToMovie(hit, newPoint, whichSprite)
       end if
  end if
end dragThing

-- withinAWindow determines whether the point designated by
-- releasePoint is within the boundaries of a window which
-- is part of the system
on withinAWindow releasePoint
  put 0 into returnvalue
  set windowL = [davidMovie, goliathMovie, jezebelMovie]
  repeat with n = 1 to count(windowL)
      put getAt(windowL, n) into whichMovie
      put the rect of window whichMovie into windowRect
      if inside(releasePoint, windowRect) then
            put n into returnvalue
      end if
  end repeat
  return returnValue
end

-- globalToLocalRect takes a point designated in global terms
-- in globalPoint and returns its offset into a localWindowRect
-- in terms local to the localWindowRect.
on globalToLocalPoint globalPoint, localWindowRect
  put getAt(localWindowRect, 1)  into windowtop
  put getAt(localWindowRect, 2)  into windowleft

  put getAt(globalPoint, 1) into pointH
  put getAt(globalPoint, 2) into pointV

  put pointH - windowtop into insetH
  put pointV - windowleft into insetV

  set productPoint = point(insetH, insetV)
  return productPoint
end globalToLocalRect


-- talkToMovie tells the movie whichMovie to display
-- sprite whichSprite at the local point releasePoint
on talkToMovie whichMovie, releasePoint, whichSprite
  if whichMovie = 1 then
      put davidMovie into targetMovie
      put the rect of window davidMovie into targetRect
  else if whichMovie = 2 then -- similar for goliath
  else if whichMovie = 3 then -- similar if jezebel
  else -- return 0 to report error
  end if
  put globalToLocalPoint(releasePoint, targetRect) into localizedPoint
  if the castnum of sprite whichSprite > 49 and the castnum of sprite whichSprite < 76 then
      transferText(whichSprite, targetMovie)
  else
      transferIcon(whichSprite, targetMovie, localizedPoint)
```

```
  end if
end talkToMovie

on transferText whichSprite, whichWindow
      put the castnum of sprite whichSprite into thisCastNum
      put the name of cast thisCastNum into thisCastName
      put word 1 of thisCastName into thisCastName
      put "thisCastName:"&&thisCastName
      if whichWindow = "jezebel" then
      put thisCastName&&"jezebel" into thisText
      put jezebelOutline into characterOutline
      else -- similar if whichWindow = 'david'
      else -- similar if whichWindow = "goliath"
      end if
      copyToClipBoard cast (the number of cast thisText)
      moveToFront window whichWindow
      tell window whichWindow
      puppetsprite targetsprite, true
       pasteClipBoardInto cast 61
       set the castNum of sprite targetSprite to 61
      set the loch of sprite targetSprite to 1
      set the locv of sprite targetSprite to 30
      set the rect of cast 61 to rect(getAt(characterOutline,1), getAt(characterOutline,2),(getAt(characterOutline, 3)
      - 15), (getAt(characterOutline, 4) - 20))
      set the ink of sprite targetSprite to 36
      set the forecolor of field 61 to 0
      set the textAlign of field 61 to "left"
      updateStage
      end tell
end transferText

on transferIcon whichSprite, whichWindow, localizedPoint
  put the rect of sprite whichSprite into targetRect
  copyToClipBoard cast (the castnum of sprite whichSprite)
  moveToFront window whichWindow
  tell window whichWindow
      set the visible of sprite targetsprite to false
      pasteClipBoardInto cast targetCast
                puppetsprite targetsprite, true
      set the rect of sprite targetSprite to targetRect
      set the loch of sprite targetsprite to getAt(localizedPoint, 1)
      set the locv of sprite targetsprite to getAt(localizedPoint, 2)
      set the ink of sprite targetSprite to 8
      set the visible of sprite targetsprite to true
      set the castnum of sprite targetsprite to targetCast
      updateStage
  end tell
end transferIcon
```

Breeder

Breeder was envisioned as an environment in which hundreds of text-chunks–be they character-, word- or phrase-sized–could each function inidivudally and with each other without the constant intervention of the programmer or user. This required that each chunk possess behavioural characteristics which determined how it appeared, how it moved, what happened when it encountered another chunk, what happened when it encountered the boundaries of its environment, etc. The appropriate approach for such

programming is an objecect-oriented one. Among its many benefits, object-oriented programming (OOP) allows the programmer to create in memory a cluster of information which is designated as belonging to a particular object. This information is persistent. Thus the programmer, instead of having to constantly make reference to multiple data strcutures to determine what is happening with widget X, can simply 'ask' widget X. It also allows the programmer to create a class of objects which details functionality that he wishes to be common to all members of that class and other functionality which can be particular to any individual member.

Lingo supports a weak version of OOP functionality with its parent-child scripting. *Breeder* uses parent-child scripting to handle its constantly changing, multiple object environment. The base class of objects developed for this is the Maggot class (so called because of the swarming behaviour they exhibit.)

