Losing Control: Software poetics in the everyday

Geoff Hinchcliffe

December, 2012

A thesis submitted for the degree of
Doctor of Philosophy in Communication
University of Canberra

The increasing presence of computer technology in our everyday lives has motivated an “Aesthetic Turn” (Udsen & Jørgensen, 2005); a growing field of research dedicated to better understanding computing in the everyday cultural context. The researchers of the Aesthetic Turn claim that computer technologies developed for the office, with their focus on efficiency and utility, are inappropriate in the personal setting of the “everyday”. They argue for an understanding of computing as an aesthetically experiential medium and call for a radical recalibration of the conceptions and production practices which shape our everyday technologies. Central to their approach is a notion of aesthetic experience informed by the Pragmatist philosophy of Dewey and Shusterman. It is a conception of aesthetics that is phenomenological, emergent and situated, and which opens the opportunity for new forms of engagement with everyday technology. Theirs is a vision in which computing need not only be useful but may be surprising, delightful, provocative or even critical.

For producers, the Aesthetic Turn offers immense creative opportunity, but also introduces significant conceptual and technical challenges; from fundamental questions regarding the nature of emergent experience, to the impact of aesthetics on production processes. Dunne declares that “The most difficult challenges for designers of electronic objects now lie not in technical and semiotic functionality, where optimal levels of performance are already attainable, but in the realms of metaphysics, poetry, and aesthetics, where little research has been carried out” (Antony Dunne, 2005b).

Adopting a production-based methodology, this research explores the creative potential of the Aesthetic Turn and addresses some of the significant conceptual and technical questions arising from its experiential framing of computing. Using reflective analysis of creative works and their development processes, the research brings a production perspective to the discourse, providing details on how general notions of aesthetics and poetics may be reconciled within the highly specified and technical context of software production. Through the process of creative and critical enquiry the research advances understanding of computational aesthetics, developing a conception of software as a material medium to be creatively explored through code. It develops the conceptual model of “data as script” to encapsulate the practices of interpretation and adaptation that are central to the poetics of the project’s creative works.

Contents

List of Figures

Introduction

The term “everyday” has come to represent computing’s increasing presence in all aspects of our daily lives, and the subsequent need to poeticise our interactions with everyday technologies. For producers, the everyday offers immense creative opportunity but introduces significant conceptual and technical issues, as Dunne declares; “The most difficult challenges for designers of electronic objects now lie not in technical and semiotic functionality, where optimal levels of performance are already attainable, but in the realms of metaphysics, poetry, and aesthetics, where little research has been carried out” (Antony Dunne, 2005b).

This thesis presents theoretical research arising from a creative practice-based enquiry into software poetics in the everyday context. It begins by providing a context for the research. Chapter two outlines the theoretical context, focusing on a growing field of researchers concerned with computing in the everyday cultural context. They argue that computer technologies developed for the office, with their focus on efficiency and utility, are inappropriate in the personal setting of the “everyday”. In what has been termed the “Aesthetic Turn” (Udsen & Jørgensen, 2005), they promote an understanding of computing as an aesthetically experiential medium and call for a radical recalibration of the conceptions and production practices which shape our everyday technologies. Theirs is a vision in which

computing need not only be useful but may be surprising, delightful, provocative or even critical.

Having established a formal theoretical context for the research, chapter three provides a practical context, looking at the work and practices of a new culture of software producers that have emerged as a direct consequence of the everyday transformation of computing. Their culture is unburdened by the historical baggage of HCI and they view software as an aesthetic medium of experience, working speculatively to develop innovative forms and techniques.

Chapters four and five concern the creative work. Chapter four outlines the creative project, the methodology and technical context of the creative production. The project involves the creative production of speculative software realisations of the Twitter social media service. It employs a practice-based (Schön, 1983) methodology in which critical reflection is used to generate formal knowledge from the creative practice. Chapter five details the creative works, providing commentary on the novelty of their forms, technologies and processes of production.

Chapters six and seven present critical and conceptual analysis emerging from the creative process. Chapter six considers aesthetics in a computational context, and looks for insights in both the established field of graphic design and the emerging field of “creative code”. Chapter seven discusses the role and importance of conceptual models and their impact on conceptions and production of computing technology. It proposes a new conceptual model to better address the practices of interpretation and adaptation which are central to production of poetic forms.

Finally, chapter eight reflects on the most significant contributions of the research in practical, theoretical and technical terms, and closes with consideration of some of the broader themes emerging from the work.

The Everyday

2.1. Introduction

In this section we examine literature from a growing field of researchers who argue that the computer's transition, from office to everyday, constitutes a major paradigm shift for the field of Human Computer Interaction (HCI). The essence of their claim is that HCI, with its focus on workplace efficiency and utility, is ill-equipped to address the aesthetic needs of the everyday. They propose that new framings are required in which computing is recognised as a sophisticated aesthetic medium rather than a workplace tool. Central to new approaches for the everyday is a notion of aesthetic experience in the spirit of the Pragmatist philosophy of Dewey (1934) and Shusterman (1992). It is a conception of aesthetics that is phenomenological, emergent and situated, and which opens the opportunity for poetic engagement with everyday technology.

2.2. The Everyday

Once confined to the workplace in the form of the office PC, computer technology today is a ubiquitous presence in our everyday lives. Many factors have perpetuated computing's transition from the office to the everyday (mobile phones, gaming consoles, CD-Roms) but none more than the Internet and the Web. The Web motivated people to use their computers for non-production tasks, for accessing information recreationally and later, with Web 2.0, for participating in the creation of information and data. Manovich describes its dramatic influence:

> *"In the 1990s, as the Internet progressively grew in popularity, the role of a digital computer shifted from being a particular technology (a calculator, a symbol processor, an image manipulator, etc.) to a filter for all culture, a form through which all kinds of cultural and artistic production were mediated. As the window of a Web browser replaced cinema and television screen, the art gallery wall, library and book, all at once, the new situation manifested itself: All culture, past and present, came to be filtered through a computer, with its particular human-computer interface" (Manovich, 2001).*

The term "everyday" has come to represent computing's general migration from the workplace into all aspects of our daily lives. This recontextualisation, from office to the everyday, raises important implications for producers of computer technology, as Petersen explains:

> *"The penetration of interactive technology in everyday life fundamentally challenges the way we understand and design for human-computer interaction. As computer systems change from being very specific tools for work to … just ordinary everyday things, the meaning of human-computer interaction also changes. …we are dealing with new devices and new qualities of use which are not necessarily related to efficiency but rather related to emotional qualities, to experiential qualities, and to aesthetic qualities"* (Petersen, Hallnäs, & Jacob, 2008).

The increasing significance of the everyday for HCI research is evident in titles such as "Designing Everyday Computational Things" (Redström, 2001); "From Use to Presence: On the Expressions and Aesthetics of Everyday Computational Things" (Hallnäs & Redström, 2002); and "Emotional Design - why we love (or hate) everyday things" (Norman, 2004). Dunne's "Critical Design", seeks "to integrate

aesthetic experience with everyday life" (Antony Dunne, 2005b); Petersen et al's "Aesthetics of Interaction" is motivated by "the integration of computing into everyday life" (Petersen et al., 2008); while Blythe, Bardzell et al go so far as to claim that "computing technology is now so pervasive that the study of Human Computer Interaction (HCI) is almost the study of everyday life" (Blythe, Bardzell, Bardzell, & Blackwell, 2008).

Rather than a simple change in context, Redstrom suggests that the everyday requires a significant recalibration of HCI practices and values:

> *"To design for everyday life involves much more than enabling people to accomplish certain tasks more effectively, and therefore, traditional approaches to human-computer interaction that focus on usability are not sufficient. To support critical discussion of, and reflection upon, the design of everyday computational things, both new design philosophies and a richer collection of design examples are needed"* (Redström, 2001).

The emphasis of the different authors cited here is on the cultural determination of technology. While we tend to regard the evolution of technology in Darwinian terms, where the best and brightest innovations triumph over those less fit for purpose, the reality is far messier. Technologies are shaped by a complex array of social, economic and technical factors (Green, 2002), and the result is that technologies reflect the biases and agendas of the cultures which produce them[1] (Wajcman, 1991). In the case of HCI, the field was formed in the era of the office computer and developed its practice around the pragmatics of the workplace. Now, with the shift to the everyday, HCI's established practices cannot adequately address computing's new cultural priorities, and could even be at cross purposes; "When used in the home to mediate social relations, the conceptual models of efficient communication embodied in office equipment leave little room for the nuances and quirks on which communication outside the workplace relies so heavily" (Antony Dunne, 2005b).

---

[1] Feminist authors such as Green and Wajcman cite the bias of technology as an issue of power, with the military, corporations and men having the most influence over the form and function of the technology which shapes our daily lives. Interestingly, the usability movement championed by Donald Norman responds to similar issues of misrepresentation in the design and manufacture of our domestic technologies. But where feminist readings see power brokers as actively seeking to maintain their privilege, User Centred Design (UCD) aims to repair the imbalance by involving users in the production process.

This is not the first time that authors have argued for HCI to expand its instrumental focus and consider computing from cultural and experiential perspectives[2], but what distinguishes the current situation is the saturation of computer technology in daily life; "the current pervasiveness of digital technology in our everyday life, including the Web and the mobile phone" makes it "difficult to distinguish a 'user experience' from any other kind of experience" (Fallman, 2011).

Clearly, there is a growing consensus[3] amongst HCI researchers that the cultural context of the everyday represents a significant change from the office environs in which HCI developed its practice. Accordingly, the researchers cited above are emphatic in their opinion that new conceptions and approaches to the production of technology are required. The following theoretical frameworks are indicative of this thinking. They attempt to identify this shifting HCI agenda and place it in a broader context alongside previous and concurrent conceptions of computing.

2.3. Third Paradigm

The first of our four theoretical framings is Harrison et al's "Third Paradigm". The authors use the concept of "paradigms" to describe the "successive and overlapping waves which fundamentally re-frame ideas" in HCI. At the heart of each of their paradigms is a different metaphor of interaction. "A paradigm shift, then, could be said to occur when a new generative metaphor is driving new choices of what to research and how, and can be identified when problems and issues that used to be marginalised have moved to the centre" (Harrison, Sengers, & Tatar, 2007).

Emerging from the field of "human-factors" with its focus on ergonomics, the First Paradigm employed an engineering metaphor of a "man-machine coupling" and pursued highly pragmatic goals relating to efficient performance and aims of optimising the fit between user and machine. The shift to the Second Paradigm

---

[2] See for example Laurel's seminal "Computers As Theatre" (1991) which identified the performative and experiential nature of computer interaction. Maeda's "Design by Numbers" (1999) argues for the computer as an artistic medium in its own right, while Fishwick's "Aesthetic Computing" (2004) brings together the disciplines of art and computer science.

[3] (McCarthy & Wright, 2007) (Petersen, Hallnäs, & Jacob, 2008) (Fishwick, 2004) (Fallman, 2007) (Fallman, 2011) (Dunne, 2005) (Greenberg & Buxton, 2008) (Sengers, 2004)

involved a transition from the First Paradigm's "man-machine coupling" to a cognitive metaphor of "mind and computer as coupled information processors" (Harrison et al., 2007). In the Second Paradigm, the first's focus on efficiency was preserved but pursued within an information-processing model.

Harrison et al argue that the reductive models of both First and Second Paradigms are indicative of HCI's valuing of "rationality and rational analysis [as] the most important mode of operation for human activity" (Flyvbjerg quoted in (Harrison et al., 2007)). Both First and Second Paradigms provided concrete models and metaphors for research and production but overlooked messier phenomena associated with human-computer interaction, particularly those characteristics which suggested variable and contextual notions of interaction. First and Second Paradigms also marginalised subjective characteristics related to the experiential and emotional aspects of interaction. By contrast, Harrison et al's Third Paradigm seeks to actively engage with all of these marginalised aspects of interaction, working with a central metaphor of "interaction as phenomenologically situated" (Harrison et al., 2007).