```
-- MAGGOT CLASS DEFINITION / PARENT SCRIPT
--
--
property listPos, myCast, mySprite, mysize, myfont, mystyle, mydirection, myspeed,nextmeme, prevmeme,
    meme, myCastOffset, fatmeme, faceoffset

on birth me, lPos, mSprite
    global gSpeed, myFontsL, firstfill, gMaggotNum, gBoxSprite
    global boxwidth, boxheight, boxtop, boxbottom, boxleft, boxright, boxH, boxV
    global gFirstMaggotSprite, gLastMaggotSprite, gFirstTextCast, gLastTextCast, TextCastNum

    -- <housekeeping>
    -- randomly choose a text-chunk from the cast
    put ((gFirstTextCast - 1) + random(TextCastNum + 1)) into myCastOffset
    -- parse the contents of the new text-chunk to determine
    -- what should go into meme, prevmeme and nextmeme
    put field myCastOffset into fatmeme
    put word 1 of fatmeme into prevmeme
    put stripMeme(fatmeme) into meme
    put word (the number of words in fatmeme) of fatmeme into nextmeme

    -- set direction vector, where the first member of the
    -- the list mydirection represents horizontal direction
    -- and the second member is vertial direction
    put [] into mydirection
    put random(2) into htemp
    if htemp = 1 then append mydirection, htemp
    else append mydirection, -1
    put random(2) into vtemp
    if vtemp = 1 then append mydirection, vtemp
    else append mydirection, -1

    -- set speed, where the first member of the
    -- the list myspeed represents the number of units
    -- moved horizontally in any given cycle and
    -- the second member is the number moved
    -- vertiaclly.
    put [] into myspeed
    append myspeed, random(gSpeed)
    append myspeed, random(gSpeed)
```

```
        -- locate the object at a point on the edge of the
        -- bounding box which is complimentary to its
        -- direction, e.g., if an object general direction
        -- is up and right (mydirection = [-1,1]), have it
        -- start in the bottom left-hand area of the bounding
        -- box.
        set the loch of sprite mySprite to boxH + (0 - (      getAt(mydirection,1) * random(boxwidth/2)))
        set the locv of sprite mySprite to boxV + (0 - (getAt(mydirection,2) * random(boxheight/2)))
        set the visible of sprite mySprite to true
        return me
end


-- moveMaggot determines what happens with an object during each
-- cycle. In this particular case, it combines speed and direction
-- information to first move an object. Then it checks to see
-- if the object has encountered any other objects. If so,
-- it calls a handler which deals with object interaction.
on moveMaggot me
  global boxtop, boxbottom, boxleft, boxright, gBoxSprite, gMaggotNum, gNomadSprite

-- in some cases the environment can get 'stranded' when single
-- words have been combined in such a way that no more combinations
-- can occur but the phrase remains incomplete. If no compatible
-- objects are find within a certain amount of time, then a nomad
-- word is randomly chosen to re-start the process of combining.
  if the timer > 600 then
    puppetsprite gNomadSprite, True
    makeNomad
    set the timer to 150
  end if
  put the mySprite of me into tsprite
  if sprite tsprite intersects gBoxSprite then
        -- does math necessary to actually move an object according to its
        -- speed and direction characteristics.
    put getAt(the mydirection of me, 1) into hD
    put getAt(the myspeed of me, 1) into hS
    set the loch of sprite tSprite to (the loch of sprite tSprite + (hD * hS))
    put getAt(the mydirection of me, 2) into vD
    put getAt(the myspeed of me, 2) into vS
    set the locv of sprite tSprite to (the locv of sprite tSprite + (vD * vS)