The goal of Third Paradigm HCI is to "support situated action and meaning-making in specific contexts" and address questions relating to "how to complement formalised, computational representations and actions with the rich, complex, and messy situations at hand around them" (Harrison et al., 2007). Unlike the First and Second Paradigms, which sought to develop generic universal models of interaction, the Third Paradigm emphasises "situated perspectives" (Harrison et al., 2007); "rich, complex, and messy situations" which are indicative of everyday contexts.

The authors provide the following table (Figure 1) as a summary of the three different Paradigms.

	Paradigm 1: Human Factors/ Engineering	Paradigm 2: Cognitive Revolution	Paradigm 3: Situated Perspectives
Metaphor of interaction	Interaction as man-machine coupling	Interaction as information communication	Interaction as phenomenologically situated
Central goal for interaction	Optimizing fit between man and machine	Optimizing accuracy and efficiency of information transfer	Support for situated action in the world
Typical questions of interest	How can we fix specific problems that arise in interaction?	What mismatches come up in communication between computers and people? How can we accurately model what people do? How can we improve the efficiency of computer use?	What existing situated activities in the world should we support? How do users appropriate technologies, and how can we support those appropriations? How can we support interaction without constraining it too strongly by what a computer can do or understand? What are the politics and values at the site of interaction, and how can we support those in design?

Figure 1. Three Paradigms of HCI (Harrison et al., 2007)

2.4. Third Wave

The concept of the Third Paradigm closely parallels that of "Third Wave" HCI (S. Bødker, 2006). In the First Wave, commencing in the early 1980s, HCI equated usability with efficiency and ease of use. It drew on cognitive psychology's information processing theories in a bid to achieve a close fit between a designer's and user's comprehension of a system. The Second Wave, at the end of the 1980s, was a response to the limits of the First Wave's "disembodied emphasis on a single user operating a single application" (Fallman, 2011). Second Wave approaches abandoned rigid systematic testing regimes in favour of participatory and contextual methods of field study such as "participatory design workshops, prototyping and contextual inquiries" (S. Bødker, 2006). The Second Wave's focus on groups and communities of practice promoted theoretical frameworks such as Situated Action (Suchman, 1987), Distributed Cognition (Hutchins, 1995) and Activity Theory (Engeström & Escalante, 1995). However the First Wave's focus on work and productivity endured.

The Third Wave emerges with the cultural shift to the everyday, broadening the contexts and nature of computing to address elements such as "meaning, complexity, culture, emotion, lived experiences, engagement, motivation, and experience" ((Fallman, 2011) (S. Bødker, 2006)). In many respects, the Third Wave

is defined by what the Second Wave is not; Third Wave is “non-work, non-purposeful, non-rational, etc.” (S. Bødker, 2006) In addition to greater attention to aesthetic, emotional, and experiential factors, the Third Wave is characterised by a more designer-centric approach, distancing the user-centred rationale of the First and Second waves.

Figure 2 compares the different phases of the Paradigms and Waves framings.

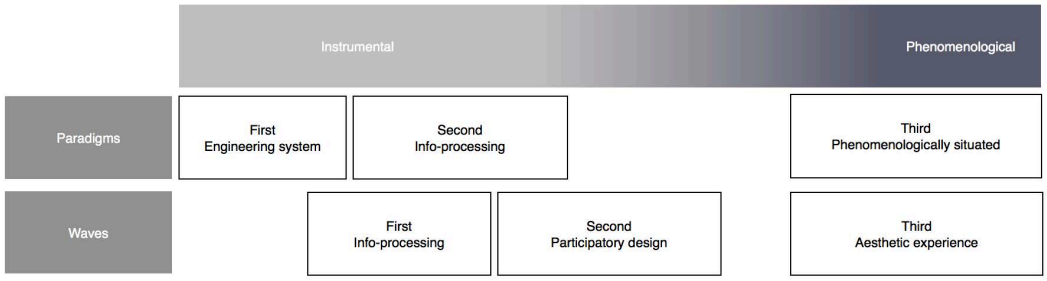


Figure 2. Third Paradigm & Third Wave.

While the details of the stages preceding the Third phases vary between the two framings, the main focus is their proposition of the Third phase as a phenomenologically-based approach to computing. The X-axis in the diagram is not indicative of time, but of the orientation between Instrumental and Phenomenological conceptions of experience with technology. Needless to say, each successive phase of both frames situates HCI practice further into phenomenological terrain.

2.5. Fifth Perspective

In yet another response to the everyday, Petersen et al (Petersen et al., 2004) augment Bødker & Kammersgaard’s “Four Perspectives on HCI” (1984). Unlike the framings of the Third Paradigm/Wave, the categorisations in the Perspectives model are less periodised. The emphasis in the Four Perspectives is on the role of the user, the model of interaction and the “interaction ideals” promoted in each of the perspectives.

The different perspectives are represented in the following table (Figure 3).

Table 1: from four to five different perspectives on HCI – elaborated on the model proposed in [5]

Perspec-tive/	System	Tool	Dialo-gue Partner	Media	Aesthe-tic Experi-ence
Man	system compo-nent	master	equal partner	Commu-nicator	Impro-visator
Man-Machine Inter-action	between equal partners	Media-ted by machine	Man machine dialogue	Suppor-ting human-human dialogue	Play
Interac-tion ideals	efficiency	Transpa-rency	human dialogue	Communi-cation	Intrigue

Figure 3. Five Perspectives on HCI. (Petersen et al., 2004)

In the System perspective user and computer are both regarded as components within a system, whereas in the Tool perspective, the user is afforded greater significance. Engeström uses Actor-Network theory to provide a compelling account of the Dialogue perspective (Engeström & Escalante, 1995). Actor-network theory's reference to human and non-human "actors" emphasises the relationship between user and computer as a form of dialogue, a negotiation about respective roles and outcomes. The Media perspective places computing technology in a mediating role within a human-human communication model. And finally, Petersen et al's fifth perspective, that of Aesthetic Experience, sees the user as "Improvisor" creatively exploring computing technology in a playful manner. Their aesthetic perspective does not treat human and technology as equal (as per the Dialogue perspective) but "acknowledges man's ability to interpret and appropriate technology" (Petersen et al., 2004). Rather than a means to an end, or mastery (as per the Tool perspective), the Fifth Perspective emphasises a highly personal appropriation of a work. The authors also stress the importance of bodily

experience in their Aesthetic perspective, resisting the idealised symbolic models of the Media perspective.

Figure 4 provides a consolidated view of the three different framings.

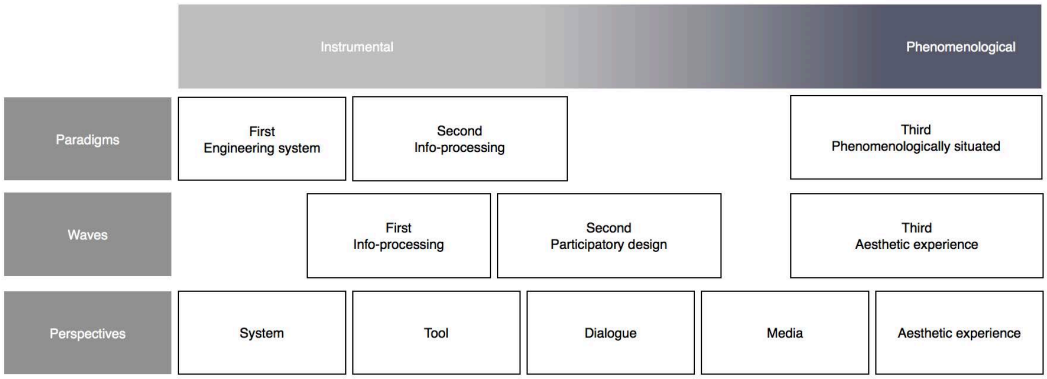


Figure 4. Paradigms, Perspectives & Waves combined.

In my interpretation, each successive perspective represents a further shift toward a phenomenological conception of computer interaction, with the final perspective, Aesthetic Experience, engaging explicitly with the phenomenological conception. The diagram provides a clear indication of the correlations between phases in the different framings, but again, the most notable feature is the congregation in the Phenomenological zone.

2.6. Aesthetic Turn

The fourth and final framing considered here is Udsen and Jørgensen's "Aesthetic Turn" (Udsen & Jørgensen, 2005). Whereas the previous framings take a long view of the cultural development of computing, positioning new conceptions of computing alongside previous phases, Udsen and Jørgensen instead focus only on recent aesthetic approaches, what they term the "Aesthetic Turn". Like the final category of each of the three previous framings, the Aesthetic Turn is an attempt to address the recent focus on aesthetics in computing research and literature. The difference with Udsen and Jørgensen's framing is that it aims to define a number of different approaches within the single Aesthetic category.

The authors identify four different approaches: Cultural; Functional; Experience-based; and Techno-futurist. These four approaches are exemplified by artefacts, key theorists, and academic traditions. Drawing from Traditional HCI and Usability, the Functionalist approach is represented by authors such as Norman, Jordan, and Tractinsky all of whom advocate aesthetics as a device for improving usability outcomes, exemplified by Norman's adage "attractive things work better" (Norman, 2003). The Cultural approach's most prominent advocates are theorists from the humanities and cultural studies such as Manovich (Language of New Media, Interaction as an Aesthetic Event), Johnson (Interface Culture), Bolter and Grusin (Windows & Mirrors). Rather than the HCI standpoint of computer interaction as a science, the Cultural approach considers it as a cultural medium and its creation as an act of cultural production. Accordingly, the authors are concerned with computer media as an expressive aesthetic form. Udsen and Jørgensen's Experience-based category includes authors such as Löwgren, Dunne, and Blythe et al, but is best represented by McCarthy and Wright, authors of "Technology as Experience" (2004), whose use of the term aesthetics is synonymous with the Pragmatist philosophies of Dewey and Shusterman. In the Experience-based approach, interacting with computer technology is understood as involving experiences that are situated, emergent, and "felt". Finally, the Techno-Futurist category sees aesthetics approached from philosophical positions, with authors such as Hallnäs, Redström, and Dourish cited as exemplars.

The following table (Figure 5) encapsulates Udsen and Jørgensen's four classifications.

	1. Cultural approach	**2. Functionalist approach**	**3. Experience-based approach**	**4. Techno-futurist approach**
Academic traditions	Humanities New media	Traditional HCI Usability	Interaction design	Philosophy
Type of interfaces	Non-informational spaces	Informational interfaces	Post-optimal objects	Ubiquitous computing environments
Theorists	Laurel Johnson Manovich Bolter & Gromala Walther Pold	Tractinsky Jordan Norman Karvonen Desmet	Blythe et al. Dunne Gaver et al. Löwgren McCarthy & Wright	Dourish Hallnäs & Redström Ishii & Ullmer

Figure 5. Four Approaches of the Aesthetic Turn. (Udsen & Jørgensen, 2005)

While the Instrumental, Cultural, and Experience-based approaches are clearly defined and well represented with appropriate theorists and traditions, the Techno-Futurist classification is less convincing. The concepts of Hallnäs and Redström (such as “slow technology” and technology “presence”) and their production prototypes have strong correlations with the work and rationale of Dunne and Raby’s Critical Design (included in the Experience-based category). Meanwhile the work of Dourish with its emphasis on a phenomenologically based understanding of aesthetic experience aligns well with the pragmatist inspired work of McCarthy and Wright (also Experience-based). Nonetheless, Udsen and Jørgensen’s focus on a purely aesthetic set of classifications is an interesting and worthy contribution. In my analysis, the most significant aspect of their classification is the inclusion of both instrumental and phenomenological conceptions within the single meta-category of aesthetics. The issues of this association are discussed in more detail in the following summary.

2.7. Four Frameworks - A Summary

At the outset of this summary, it is worth noting a couple of points regarding the various classifications. Firstly, while the framings identify different phases in HCI, they do so without suggestion that any phase is either universally adopted or protected from contamination by previous paradigms/waves/perspectives. For example, in their definition of the Third Paradigm, Harrison et al argue against a Kuhnian absolutist appraisal in which only one paradigm can be considered “right”. Instead, the authors favour Agre’s “generative metaphors” (1997) for its ability to support multiple parallel framings, enabling designers to adopt “old” metaphors where appropriate, or mix concepts and metaphors from different waves, perspectives and paradigms to suite their particular needs.

Secondly, it is also worth noting that even when a paradigm is associated with a particular period, producers are not. For example, Laurel’s seminal 1991 text “Computers as Theatre” could be placed within the Third Wave, whereas Tractinsky’s 2012 model of aesthetic processing fits comfortably with the information processing ideals of the Second Paradigm (Tractinsky, 2012).

Acknowledging these points and differences in nomenclature and timings, the different framings share many common classifications. The following diagram (Figure 6) consolidates the four different framings.

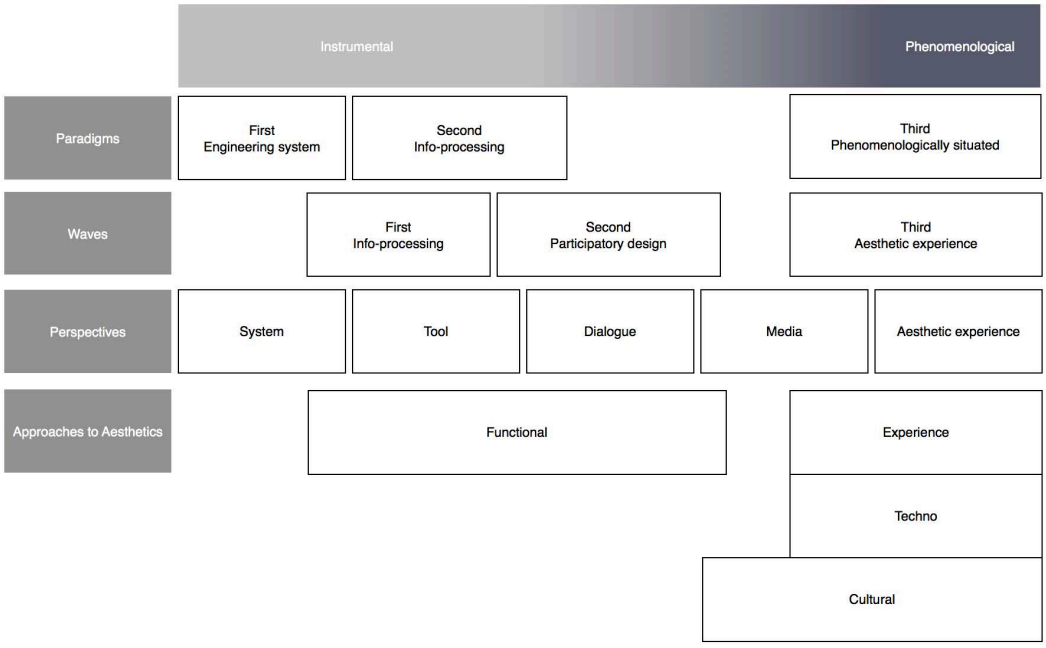


Figure 6. Four frameworks combined.

Udsen and Jørgensen's title "The Aesthetic Turn" is an apt descriptor for the core concern of the new waves/paradigms/perspectives. While the framings have been inspired by the everyday context, they are ultimately about an aesthetic agenda. In my analysis, the Third Wave/Paradigm and the Fifth Perspective share much in common with the Udsen and Jørgensen's Experience-based category. The emphasis in all of these categories is on experience with technology as phenomenological, situated and emergent, which, in my appraisal, the Techno-futurist and Cultural approaches also share. The exception to this phenomenological focus is the Functional approach included in Udsen and Jørgensen's classification. The variations between their Cultural, Experience-Based and Techno-Futurist approaches are relatively insignificant in comparison to those between instrumental and experiential approaches to aesthetics. Udsen and Jørgensen are right to cite the Functional approach alongside their other aesthetic categories, for all are actively utilised by contemporary researchers (Tractinsky, 2012). However, the Functional aesthetic approach would not be located within the Third Wave/Paradigm (or Fifth Perspective) and is more aligned with Second Wave/ Paradigm aims and practices. Despite this apparent disparity, the Functional

approach provides a useful counter to experiential approaches, and helps to emphasise what is truly novel about adopting a phenomenological conception of aesthetics for computing technology. The following analysis focuses on the experiential aesthetics which characterise the new wave in computing, and considers the implications to technology forms, audiences and producers.

2.8. Instrumental Aesthetics

The Functional approach to aesthetics represents the least significant change to established HCI conceptions and practices. The Instrumental/Functional approach to aesthetics is epitomised by Norman's adage "attractive things work better" (Norman, 2003). In simple terms, functional/instrumental approaches consider aesthetics to mean visual beauty, and treat it as another usability factor (Bertelsen & Pold, 2002) to be measured and engineered (Ben-Bassat, Meyer, & Tractinsky, 2006). In his recent chapter on "Visual Aesthetics" in the Encyclopedia of Human-Computer Interaction, Tractinsky outlines his understanding of the term: "I use the term "aesthetics" in its fairly ordinary and common sense as reflected in dictionary definitions such as "an artistically beautiful or pleasing appearance" (The American Heritage Dictionary of the English Language), or as "a pleasing appearance or effect: Beauty" (Merriam-Webster's Collegiate Dictionary)." (Tractinsky, 2012)

Bardzell's comprehensive critical response to Tractinsky's entry is published in the same volume (J. Bardzell, 2012). In it, Bardzell argues that by sidestepping the extensive field of aesthetic philosophy (citing only dictionary definitions) and instead placing all focus on scientific scholarship, Tractinsky adopts an a-theoretic position as part of a rhetorical strategy to avoid engaging with concepts that are considered messy or inconvenient to the information processing model at the centre of his thesis. Through a close critical reading of Tractinsky's chapter, Bardzell demonstrates that Tractinsky's supposedly agnostic scientific position is actually closer to a philosophical theory of "aesthetic processing". As Bardzell sees it, by underplaying the significance of aesthetic theory, including that of his own work, Tractinsky's "a-theoretic position exempts itself from critical scrutiny, since it denies the existence of its own theoretical constructedness and normative commitments, presenting itself innocently as mere empirical data; such data, this position implies, can be scientifically but not philosophically interrogated." (J. Bardzell, 2012)

In simple terms, Tractinsky seeks to avoid the subjective nature of aesthetic experience which is at the heart of the majority of thinking and philosophy on the subject. Instead, he adopts a reductive cognitive model of stimuli and response. While Bardzell is uncomfortable with the “aesthetic processing “ model that Tractinsky proposes, his criticism is largely concerned with Tractinsky’s unwillingness to engage with the philosophical basis of his own theoretical model. This Bardzell claims is not only deceptive but also limiting to HCI scholarship.

On a more practical level, Matthews et al express grave concern with deterministic approaches to aesthetic experience. In prescribing experiential outcomes, Matthews et al warn that designers perpetuate “an understanding of successful design as contingent upon accurate predictions of users’ interpretations and behaviour” (Matthews, Stienstra, & Djajadiningrat, 2008), where any deviation from “intended use” is regarded as problematic. The issue with attempting to measure and predict users’ interpretations and behaviour is that irrespective of the process employed, our knowledge of user and use can never match the complexity or subtleties of actual use. By proceeding with narrow parameters of use and user, we “risk trapping people in a situation where the use of our designs has been over-determined and where there is not enough space left to act and improvise” (Redström, 2006).

If as Redstrom fears, the ability to interpret and improvise is lost, the result is a mode of use “that is felt as coercive, not satisfying, with decreasing outlets for individuality” (Jones, 1988, p. 221 in (Redström, 2006)). Ironically, what Redström and Jones point to here is that the designer’s efforts to determine use, motivated by their desire to assist their audience, ultimately constrain them to a negative extent. Dunne sees this as a form of enslavement “to the conceptual models, values, and systems of thought” (2003) embodied in the designed artefacts.

In essence, the key concern with the Instrumental approach is that it is largely a perpetuation of existing HCI practice; office efficiency with aesthetics sprinkled on top. From the chorus of comments that commenced this chapter, it is evident that many researchers believe the everyday demands a deeper engagement with a phenomenological understanding of aesthetic experience.

2.9. Aesthetic Experience

In contrast to the mechanistic aesthetic models of the Functional approach, the aesthetics proposed in the experiential categories (Third Wave/Paradigm, Fifth Perspective, Experience-based) are informed by phenomenology and the Pragmatist philosophy of Dewey, Shusterman et al. In which aesthetic experience is viewed as uniquely personal, situated and emergent (Battarbee, 2003) and as being "as much a product of what the user brings to the situation as it is about the artefacts that participate in the experience" (Wright, McCarthy, & Meekison, 2004). As Petersen explains, "systems are not necessarily understood and used as designed. They are appropriated in use. Meaningfulness and aesthetic experiences emerge in use, they are not predefined" (Petersen et al., 2004).

The impossibility of predetermining experience does not negate the role of the designer, it simply means that instead of aiming to prescribe experience, they design *for* experience, and "invite people to actively participate in creating sense and meaning" (Petersen et al., 2004). In an attempt to "account for the active role that both designers and users have in negotiating the technology's consequent meaning and use" (Matthews et al., 2008), Matthews draws on Akrich's notions of "scripts" and "de-scription" (1992). In Akrich's model, the designer inscribes an artefact with their intentions for its use and meaning, making the artefact a kind of script. However, as actor, the user is not compelled to comply with the script and, depending on the technology and context, may instead define their own role outside the scope of the designer's projections. Matthews explains: "on the one hand, the object redefines the user's world by virtue of what it is; while, on the other hand, the object itself is redefined through being "displaced" into a setting that was not completely or accurately envisaged for it, and one in which it is never only used according to plan" (Matthews et al., 2008). Technology, use, actors and contexts all influence each other. Akrich describes this interrelation as "de-scription".

Like the script for a theatre or cinematic production, the script to which Akrich et al refer does not operate in isolation and will draw on conventions from the domain in which it exists. For example, in the case of a movie, its genre provides an audience with immediate ideas about the intentions of the producers. Geertz (1973) uses the term "commonsense" to refer to "ways of understanding the world that are not only shared but also known to be shared" (Wright, Wallace, & McCarthy, 2008). Wright describes these commonsense understandings as a form of literacy

that we are enculturated into. Returning to the example of the movie, a viewer can use their cultural literacy to sense how the producers intended a movie to be read, but their personal experiences may dictate a very different reading and response. Using the movie analogy, Wright et al discuss various common factors influencing a person's appropriation of a movie: "A person watching a movie for a second time may have different feelings about it and understand it differently the second time. Moreover, two people's experiences of the same movie will have some commonalities but there will also be differences because they bring different experiences to the movie. This involves not only different experiences of past films, but also different experiences of the day they have just had. For example, the quality of one person's felt experience of the film after a bad day in the office or in anticipation of a difficult day tomorrow may be entirely different to that of another person's after a relaxing day at home" (Wright et al., 2008).

Ultimately, the authors emphasise the dialogical nature of sense-making: "The gap between culturally received ways of making sense of a situation and how we choose to appropriate it is a dialogical one, a relation between self and community. Our commonsense understanding and our personal response coexist and their relation helps define our experience of the film" (Wright et al., 2008).

2.10. Critical Aesthetics

So far the discussion has largely focused on notions of determinism; whether it is possible or even desirable to engineer aesthetics to achieve a prescribed experiential outcome. Clearly, the proponents of experience-based aesthetics would argue that it is neither possible nor desirable to view aesthetics in such a reductive way. Instead they advocate richer, more complex and less predictable models which acknowledge that experience does not exist a priori and is significantly shaped by undesignable factors such as context and audience. An experiential and emergent view of aesthetics is a distinctive characteristic of the latest aesthetic waves/paradigms in HCI. It is also a key differentiator between the aesthetic waves/paradigms and functionalist approaches which attempt to reconcile aesthetic concerns within a conventional information processing model of interaction.