        -- the following lines are where ANY desired behavioural characterisitcs
        -- be inserted by the programmer. After a check to make sure that it is
        -- not interacting with itself, the object then calls compatibleMaggot
        -- to see if the object it has encountered has either a previous
        -- string or next string with which to make an inverse match. This
        -- handler is presently written to randmly choose another object
        -- to check for encounter and compatibility; this is done because
        -- otherwise the interactions would happen to quickly to observe. But
        -- it can be easily rewritten to cycle through all the other objects,
        -- using a repeat loop to march through the object list.
    if the mySprite of me <> gNomadSprite then
      put random(count(maggotL)) into n
      put (the mySprite of (getAt(MaggotL, n))) into target
      if sprite (the mySprite of me) intersects target then
        compatibleMaggots(me, getAt(MaggotL, n))
      end if
    end if
    return [1,0,0]
  else
```

```
    return [0, the mySprite of me, the listPos of me]
  end if
  updateStage
end


-- compatibleMaggot compares two Maggot objects to see
-- if either their next and previous, or their previous
-- and next memes match up. If they do, then they can
-- be combined to form a phrase.
on compatibleMaggots me, mate
  global gSpace, thisone, gNomadSprite
  put false into killmate
  put the meme of me into mymeme
  put the prevmeme of me into myprev
  put the nextmeme of me into mynext

  put the fatmeme of mate into itsfatmeme
  put word 1 of itsfatmeme  into itsfirst
  put the number of words in itsfatmeme into num
  put word num of itsfatmeme into itslast
  put the meme of mate into itsmeme
  if myprev = word (the number of words of itsmeme) of itsmeme then
    put the loch of sprite mySprite into oH
    put the locv of sprite mySprite into oV
    set the loch of sprite mySprite to -200
    set the locv of sprite mySprite to -200
    put itsmeme&&gSpace before mymeme
    put mymeme into field the faceoffset of me
    setTextWidth(faceOffset, the mySprite of me)
    put itsfirst&&mymeme&&mynext into myfatmeme
    put myfatmeme into field the myCastOffset of me
    set the meme of me to mymeme
    set the fatmeme of me to myfatmeme
    set the prevmeme of me to the prevmeme of mate
    put true into killmate

  else -- do same except checking for compatiblity between
                  mynext and the first word of the encountered object
  end if
  if killmate then -- dispose of the object
end
```

Telecommunication

For the most part, *Telecommunication* uses the *Breeder* engine. However I developed the beginnings of a timer class which uses Lingo's pseudo-OOP functionality to create timers. I found as the Dynamic Poetry project went on that I was becoming more interested in controlling the dynamics of a piece in multiple times, e.g., processor-time, real-time, interaction-time. Real-timing can be a chore in Lingo, as its built-in timer, called "the timer", it is very confusing to use, returns results in ticks as opposed to seconds or minutes, and is awkward when you want to keep track of more than one timestream. The timer class below addresses these problems by letting the programmer

instantiate each timer as an object which “knows” when it is started, when it is stopped and can, when queried, return the amount of time elapsed since it was started in either seconds or minutes, and reset itself.

```
-- TIMER CLASS DEFINITION / PARENT SCRIPT
--
-- Timer creats an object which keeps track time. It supports
-- starting, stopping, resetting a timer, as well as returning
-- values as either seconds or minutes. It could be easily
-- extended to handle lap and countdown functionality.
property startTime, presentTime, stopTime, timeElapsed, units

-- timebase : “s” for seconds or “m” for minutes.
on birth me, timebase
  if timebase = "s" then put 60 into units
  else if timebase = "m" then put 3600 into units
  else if voidP(timebase) then put 3600 into units
  else put "ERROR Incorrect timebase: use M = minutes or S = seconds"
  put the ticks/units into startTime
  put "startTime:"&&startTime
  return me
end

on getElapsed me
  put the ticks/units into  presentTime
  put "presentTime:"&&presentTime
  put presentTime - startTime into timeElapsed
  put "timeElapsed:"&&timeElapsed
  return timeElapsed
end

on restartTimer me
  put the ticks/units into startTime
end
```

Telecommunication also led me to write a short but very useful piece of code which, given two objects A and B, will handle moving object A to object B. As presently written it does not find the shortest path but rather the closes both the horizontal and vertical distance in equal measure until the distance along one dimension is zero. Then it closes the distance on the remaining dimension. Extending it to find the shortest path – or even specified paths – should not be difficult.