Another important (and differentiating) characteristic of the aesthetic-experience approaches is an understanding of aesthetics as addressing more than cosmetic beauty or pleasure, and encompassing modes of engagement beyond ease and affirmation. The notion of a critical aesthetic for everyday technologies is represented in Dunne's seminal work "Hertzian Tales" (2003), in which he introduces his concept of "Critical Design". As its title suggests, Critical Design takes a critical approach to everyday technologies and seeks to "explore a new role for the electronic object, one that facilitates more poetic modes of habitation" (Antony Dunne, 2005b). But in order to fulfil this ambition, Dunne argues that we must look beyond the goals of optimal performance and efficiency which dominate the design and production of everyday technologies. Just as it has been suggested that over-determining the use of an artefact can negatively impact users, reducing their ability to interpret and improvise, Dunne raises similar concerns regarding the negative implications of notions such as "ease-of-use" and "transparency".

2.11. Transparency

The concept of transparency refers to an ideal of technology use as seamless, frictionless and, ideally, completely unnoticeable to its operator. This is a Heideggerian notion of transparency, one in which a user's familiarity with a tool renders it invisible to them. (Being and Time, 1927) In this sense, transparency is not related to visibility, but to a more general awareness of a technology. Transparency has long been championed by the usability movement because of its associations with ease and efficiency. The title of Krug's famous web design text, "Don't Make Me Think!" (2000), perfectly encapsulates the commonly held belief that mental effort (or cognitive load in usability speak) should be reduced at all costs - a view which Donald Norman clearly shares: "The real problem with interface is that it is an interface. Interfaces get in the way. I don't want to focus my energies on interface. I want to focus on the job" (Norman, Why Interfaces Don't Work).

The prime strategy for achieving transparency has been through normalisation. Usability advocates such as Jacob Nielsen promote the strict adherence to common design conventions in order to establish and reinforce design standards, ideally to the extent that they become transparent to users: "If you are thinking

about how to design a certain page element, all you have to do is to look at the twenty most-visited sites on the Internet and see how they do it" (Nielsen, 1999).

To better appreciate the deeply conservative attitudes represented here, it is worth noting that they have emerged from the culture of desktop computing. Software design within the context of the Operating System (OS) sees designers working with uniform interface templates and complying to strict design guidelines regarding their implementation. And rather than lament the loss of creative opportunity, the genericism of the OS interface has been championed as a means to improve users' comprehension and ability. Norman writes: "When we have standardisation of our keyboard layouts, our input and output formats, our operating systems, our text editors and word processors, and the basic means of operating any program, then suddenly we will have a major breakthrough in usability." (1988)

There is no denying the sense of Norman's logic - familiar and predictable is easy, but the everyday context asks for more than fast and predictable.

2.12. Reflectivity

Transparency and the efficiencies it affords have obvious relevance in situations where efficient control is is of prime importance, but as computing technologies have matured and moved into the everyday context, the concept of transparency has attracted many critics, with none more vocal than Jay David Bolter. Bolter questions the dominance of the transparency agenda and HCI's resistance to aesthetic alternatives. In describing the divide between instrumental and aesthetic conceptions of computing, he writes "we are talking about two competing visions of digital technology: the pragmatic vision offered by HCI experts, for whom computers are information appliances, and the vision offered by digital artists and interaction designers. They are competing in the sense that each vision is an attempt to convince our culture at large" (Bolter & Gromala, 2003).

Bolter uses the terms "windows" and "mirrors" to describe these competing views. The "window" represents instrumental aims of aesthetic transparency, while the "mirror" represents a poetic and reflective aesthetic. Just as we appreciate the need for both prosaic and poetic forms of communication, Bolter asserts that

“every digital artifact oscillates between being transparent and reflective. The mistake that Nielsen and Norman make is to assume that the single goal of all design is to make the interface transparent, when in fact the goal is to establish an appropriate rhythm between being transparent and reflective” (Bolter & Gromala, 2003).

Bolter provides the following summary (Figure 7) of the two concepts:

Window	**Mirror**	
Information	Experience	(value)
Looking through	Looking at	(user response)
Transparent	Visible or reflective	(mode)
Single	Multiple	(representation)

Figure 7. Windows & Mirrors.
(Bolter, 2009 “Drop Shadow Talks” http://vimeo.com/12178023)

Bolter’s misgivings about transparency are not concerned with whether technology is discreet or conspicuous but are about the aesthetic constraints imposed by concepts such as transparency and ease-of-use. Dunne shares Bolter’s concerns; “Although transparency might improve efficiency and performance, it limits the potential richness of our engagement with the emerging electronic environment and encourages unthinking assimilation of the ideologies embedded in electronic objects.” (2005).

2.13. Mindful Disruption

To better illustrate how transparency promotes such “unthinking assimilation” we deviate from electronic media, to a case study in urban planning involving the innovative Dutch traffic engineer, Hans Monderman. The subject of this case study is the main intersection of the Dutch town of Drachten. Monderman’s task was to improve traffic flow and safety for thousands of cars, bicyclists and pedestrians[4]

[4] Over 22,000 cars per day. Millward. UK Telegraph. 4 Nov 2006

passing through the intersection on a daily basis. His solution was to convert the intersection into what he called a “shared space”. The shared space is remarkable for what it lacks: road signs, traffic lights, and other signals to direct the traffic. As Wired’s Tom McNichol writes, “To an approaching driver, the intersection is utterly ambiguous - and that's the point” (McNichol, 2004). With no signs or signals to indicate right of way, Monderman’s shared space requires pedestrians and drivers to communicate and negotiate in much the same way that ice skaters share an ice rink. Monderman explains: “Skaters work out things for themselves and it works wonderfully well. I am not an anarchist, but I don't like rules which are ineffective and street furniture tells people how to behave” (Millward, 2006). Monderman’s counterintuitive approach has worked successfully in various cities in Europe and England, where it has contributed to dramatic reductions in accidents and delays (“Naked streets are safer, say Tories”, London Times 2007-01-22, Ben Webster). As Monderman sees it, the road tells a story and a wide road with a lot of signs says to drivers “go ahead, don't worry, go as fast as you want, there's no need to pay attention to your surroundings. And that's a very dangerous message” (Millward, 2006).

Monderman’s road design is a truly fascinating example of challenging transparency with mindful disruption - intentionally disrupting people’s expectations in order to elicit a mindful reflection about their actions and engagement with an environment. Monderman demonstrates that the “street furniture” designed to make driving easier and safer also makes us blind to the experience of driving. Removing those aids and making driving more difficult brings the driving experience into focus and demands that we reconsider what has become a transparent interaction. Monderman’s shared space also makes us aware of the broader urban and social impact of our traffic-scapes; “that the way you build a road affects far more than the movement of vehicles. It determines how drivers behave on it, whether pedestrians feel safe to walk alongside it, what kinds of businesses and housing spring up along it” (McNichol, 2004).

Unlike Dunne’s arguments for richer poetic experience or Bolter’s calls for creative expression, Monderman’s motives for resisting transparency are far more pragmatic: he is simply trying to reduce congestion and improve safety, but in the process, his designs make a profound statement about our urban spaces; about the power relationships between people; about individual responsibility; and ultimately about transparency and mindfulness.

2.14. User Unfriendliness

Traffic design and everyday technologies are not interchangeable concepts, but the Monderman example has relevance to the current discussion of transparency and reflectivity. Monderman identified instances where making driving easier cultivated an unthinking driving culture which was culturally damaging and dangerous. In the context of the everyday, Dunne argues that designers face much the same problem. The concepts of ease and transparency and the "unthinking assimilation" they perpetuate become an impediment to richer experiential possibilities for everyday technologies. To move the everyday design agenda beyond the easy and affirming, Dunne advocates embracing a sense of "user-unfriendliness": "If user-friendliness characterises the relationship between the user and the optimal object, user-unfriendliness then, a form of gentle provocation, could characterise the post-optimal object" (2005, pg.35).

The strategy Dunne suggests is much like that employed by Monderman - removing the trappings which promote a thoughtless engagement (affirming user-friendliness) and instead giving users cause to reflect on their interaction with an artefact (critical user-unfriendliness). The sense of user-unfriendliness is also apparent in Niedderer's concept of the "performative object" (Niedderer, 2007). Like Dunne's "post-optimal object", the "performative object" aims to intentionally disrupt user expectations in order to incite mindfulness. While these approaches advocate a strategic disruption to user expectations, it should be noted that "user-unfriendliness does not have to mean user-hostility. Constructive user-unfriendliness already exists in poetry" (Dunne, 2005, pg.35).

In poetry for example, the author of a text does not aim to expedite delivery. Purpose and meaning emerge from experience through the act of reading the text. It is also commonly understood that the rewards of a text will require an investment of effort from the reader - complexity is potentially desirable and beneficial rather than an inconvenience for an audience. The subject matter of a text may also be "constructively unfriendly", presenting challenging or contentious ideas that are anything but easy. It is the potential of complex, provocative and poetic experiences with technology which Dunne hopes to bring to the realm of the everyday.

2.15. Affirmative Design

User-unfriendliness is a playfully provocative term which describes an understanding of aesthetic experience as meaningful, rather than easy or efficient. The term also aims to distance itself from functionalist approaches in which aesthetics are equated with cosmetic beauty and employed to improve usability outcomes, as represented by titles such as “What is beautiful is usable” (Tractinsky, Katz, & Ikar, 2000) and “Attractive Things Work Better” (Norman, 2003). We find a similarly reductive understanding of aesthetics in commercial contexts, where aesthetics are viewed as a way to attract and seduce consumers, and translate brand values into product form. Whether for instrumental or commercial aims, conceptions which limit aesthetics to the visually beautiful and affirming impede more ambitious ideas for the aesthetics of everyday technologies. “The challenge for those concerned with designing (for) experience in interactive products is shaping meaning through action with an artefact. This is a demanding task and the dominance of the commodity metaphor may be limiting rather than helping. This is because it understands experience as an ingredient to a product, an exotic, luxurious, slightly superfluous spice, a synonym for ‘‘emotions added to practicality’’ and higher prices. This does no justice to the concept” (Blythe, Hassenzahl, & Law, 2009).

Dunne puts it even more simply: “Design is assumed to only make things nice. It’s as though all designers have taken an unspoken Hippocratic oath. This limits and prevents us from fully engaging with and designing for the complexities of human nature which of course is not always nice…Nearly every other area of culture accepts people are complex, contradictory and even neurotic, but not design, we view people as obedient and predictable users and consumers” (Anthony Dunne & Raby, n.d.).

2.16. Summary

As a result of the transition of computing from the office workplace into the everyday cultural context, we find a growing field of research dedicated to understanding the implications of this metamorphosis. The literature of this emerging field shows a strong consensus that designing for the everyday represents a new paradigm for HCI, one that is less concerned with utility and

efficiency and more focused on aesthetic and experiential aspects of interacting with computing technology. However, despite the broad agreement regarding the significance of the everyday and the need to address aesthetics, there is less consensus regarding exactly what "aesthetics" entails, and how it should be addressed in the production of technology. At the heart of the issue is the difference between two fundamental approaches; instrumental and experience-based. Inspecting these positions more closely, it becomes apparent that the Instrumental approach largely preserves established HCI conceptions and practices, whereas experiential approaches propose ambitious new prospects for computing as a sophisticated aesthetic medium.

The Instrumental approach provides clear boundaries and directives for design but, by attempting to integrate aesthetics within existing information-processing models of interaction, it produces a reductive and impoverished version of aesthetics. By contrast, as McCarthy & Wright explain, adopting an aesthetic-experiential approach expands the experiential gamut of computing technology:

> *"This lens makes visible the potential for charm, enchantment, love, excitement, alienation, and irritation in our relations with technology. It also makes visible the quality of space-time in which we relate with technology and the sensory and sensuous character of the experience. Perhaps the most important aspect of experience that it makes visible is the potential for surprise, imagination, and creativity, which is immanent in the openness of each moment of experience. In demonstrating the potential of an aesthetic lens, we hope we have heightened sensibilities to the aesthetic and emotional in action and interaction" ("Technology as Experience", McCarthy & Wright).*

While an aesthetic-experiential approach provides us with a richer and truer account of a person's experience with an artefact, and immense creative opportunities for producers, it also presents some complex challenges. Perhaps most fundamentally it raises the question, how can we design for something as ephemeral as emergent personal experience? The simple answer to that question is that you can't, at least not in a deterministic sense, as Redstrom explains:

> *"To say that designers should refrain from over-determining use and users is not to say that ideas about use should not be part of our concern or even that it should not be our main concern, but that we need to acknowledge what it is we are designing and what falls outside of that…Actual use, as well as 'users' and their 'experiences', ultimately is not there for designers to design. Therefore, it*

does not seem to be a very good idea in general to define what design is about in such terms" (Redström, 2006).

Instead, the literature suggests designing *for* experience; prioritising aesthetic and experiential concerns while at the same time leaving room for an audience to interpret and adapt technology to their own ends. With the proposition of "de-scription" Akrich, Matthews et al demonstrate that it is possible to develop a model of design which is practicable but open to the situated and emergent nature of experience; a model which includes an audience in a dialogical rather than deterministic relationship with technology.

As indicated by McCarthy & Wright's comments above, the great ambition of the aesthetic-experience approach is to create everyday technologies which are more than work tools or domestic appliances, and which "provide new experiences of everyday life, new poetic dimensions" (Antony Dunne, 2005b). This ambition indicates a much larger scope for aesthetic experience beyond simplistic notions of visual beauty. Aesthetic experiences may be provocative, surprising, delightful, intriguing, beguiling and more. It is a conception of aesthetics like that which exists in other fields of creative production such as literature, art, cinema, theatre, and architecture. The proposition of a critical aesthetic promises much for both audiences and producers. However, the realisation of a more sophisticated aesthetic for everyday technologies is contingent upon challenging entrenched HCI usability concepts such as transparency and ease-of-use, as well as deeply engrained expectations that design must always be pleasant and affirming. As Petersen argues, "aesthetics is not only an adhesive making things attractive… Aesthetics cannot be sat aside as an "added value". Emerging in use; it is an integral part of the understanding of an interactive system" (Petersen et al., 2004). The aesthetics referred to here casts computing technology as more than useful, easy or affirming. It is an aesthetics with "the ability to surprise and provoke and to move the subject to a new insight of the world" (Petersen et al., 2004). It is a critical aesthetic which encourages skeptical sensitivity to the values and ideas the environment embodies (Antony Dunne, 2005b). It is an aesthetics which aims to bring new poetic dimensions to our everyday computing interactions.

Speculative Production

3.1. Introduction

The previous section demonstrated that the everyday introduces exciting opportunities to rethink computing, but also reveals the limits of traditional engineering approaches as we attempt to engage with richer concepts of aesthetics and experience. The crux of the problem is the emergent nature of aesthetic experience. Traditional approaches in HCI, which focus on the "fit" between user and technology (Hackos & Redish, 1998), are inappropriate as experience cannot be defined a priori. Designing for experience requires methods that are not overly reductive and deterministic. In simple terms, notions of aesthetic experience provide a much richer and truer account of a person's interaction with computing technology and present great creative opportunities for producers, but also raise significant conceptual and technical issues. But while the aesthetic priorities of the everyday represent a significant challenge to conventional HCI practice, we cannot assume that other communities of practice are facing the same crisis. As a direct consequence of the everyday transformation of computing, new cultures of software production have emerged, cultures without the historical baggage of HCI and with a dedication to software as an aesthetic medium of experience.

This section argues that, in addressing the aesthetic priorities of the everyday, we have much to gain from consideration of new creative cultures of practice and of innovative practitioners already engaging with the aesthetics and poetics of everyday computer technology. As an example of such a culture, it looks to the Flash scene of the early 2000s and considers if and how their products and practices can inform our current efforts to invent new poetic forms for the everyday.

3.2. Democratising Production

The personal computer has radically transformed a wide variety of disciplines through the digitisation and democratisation of production. Once exclusive disciplines such as sound engineering, film making, motion graphics, graphic design, industrial design and architecture have been laid open to novices through the availability of accessible computer hardware and production software (McKeich & Dziekan, 2005). The same scenario has occurred in relation to software production. Accessible software has allowed novice programmers from diverse disciplines to participate in software production (and more recently, in hardware production through platforms such as Arduino). In the late 1980s and early 1990s, applications such as HyperCard (1987) and Director (1988) enabled non-expert programmers to produce desktop multimedia for distribution on floppy disk or CD-Rom.

The 1990s saw the introduction of the Web with its simple HyperText Markup Language (HTML) which offered novices easy production and access to a potential world-wide audience. However, the creative scope of the early web was curtailed by its limited capabilities (simple hyper-linking, basic layout, crude graphics and fonts, and large disparities in browser display) and technical constraints (low bandwidth, small screen sizes). In 1996 Netscape introduced a browser plug-in architecture to work around the limitations of HTML and permit additional media types to be accessed in the browser. Early examples included Adobe's PDF plug-in (the initial inspiration for the development of Netscape's plug-in architecture), Macromedia's Shockwave plug-in for Director multimedia, and Apple's Quicktime plug-in for sound and video. But more than any other technology of the early web, it was Macromedia's Flash plug-in which allowed producers to explore the potential for sophisticated browser-based software.

3.3. Generation Flash

Into the browser space usually occupied by basic fonts, tiny images and crude layout, Flash allowed producers to deliver vibrant animation, graphics and sound. Initially, the Flash authoring application featured little more than a basic timeline for creating vector animations. But as it gained greater technical features, such as buttons for navigating between frames, producers found new and unlikely uses, as its creator Jon Gay explains: “At the outset, I didn’t foresee the importance of interactivity in Flash… One of the craziest Flash movies I remember was someone building a pinball game in Flash 2. The only interactivity in Flash 2 was the timeline, button objects and “go to frame” actions so the author of this game added every possible ball position to the movie. After seeing that, I instantly knew that our users, using only these simple tools, were creative and stubborn enough to do amazing things” (Simpson, 2008).

With the release of Flash 4 in 1999 and its improved ActionScript programming language, Flash offered accessible tools for the novice user and sophisticated programming potential for more adventurous developers. As a result it became the web production platform of choice for designers and artists in the late 1990s and early 2000s. Marking the rise of this creative community, the Design Museum in London hosted “Web Wizards: Designers Who Define the Web” (November 2001 - April 2002), an exhibition featuring the works of a small group of the most renowned creative innovators of the day: Yugo Nakamura, Joshua Davis, Daniel Brown, Tomato Interactive, James Paterson and Amit Pitaru. Apart from Brown (who worked with Director) the same names can be found in the 2003 text “New Masters of Flash” - another artefact of the Flash creative scene. There were also countless “How-To” texts[5] and a series of international conferences[6] featuring renowned Flash developers of the day.

The early Flash producers represented an important creative community who developed a remarkable vision of computing as a creative form. In his 2005 text for

---

[5] Amazon.com lists 1434 paperback and 112 hardback titles under the search “Macromedia Flash”.

[6] Notable conferences include FlashFoward, Flash on the Beach, FITC, Flashbelt. It is interesting to note that most of the conferences today have enlarged their scope to “creativity and technology” themes, and in some cases changed their titles accordingly - for example, Flash On The Beach is now Reasons To Be Creative, and FlashBelt has transformed into the Eyeo Festival.

the Whitney Biennial[7] entitled "Generation Flash"[8], Manovich describes the significance of the early Flash creative culture:

> *"Flash aesthetics exemplifies the cultural sensibility of a new generation. This generation does not care if their work is called art or design. This generation is no longer interested in "media critique" which preoccupied media artists of the last two decades; instead it is engaged in software critique. This generation writes its own software code to create their own cultural systems" (Manovich, 2005).*

While at the time they were celebrated by members of the Flash creative community and recognised critically, for many others their works were utterly perplexing, if not outright annoying. Usability advocate Jacob Nielsen's report entitled "Flash: 99% Bad" encapsulates his main criticisms of Flash media: "About 99% of the time, the presence of Flash on a website constitutes a usability disease" (Nielsen, 2000). A review of the Design Museum's 2001 exhibition Web Wizards expresses disdain for the experimental Flash works presented, stating that "the exhibition mainly deals with the young pretenders of interactive web design… There is no pretence at functionality here or indeed at any stage during the exhibition" (Betts, 2002). Describing the work of Joshua Davis, also featured in the exhibition, a BBC reviewer states that "the navigation can seem wilfully obscure" before conceding that if one is to "scratch the surface you might find that your patience is rewarded" (Walton, 2001).

The collection of criticisms provide a glimpse of the context in which the Flash practitioners operated. At the heart of the criticisms are expectations of what web media should be and how it should behave. The experimental Flash works failed on many counts to adhere to those expectations and for that reason were criticised. Again, Nielsen presents the argument for standardisation:

> *"Flash designers introduce their own nonstandard GUI [Graphic User Interface] controls. How many scrollbar designs do we need? Actually, we probably do need a new scrollbar design for online content; the current scrollbar was designed for office automation content that users wrote themselves. However, the specification of a new GUI widget is a major human-factors exercise. The current Macintosh and Windows scrollbars emerged after*

---

[7] www.whitneybiennial.com

[8] Manovich's definition of "Flash Generation" includes practitioners using technologies other than Flash. His focus is the aesthetic epitomised by Flash work of the period.

the world's best interaction designers worked for years testing numerous design alternatives. A new scrollbar designed over the weekend is likely to get many details wrong. And, even if the new design was workable, it would still reduce a site's overall usability because users would have to figure out how it worked. They know how to operate the standard widget. When you use standards, users can focus on content and their reasons for visiting your site. Deviate, and you reduce their feeling of environmental mastery" (Nielsen, 2000).

The Graphic User Interface (GUI) customisations which so frustrated Nielsen were a primary focus of many Flash works, as were the unconventional forms of control. The scrollbar example exemplifies what was a common aim of the Flash creative community - taking control of the interface as a creative medium. In pursuing this aim, the Flash producers were aware that their experimental works were likely to alienate general users, as Yugop explains in a 2001 interview:

"The current monocrafts site is, well, let's be honest, it's a million miles from the ideas of what someone like Jakob Nielsen says about usability. I can tell you, at least once every couple of days, I get mail saying "this site is so user unfriendly!" or "I've never seen such an unnavigable site" and sure, I completely agree! The techniques we're trying out at the moment, they're just not mature yet" ("YUGO NAKAMURA - interview," n.d.).

Bolter and Gromala see the tensions expressed here as indicative of a cultural turf war which they describe as a "great, almost religious difference" between "Structuralists" such as Nielsen, and "Designers" like those of the Flash scene:

"The Structuralists were separatists, believing that form and content could and should be separated. For them, a Web site is a pipe through which content flows to the user. They opposed elaborate visual design, which they thought impeded the flow of information. The Designers, on the other hand, were unitarians, who believed that form and content could not be separated… For the Designers, a Web page is an experience, and they wanted complete control over it" (Bolter & Gromala, 2003).

For the Flash scene, software was a creative medium to be celebrated rather than constrained on utilitarian terms. The Flash producers were not the first to conceive of software as an experiential form[9] but through the web they had

[9] See for example Laurel's 1991 text "Computers as Theatre".

unprecedented ability to propagate the view. Through the web, not only could new works rapidly reach large audiences but producers could also easily share their processes and techniques, making software production both public and social. This open and accessible forum instigated a massive influx of curious new software producers. The democratisation of production represented a radical departure from established production culture; interface design, once considered “a major human-factors exercise”, was being conducted by designers from diverse fields of practice in the public forum of the web.