```
-- moveYtoX will move an object Y to the location of object X.
--
-- xObj : the sprite number of object X
-- yObj : the sprite number of object Y
-- xHoffset : the amount of distance you wish the horizontal location of
-- y to be offset from the horizontal location of x.
-- xVoffset : the amount of distance you wish the vertical location of
-- y to be offset from the vertical location of x.
-- speed : the number of pixels you wish Y to move with each cycle.
on moveYtoX xObj, yObj, xHoffset, xVoffset, speed

  put the loc of sprite xObj into xLoc
```

```
  put getAt(xLoc,1) into xH
  put getAt(xLoc,2) into xV
  put xH + xHoffset into xH
  put xV + xVoffset into xV

  put the loc of sprite yObj into yLoc
  put getAt(yLoc,1) into yH
  put getAt(yLoc,2) into yV

  if xH > yH then put 1 into hDir
  else put -1 into hDir

  if xV > yV then put 1 into vDir
  else put -1 into vDir

  put 1 into moveH
  put 1 into moveV

  repeat while moveH or moveV
    if not  (yH < xH + speed and yH > xH - speed) then
      set the locH of sprite yObj to yH + (hdir * speed)
      put the locH of sprite yObj into yH
    else
      put 0 into moveH
    end if
    if not ( yV < xV + speed and yV > xV - speed) then
      set the locV of sprite yObj to yV + (vdir * speed)
      put the locV of sprite yObj into yV
    else
      put 0 into moveV
    end if
    updateStage
  end repeat
end
```

Appendix B Illustrations

Bibliography

Apollinaire, Guillaume. *Calligrammes*,. Trans. Anne Hyde Greet. Los Angeles: University of California Press Berkeley, 1980.

Arnheim, Rudolf. *Film As Art*. Berkeley: University of California Press, 1957.

Bolter, Jay David. *Writing Space: The Computer, Hypertext, and the History of Writing*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1991.

Brown, Alex. *inPrint : Text and Type*. New York: Watson-Guptill Publications, 1989.

Bush, Vannevar. "As We May Think". *Atlantic Monthly* (July, 1945): 101-108.

Campbell, Jim. "Delusions of Dialogue: Control and Choice in Electronic Art". Given at Interval Research Corporation, Palo Alto, California: November, 1996.

Celan, Paul. *Poems of Paul Celan*. Trans. Michael Hamburger. New York: Persea Books, 1988.

Cliff, Stafford, ed. *Cutting Edge Typography*. London: B.T. Batsford, Ltd., 1994.

Dair, Carl. *Design with Type*. Toronto: University of Toronto Press, 1967.

Danto, Arthur C. "TV and Video." *The Nation* 26, no. 7 (September 1995): 248

Drucker, Johanna. *The Visible Word: Experimental Typography and Modern Art, 1909 - 1923*. Chicago: University of Chicago Press, 1994.

Eisenstein, Elizabeth. *The Printing Press as an Agent of Change: Communications and Cultural Transformations in Early-Modern Europe*. Vol. 2. New York: Cambridge University Press, 1979.

Eisenstein, Sergei M. *The Film Sense*. Trans.. and ed. Jay Leyda. London: Faber and Faber Limited, 1953.

Eisenstein, Sergie M. *Film Form*. Trans. Jay Leyda, Cleveland: World Publishing, 1957.

Febvre, Lucien. *The Coming of the Book: The impact of printing 1450-1800*. Trans.. David Gerard. London : Verso, 1976.

Massachusetts Institute of Technology Visual Language Workshop. "Designing an Information Landscape in Time and Space : The Dynamic Visualization of Information in Two and Three Dimensions." *Frames*: 32 (1994): 1 -12.

Frutiger, Adrian. *Type Sign Symbol*. Trans. Andrew Bluhm. Zürich: ABC Verlag, 1980.