Of course, not everyone was comfortable with software design and production being appropriated by unqualified practitioners. In his critique of Flash, Nielsen makes reference to “the world’s best interaction designers” whose work emerged from years of testing, in comparison with something a Flash producer “designed over a weekend” (Nielsen, 2000). Ironically, Nielsen’s criticism targets a revolutionary development in software production - that it could be made and published over a weekend. At no time previously had it been possible for an individual to produce and publish software so easily and to such a broad international audience. What Nielsen describes is the emergence of a form of production that was cheap, light and fast. By introducing a different scale of economy, web technologies such as Flash lowered the risks of production. With no requirement to manufacture and distribute a physical commercial product, developers could publish small software sketches[10] rather than fully-fledged applications, and do so confident that they could easily update the software as required. Principles such as these have today become staples in software development, as epitomised in the agile adage: “early and continuous delivery of valuable software”[11].

---

[10] In its day, the MONOcrafts site operated smoothly at fullscreen size (which was at that time revolutionary) and over 14 years after its initial publication in 1998, the site continues to perform seamlessly - it would be difficult to find even a handful of websites of similar vintage still functioning so competently.

[11] http://agilemanifesto.org/principles.html See also 37Signals “Getting Real” http://gettingreal.37signals.com/ch01_What_is_Getting_Real.php

3.4. YUGOP

The Flash culture was a product of the everyday, and despite the fact that the Flash application began its life as a simple animation tool, it evolved into a powerful production platform which supported new cultures and processes of software development. Flash production was lightweight, agile, highly accessible and, in combination with the web, supported a form of software development that was public, social and highly experimental. Much of the creative experimental work produced by the early Flash practitioners was the result of a playful exploration of the exciting possibilities afforded by the new web technologies of the day, and while such experiments could be regarded as frivolous, they ultimately had a significant impact on the form and function of contemporary computing. Recognising the value of this creative community, the aim of this chapter, as stated earlier, is to consider "new creative cultures of practice and innovative practitioners already engaging with the aesthetics and poetics of everyday computer technology", and so we look to the work and practice of one of the most remarkable Flash pioneers; Yugo Nakamura, commonly known as Yugop. Yugop's early Flash work (1998-2004) has been widely heralded and featured at le Center Pompidou (Paris), Kunstlerhaus (Vienna), Ginza Graphic Gallery (Tokyo), MOMA (New York) and the Design Museum (London). His 1998 site MONOcrafts 2.0 [12] was voted the most influential site of its day in a 2006 poll on the FWA website (Ford, 2006) and was also featured in "Digital Archaelogy" for 2011 Internet Week New York; an exhibition presenting a selection of the most important sites of the web's short history.

Yugop was instrumental in popularising the web folio as a platform for publishing speculative experimental interactive and generative Flash works. His MONOcrafts folio site was unlike typical websites of the day, not only because of its use of Flash, but because of the imagination and innovation that it brought to the web. Even from the site's initial opening moments it is clear that it is something out of the ordinary; it begins with a solid black void into which float eight soft spheres. The spheres configure into the form of a cube. As the site loads the cube rotates in response to the movement of the mouse, its elastic motion encouraging play. Once loaded, the statement "READY. CLICK TO START" appears and with one brief blip, the cube exits in a choreographed flourish of movement. Into the black void of the browser elements transition quickly and smoothly. A line appears and expands into

[12] http://yugop.com/ver2/

a monochrome photographic background image; boxes slide into view and lock together with a satisfying “tunk” sound; text dances across the screen “WELCOME TO MONO* CRAFTS”, “JUST MOVE YOUR MOUSE”. The cascading chain of boxes responds to the gestures of the cursor - its velocity and scale altering constantly and fluidly (Figure 8). On first use, the connection between the cursor movement and the sliding and scaling of the menu is not immediately apparent, but after some dedicated play, cause and effect are discerned. It is an immersive and somewhat vertiginous effect. Descriptions appear as the boxes pass under the cursor, and when clicked, the chain of boxes scales, slides and fades away; text items retreat from the screen, and the selected sketch transitions in…

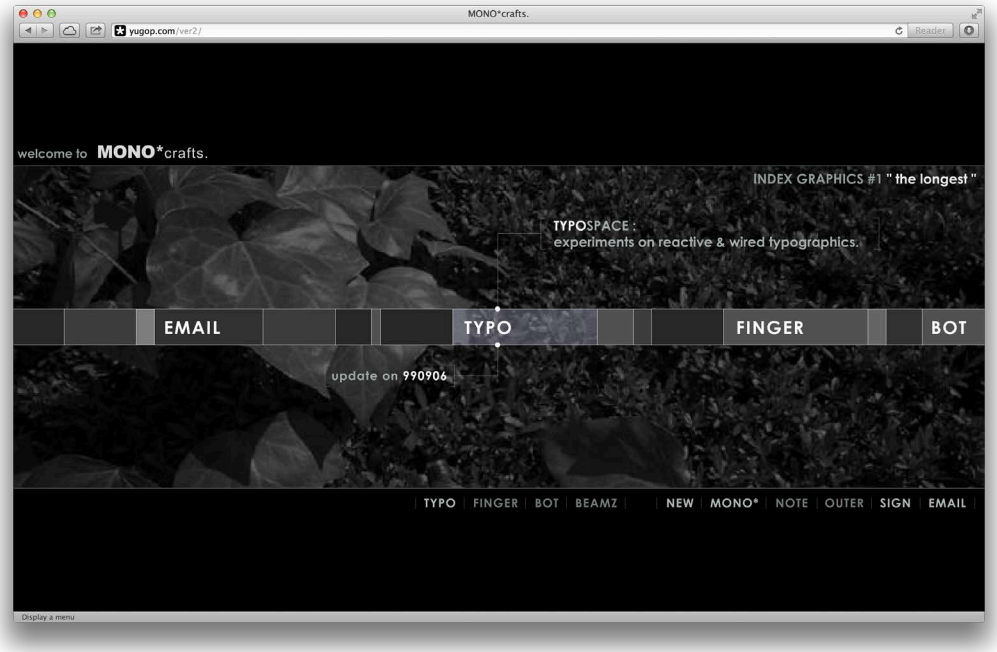


Figure 8. MONOcrafts. (Nakamura, 1998)

Even by today’s web standards, MONOcrafts possesses rare charm and originality. In contrast to the web of the early 2000s it offered a radical and remarkable sophistication for browser-based software, as Darrough attests; “[Yugop’s] site was among the first to effectively and convincingly instil “life” - flexibility, mutability, playfulness, depth, speed - into what was then a very static experience” (Darrough, 2010). The MONOcrafts folio presents an entirely novel view of web media. In most of the pieces within the site, simple interactions, typically taken for granted in conventional computing, are dramatically poeticised with elegant and often whimsical animation and interactivity. For example, in “TypoSpace” (Figure 9), what would be rendered as a simple text form in a

conventional web page is realised as a device for kinetic composition, transforming the banal task of text input into a delightful ludic experience. As the user types, their key strokes trigger simple blips of different tones, with each letter tumbling in a graceful arc to land in sequence at the top of the screen. When a message is complete it can be replayed - the rhythm of the typing dictating a choreography of sound and motion. Each message is a performance and the poetics of a message lies in the syncopation of its playback as much as the composition of its words.

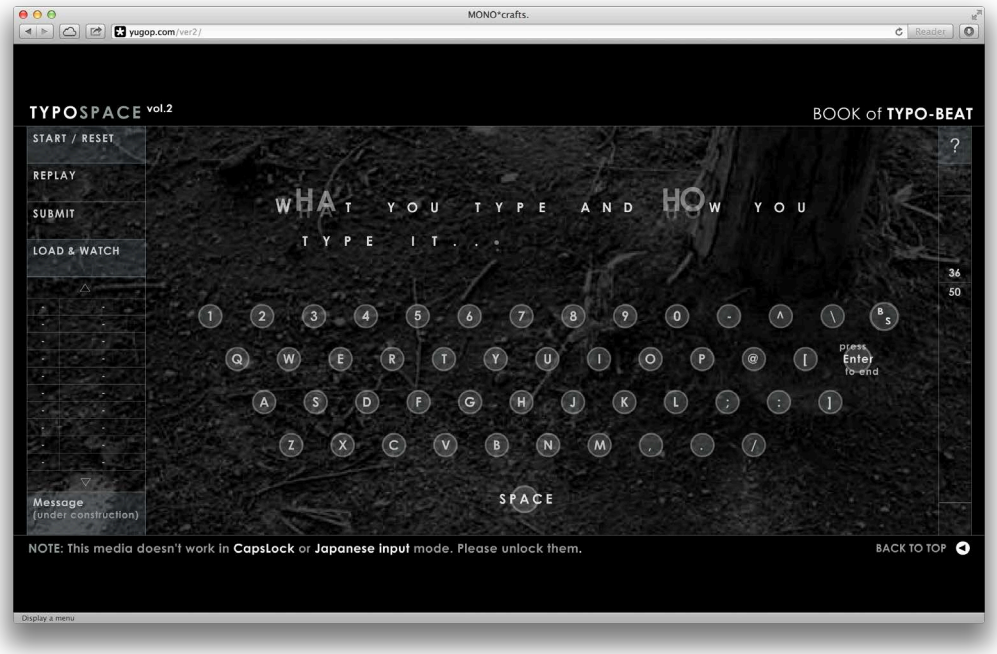


Figure 9. TypoSpace. (Nakamara, 1998)

3.5. Play

Yugop's approach is diametrically opposed to conventional views in which interface is designed to expedite interaction. Instead, he re-imagines mundane transactions as opportunities for playful enquiry, discovery and delight. Play is often associated with children and toys but in Yugop's work we see how play is used to positively frame characteristics that would be deemed undesirable on typical usability grounds (inefficiency, non-conformity, cognitive demand). More than only a means to have fun, the suggestion here is that play is integral to shaping negotiations between artefact and audience. By inviting an audience into playful exploration, Yugop resists conventions of invisibility and ease, instead demanding mindful attention. In simple terms, he asks his audience to invest effort in exchange

for a more enriching experience. In many respects his approach is indicative of Hallnas and Redstrom's concept of "slowness" which they see as resulting from engagement with "reflective technology". The authors explain that inciting a reflective and conscious use of technology is achieved by "amplifying the presence of things to make them into something more than just a silent tool for fast access to something else. This amplification is not just a matter of aesthetical surface, but concerns the whole thing as it is used. We do not talk about functionality and design, but about the complete expression of a thing as it appears in the given context" (Hallnäs & Redström, 2001).

3.6. Materiality & Presence

With regards to Yugop's work, Hallnas and Redstrom's notions of "slowness" and "reflective technology" have clear relevance but so too does their understanding of an aesthetic wholeness, a form of experiential gestalt or "presence". According to Hans Gumbrecht, presence is concerned with a state of being rather than an intellectual meaning ascribed to an artefact. Gumbrecht argues that the academic humanities have perpetuated a "meaning culture" in which exclusive focus is given to the identification and/or attribution of meaning (2004, pg.1). As Whitelaw explains, meaning here is "aligned with concepts such as "essence", "truth", "mind", "spirit" and "the immaterial". The human mind in this paradigm is *in* the world, but not *of* it" (Whitelaw, 2012). In contrast with this disembodied conception, presence cultures understand humans as bodies within a material cosmology. Accordingly, Gumbrecht "asks us to imagine a form of knowledge that is not exclusively conceptual, prior to, or not dependent on, interpretation" but which is revealed (Whitelaw, 2012). With regard to computer media, this notion of presence is distinct from instrumental purpose or semiotic meaning and is intimately connected to a "material aesthetic" (M. Bødker, 2004).

It is this material aesthetic which is so distinctive in Yugop's work and which sits outside the usual functional and semiotic foci of web software. For example, a work like "Jampack 01" (Figure 10) serves as an exemplar of a material and embodied aesthetic. As with so many of the MONOcrafts works, Jampack's functionality is unremarkable, and its visual aesthetic has been reduced to an absolute minimum; simple black discs of varying sizes set against a stark white background. What is remarkable is the depth of feeling that such simple elements can arouse: the initial

awe and delight as the discs begin descending into view, the sense of weight and friction as the discs bump and bounce into one another, the hypnotic metronomic pulse of a short audio “blip” signalling the removal of a disc and the descent of another. The piece operates with certainty but its meaning is mysterious; it is alien and yet utterly compelling.