Heaney, Seamus. “Crediting Poetry. Seamus Heaney’s Nobel Lecture.” World Wide Web site: <http://svenska.gu.se/SvenskaAkademien/nobel/heaney_eng.html> (1995).

Ishizaki, Suguru. “Multiagent Model of Dynamic Design : Visualization as an Emergent Behavior of Active Design Agents”. pp. 347 - 354, *Conference Proceedings, Computer Human Interaction ‘96*. New York: The Association for Computing Machinery, 1996.

Kang, T. Jin, and Muter, Paul. “Reading dynamically displayed text.” *Behavior and Information Technology* 8, no. 1, p. 33-42.

Kelly, Kevin. “Interactivity Matrix”. *Wired* 1.1 (September/October 1993): 42.

Landow, George P. *Hypertext : The Convergence of Contemporary Critical Theory and Technology*. Baltimore: The John Hopkins University Press, 1992.

Lanham, Richard A. “Digital Literacy”. *Scientific American* 273, no. 3 (September 1995): 160-161.

Lechte, John. *Fifty Key Contemporary Thinkers : From Structuralism to Postmodernity*. London: Routledge, 1994.

Lupton, Ellen. "Words Made Flesh: Johanna Drucker.” *Eye* 5, no. 8 (Autumn 1995): 72–77.

Maeda, John. “Metadesigning 1 - 3.” World Wide Web site: <http://www.imrf.or.jp/MdN1.html.> (1995).

Maeda, John. *The Reactive Square*.Tokyo: Digitalogue, 1995.

Mann, Ron ed. *Poetry in Motion*. New York: The Voyager Company, 1992. CD-ROM.

Manovich, Lev. “The Aesthetics of Virtual Worlds : Report from Los Angeles”. *CTHEO RY* 19, no 1-2. Electronic magazine: <ctheory@concordia.ca>: September 1996.

Müller, Thomas. *Understanding Concrete Poetry*. Pasadena, California: self-published, 1995. CD-ROM.

Nelson, Theodore. *Literary Machines*. Self-published, 1987.

Ong, Walter J. *Orality and literacy : the technologizing of the word*. New York: New Accents, 1982.

Ong, Walther J. *Interfaces of the Word*. Ithaca, New York: Cornell University Press, 1977.

Poynor, Rick and Edward Booth-Clibborn. *Typography Now : The Next Wave* . London, England : Internos Books, 1991.

Roberts, Philip Davies. *How Poetry Works: The Elements of English Poetry*. London: Penguin Books, 1991.

Rushkoff, Douglas. *Media Virus : Hidden Agendas in Media Culture*. New York: Random House, 1994.

Sklar, Robert. *Film: An International History of the Medium*. London: Thames and Hudson Ltd., 1993.

Small, D. "Expressive typography: high quality dynamic and responsive typography in the electronic environment". Masters thesis, Massachusetts Institute of Technology, 1987.

Smith, Keith A. *Structure of the Visual Book*. New York : The Sigma Foundation, 1992.

Solt, Mary Ellen. *Concrete Poetry : A Word View*. Bloomington, Indiana: Indiana University Press, 1971.

Spencer, Herbert. *Pioneers of Modern Typography*. New York: Hastings House, 1969.

Staples, Loretta. "What happens when the edges dissolve." *Eye*. 18, vol. 5. (Autumn 1995): 6–7.

Tschichold, Jan. *The Form of the Book : Essays on the Morality of Good Design*. Ed. Robert Bringhurst, trans.. Hajo Hadeler. London: Lund Humphries, 1991. Originally published as *Ausgewählte Aufsätze über Fragen der Gestalt des Buches und der Typographie*,(Basel: Birkhäuser Verlag, 1975).

Williams, Emmett, comp. *An anthology of concrete poetry*. New York : Something Else Press, 1967.

Wong, Yin Yin. "Temporal Typography : A Proposal to Enrich Written Express." *Conference Companion, Computer Human Interaction '96*. New York: The Association for Computing Machinery, 1996.

Wong, Yin Yin. "Temporal Typography: Characterization of Time-Varying Typographic Forms." Master's thesis, Massachusetts Institute of Technology, 1995.