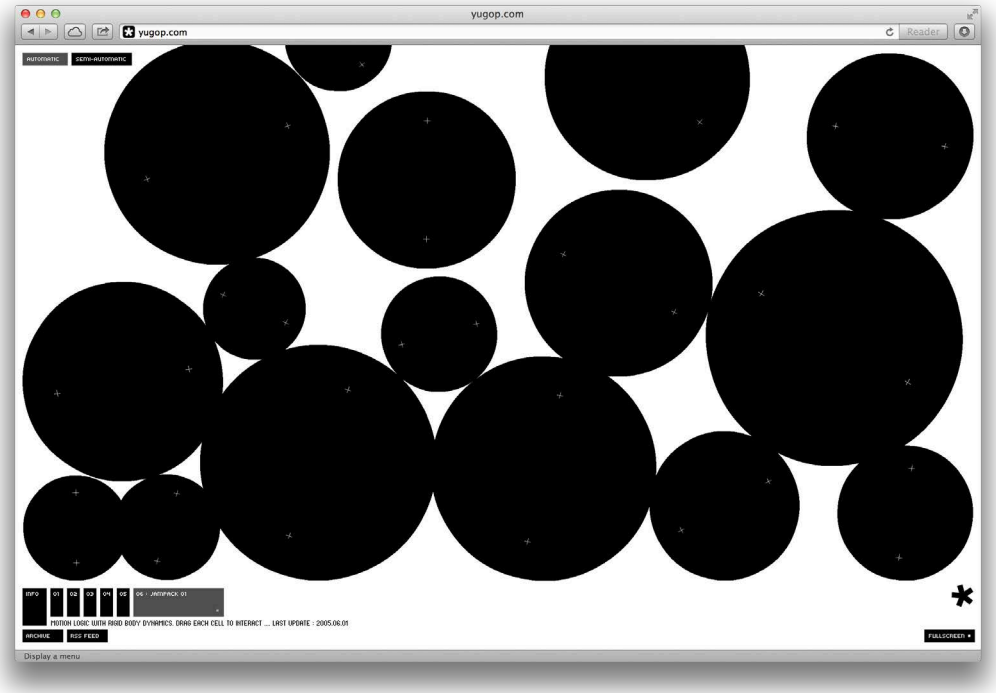


Figure 10. Jampack 01[13]. (Nakamara, 2004)

3.7. Computational Aesthetics

Yugop’s was a practice led by a material enquiry; an aesthetic investigation of computing inspired by his first encounter with the work of John Maeda;

> *“up until seeing John Maeda’s work I thought computer graphics simply meant creating pretty pictures using Illustrator or Photoshop. But what I saw in Maeda's work was very different. He was attempting to create something for the computer, not with the computer. It was about making something more natural to the computer’s environment. It’s about adapting to a world of calculations running on a massive scale. I think I did learn what it might mean to*

[13] http://yugop.com

express something from within the computer's own environment. That was the basic paradigm I learned from John Maeda-san" ("Yugo Nakamara Interview," 2001).

Figure 11. Dropclock.[14] (Nakamara, 2006)

There is a strong computational aesthetic evident in all of Yugop's work; systems, matrices, arrays, counters, timers, and voids of space are constants. His clock-based works[15] are perhaps one of the purest expressions of his interest in a fundamental computer pragmatic. Each of the clocks is a meditation on the primacy of time and the computer as a relentless counting machine. The essentialism of the clock makes it an ideal subject for focused study on the aesthetics and poetics of interface. Probably the most widely celebrated of the various clock works is "Dropclock" (2008) (Figure 11), a full-screen motion piece featuring monochrome Helvetica numerals falling into water in super-slow motion. The most minimal and utilitarian of interfaces, four digits representing the time are

[14] http://tha.jp/3 and http://frm.fm/en/gallery/2012/02/26/1208/

[15] Examples include Industrious clock, Clockblock, Dropclock, Clock of Light, DotClock for Framed. Clocks, timers, and numerical displays are also a common component within his web interfaces.

rendered with astonishing drama. It is a quintessential example of how familiar and forgotten artefacts of the everyday can be re-imagined and realised with unexpected poetry. “Industrious Clock”, a 2001 sketch recently adapted for Ars Electronica’s Vienna airport installation, has an entirely different character. Instead of the melancholic atmosphere of Drop Clock, Industrious Clock’s stop-motion animation aesthetic embodies a playful, frenetic energy.

> *“Like a conventional timepiece, its numbers change by the second, minute and hour, but in this case they’re drawn by hand, erased, redrawn and erased once again. In what might be called a handbook of temporal display, standardised, objectified timekeeping is juxtaposed to so-called felt time” (ArsElectronica, 2012).*

On a more technical note, the works’ use of computationally controlled video/animation is a distinctive Yugop feature. “Claygrid” for example, uses monochrome low-resolution stop-motion animation to render a pure mathematical algorithm with the character of the “Red and the Blue”[16]. The stop-motion footage of a clay ball gives the work a hand-crafted feel, but its behaviour is entirely generative; clay balls divide and merge, then divide again, working to the rules of their governing algorithm. In another experiment (Nakamura, 2010), stop-motion footage is mapped to the amplitude of a soundtrack; Lego bricks, a burning cigarette, an ice cube, Russian dolls, dice, and a wristwatch all act as a VU meter for the backing sound. For Yugop, video/animation is a dataset, randomly accessible and infinitely reconfigurable.

The notion of a computational aesthetic is evident in Yugop’s interest in the inherent array structures of content, whether in computer data or traditional forms such as stop-motion animation. Possibly the most comprehensive example of Yugop’s deep engagement with computational aesthetics is the extensive work he has completed as creative director for international clothing company, UNIQLO, which he describes as “Branding through algorithms” (San Francisco FITC 2010[17]). Yugop’s aim is to bring to life the systems behind the UNIQLO brand, like those that govern the vast ranges of clothing - colours, cuts, sizes, textures, seasons (FITCsf 2010). It was this fascination with the inherent algorithmic qualities of the

---

[16] http://www.youtube.com/user/MISSERISTUDIO
http://en.wikipedia.org/wiki/The_Red_and_the_Blue_(series)

[17] http://www.fitc.ca/events/about/?event=110 and http://vimeo.com/14397973

UNIQLO collection which drove the design of the site[18] for the company's 2006 rebranding. The primary feature of the site is its novel presentation of the massive UNIQLO clothing catalogue. The interface for the catalogue is based around a recursive interface in which the pixels of a feature image are converted into hundreds of miniature clothing items, each matched to the colour and hue of the original pixel. When clicked, the small image enlarges to become the new full-scale feature image. If clicked again, the image pixels are replaced by clothing items and the cycle continues. In Bolter's terms, it is a finely tuned transition between states of transparency and reflectivity; transforming the screen from catalogue photograph into an interactive interface, and back again. The same sentiment is clearly evident in the various UNIQLO websites, most of which incorporate a wonderful fusion of photography, video, and novel interactivity with computational compositing, filtering, ordering, juxtaposing and transitioning.

3.8. Array Aesthetics

A fascinating aspect of the UNIQLO work is that as creative director Yugop has been able to implement a computational aesthetic across the entire brand, from shop interiors to TV commercials, print graphics to packaging ("Yugo Nakamura - Creativity 50 - Creativity Online," 2008) (Nakamura, 2010). An unusual aspect of the brand development is that it is often driven by the web collateral. For example, the 2008 UT Loop campaign began with production of a promotional website in which a user acts as a kind of video DJ, composing a sequence of juxtaposed clips of various models wearing UT t-shirts and speaking random words. The site also allowed users to save and share their compositions, encouraging a viral level of promotion. Once the website was complete, the TV ad for the campaign was produced using the same video footage, music, editing and visual aesthetic of the website. Perhaps unsurprisingly, the TV ad possesses a strong programmatic energy; the juxtaposed editing feels as though it has been computationally generated.[19]

---

[18] http://tha.jp/87

[19] TV commercials for other companies such as IIDA and Brother also explore a computational aesthetic.

Manovich talks of computational concepts becoming a normal part of our vernacular in the same way that concepts from literature and cinema have previously been incorporated; bandwidth, resolution, storage, processing (Manovich, 2001). Elsewhere, Whitelaw (Whitelaw, 2011b) has attempted to document the incidence of what he terms “array aesthetics”, whereby the pragmatics of the computer screen are transferred into new physical contexts. Similarly, Bridle’s concept of the New Aesthetic, describes “an eruption of the digital into the physical” (Sterling, 2012). Through Yugop’s UNIQLO work we see how the same seepage is occurring at an aesthetic level; the metronomic timing and programatic composition of his computational works being rendered in video, print and even in interior design. Tha Ltd’s “FRAMED” (Figure 12) is another more literal example of bringing computational aesthetics into the everyday. In a conscious effort to move his experimental work away from the PC, Yugop instigated the development of FRAMED; a domestic appliance for computational artworks (Nakamura, 2010). Released in 2011, FRAMED is a large digital screen with a powerful inbuilt computer customised for displaying artworks published via the FRAMED GALLERY. The online gallery features a selection of Yugop’s MONOcrafts sketches reworked for the FRAME context and a variety of other generative works, both ambient and interactive.

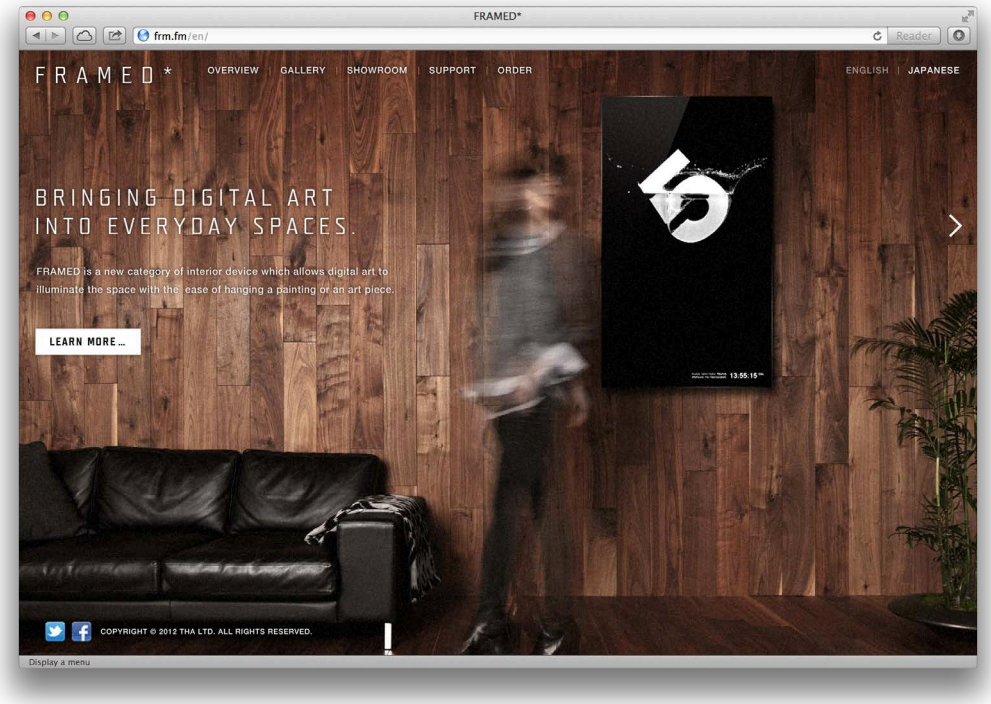


Figure 12. Framed. http://frm.fm/en/ (Nakamura, 2010)

3.9. Creative Practice

Our purpose in considering Yugop's work was to assume the perspective of a practitioner whose approach emerged from within the everyday. In the analysis so far we have discussed notions of play, reflection, slowness, presence, and computational aesthetics, and undoubtedly there are many more aesthetic lenses that could be applied to the work of Yugop. However, it is apparent that the aesthetics exemplified in Yugop's work are very much akin to the aesthetics of the third wave/paradigm described in the previous section; a Pragmatist aesthetic that is emergent, situated and embodied; an aesthetic that is not confined to any specific attribute but which is based on a holistic appraisal of a work. There is benefit in attributing tangible forms to these very elusive notions of aesthetics, but in doing so there is no suggestion that Yugop's works be considered definitive representations. Having identified an exemplary practitioner in the area of computer aesthetics, our interest is not only in the novel forms he has generated but in the processes by which they were produced. For Yugop, the realisation of innovative forms is not simply a process of exposition but the result of a more open approach to design and production. In describing the intent for his seminal MONOcrafts site, Yugop says "In the beginning, there was no concept for MONOcrafts. I was just playing with web technology. I was always obsessed with making things with my hands." This comment refers to a material approach to production; playing with materials to better understand their characteristics and the nature of the medium. Production is not treated as a means to an end but as a process of ideation and enquiry, with experimental sketches used to build knowledge of both the poetics and technicalities of the software medium. It is a reflective model of practice in which questions and propositions are formed *through* the act of production. That is not to say that there is no place for preconceptions of a work, but only that a creative process maintains an openness to the possibilities that emerge in the course of production. Echoing this sentiment, Yugop says: "I treasure a flash which hits me while at work or a shape which appears accidentally because of a bug. It won't be an interesting thing if every plan in my head is realised as scheduled. In a way, it's "luck"" (Fukushi, 2004).

3.10. Impact

From the discussion of Yugop's extensive body of work it is apparent that he, like many of the early Flash pioneers [20], has built a highly successful career on the back of his early experimental work. It was on the reputation of MONOcrafts that Yugop built his commercial practice, which was incorporated into Tha Ltd [21] in 2004. The MONOcrafts works have also served as prototypes, with various sketches growing into full-scale commercial works, and others being incorporated as elements within new contexts [22]. Even when not implemented in a literal fashion, the experimental works inform Yugop's commercial work, both in aesthetic character and programmatic technique. The experimental works have had significant influence on Yugop's professional life, but they have also had a broader cultural influence.

In a 2006 article entitled "Interaction as an aesthetic event", Manovich described the growing trend in the design of interfaces for computers and mobile devices in which interaction is "conceived as a carefully orchestrated experience, rather than just a means to an end. The interaction explicitly calls attention to itself. The interface engages the user in a kind of game. The user is asked to devote significant emotional, perceptual and cognitive resources to the very act of operating the device" (Manovich, 2006). Today, the interface aesthetics to which Manovich refers are evident, to varying degrees, in many of the graphic interfaces with which we interact on a daily basis - from computers to mobile devices, but also in facilities such as ATMs and public kiosks. They are also broadly evident on the web - both in the interfaces we see (transitions, animations, text-effects, fullscreen imagery, video, sound, 3D) and in the code that drives them (HTML, CSS, Javascript). Inclusion of "time" and "spatial transforms" as CSS styling options is a fascinating development which will be addressed in more detail elsewhere in this volume, but for our immediate discussion they serve as another clear indicator of the prevalence of "interaction as an aesthetic event".

While the conception of interface as an aesthetic event cannot be linked to any individual designer or community of practice, there is no doubt that Yugop and

---

[20] Josh Davis, Daniel Brown, James Paterson, and Amit Pitaru all forged successful art and design practices off the back of their early interactive endeavours.

[21] http://tha.jp

[22] For example Ars Electronica FutureLab commissioned a version of Yugop's 2004 "Industrious Clock" for their "ZeitRaum" installation for the Vienna Airport. Yugop repurposed "Jampack 01" in his "GridPlay" site for Intel.

others of the early Flash scene forecast the aesthetic agenda and were instrumental in popularising the techniques that characterise Manovich's "aesthetic interface". Consider the presentation and interactivity of MONOcrafts 2.0, published in 1998. In a domain dominated by simple static HTML pages, MONOcrafts transformed the browser window into an interactive cinematic frame with concepts of speed and motion, gestural interactive movement, time, space, and a wonderful sense of play. All of these characteristics are today staples of interface design for mobile and tablet apps, and increasingly the interfaces for desktop software[23]. Another more literal example connecting Yugop's early Flash work to Manovich's "aesthetic interface" is the "Infobar A01" touch phone. Produced by Yugop with Tha Ltd in 2011, the Infobar UI is based around a continuous scrolling band composed of a mosaic of user-customisable icons, photos, imagery and apps. The interface possesses a joyful character and features the playful interaction and fluid transitions that are signature Yugop.

The connections between Yugop's experimental and professional work are direct and easily demonstrated. Certainly there is no suggestion that Yugop or the Flash creative collective are the authors of the conventions that today drive the aesthetic trend to which Manovich refers, but their influence is clearly evident.

3.11. Speculative Cultural Prototypes

In their day, the experimental works of the early Flash producers went largely unnoticed by the general public, but their influence is evident in the form and function of contemporary software. In this respect, the Flash producers acted like a software avant-garde - working at the fringes and using accessible technologies to develop novel software forms. Of course, the Flash creative community is only one such culture, with many other notable examples operating in much the same way[24]. As with avant-garde cultures in other fields of creative production, we see that the software fringe acts as a sort of cultural incubator, generating concepts and conventions which are then appropriated and adapted for mainstream consumption.

---

[23] Mac OS 10.8 Mountain Lion attempted to port many conventions from the iOS to the desktop, while Windows 8 takes an even more ambitious attempt to bring interaction aesthetics from the mobile into the desktop setting.

[24] Other notable fringe software production cultures include: Processing, Cinder, OpenFrameworks, Unity, and the multitude of cultures operating under the broader web umbrella.

In Yugop's case we find a practitioner who has employed this same model within his own practice; building an experimental folio without commercial or utilitarian constraint, and then leveraging that work in his own commercial practice. Of course, his MONOcrafts experimental site was not conceived with such purposes in mind. Links between creative and commercial practice developed gradually, as Yugop explains: "as the web rapidly shifted towards commercialism, I began to seriously think about how to combine what I like (which is interactive expression) with design. Thus there are two sides – one is the initial enjoyment that I get from interactive expression, and the other is how to then construct it into reasonable design: currently I work trying to find a way between the two" (Fitzpatrick, 2008).

Having developed this model, even if by accident, it stands as a compelling example. The model exploits the properties of the Flash/web combination; lightweight Flash prototypes are easily produced and published with little risk, while the web offers a platform for public discourse through the work and around it. The model also benefits from the portability of code - the source code of successful prototypes can literally be reused, or easily adapted for new contexts. The proposition here is that the experimental works serve as speculative cultural prototypes, operating at the edges of existing cultural boundaries to develop new techniques and concepts. Producers harvest successful experimental concepts and techniques and incorporate them into commercial/applied works to be disseminated more widely. The following diagram (Figure 13) maps this process.

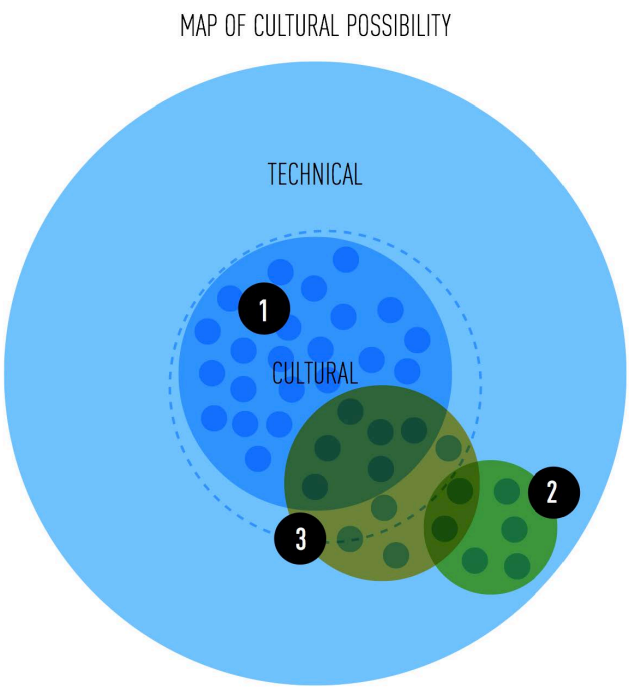


Figure 13. Map of Cultural Possibilities.

The large blue circle encompasses the range of technical possibilities. At its centre, is the range of what is culturally feasible. The small dark blue dots represent designs.

1. In the middle are those conceptions accepted by the mainstream majority - they are both technically and culturally feasible.

2. At the fringe are experimental works. While within technical scope, they are outside the bounds of mainstream cultural acceptance.

3. Concepts and technologies of the fringe (2) make their way into the mainstream either by being adapted to make them more culturally acceptable or through an expansion of mainstream cultural possibilities (the dotted line).

It is a common model of diffusion but the web mobilises it in specific ways. It is web technologies which allow sketches to be produced quickly and easily, and the web that acts as a platform for publication and social discourse. And as the web gains in ubiquity and technical sophistication, it is increasingly the final destination for software, rather than just the sketchpad; which means that porting a successful sketch into a new work requires no additional translation. [25]

[25] There is no suggestion that software art was invented with the web. Software artists such as Myron Krueger have practiced for decades but the barrier to participation in traditional software engineering is extreme; both on technical and cultural fronts. By contrast, the ubiquity and accessibility of the web incites broad participation in production, promotes speculative experimentation and enables distribution.

The model responds to the ways in which the concepts and techniques of the Flash experiments have been disseminated more broadly. However in recognising the potential value of creative works within a broader commercial setting, a tension is formed. The expectation of potential commercial value limits the creative freedom on which the speculative works depend. For the model to work, there is a need for independence from the normalising forces of mainstream commercial and cultural constraints.

3.12. Big Steps

The speculative model is particularly relevant in the context of the everyday where our problem, in simple terms, is that the solutions we seek do not yet exist and therefore sit outside the current boundary of cultural possibility. Diffusion theory (Rogers, 1962) shows us that adoption of new concepts is not as simple as dropping them into the market, and invariably new/alien ideas are rejected even if they represent advantages over familiar technologies. In many respects, the problem is akin to that of the “fitness landscape” (Kauffman, 1993). A fitness landscape is a model in which peaks represent solutions and the higher peaks better solutions. The difficulty is that to find new slopes requires moving away from local maxima, taking some big steps (and possibly travelling downhill) to cover ground quickly. The need for big steps is particularly significant in the domain of computer interface design where the culture is one of hill climbing in small steps; compliance and iterative improvement rather than dramatic innovation. This focus is evident in HCI’s almost exclusive dependence on user testing as a form of research validation, as noted in Greenberg et al’s paper entitled “Usability evaluation considered harmful (some of the time)” (Greenberg & Buxton, 2008). According to Norman, user testing and the user-centred ethnographic development methodologies favoured by HCI have been exceptional at incremental improvement (hill climbing in fitness terms), but have been unable to deliver significant innovation (the big steps that discover new peaks) (Norman, 2010). Advocates of user-centred ethnographic methodologies have supposed that if user methods can inform designers of how to improve an existing work, user analysis could also identify opportunities for invention. But as Norman concedes (Norman, 2010), user analysis has proven completely ineffective as a technique for generating the innovative big steps. The reason, Redstrom suggests, “is that although people

active in the domain one is designing for certainly possess knowledge about that domain, their ideas about future use are just as much a prediction as anyone else's" (Redström, 2006). Norman suggests that innovations arise most often as a result of the sheer will of inventors rather than in response to some existing user need. User need, he argues, forms after the fact in response to innovative propositions.

3.13. Propositions

Coming from the godfather of usability and user-centred design (UCD), Norman's statements are remarkable, but should not be read as an abandonment of UCD. His point is only that UCD methods do not cultivate innovation. In essence, he draws a line between practices of innovation and those for optimisation. Innovation he argues, is about propositions.

Verganti goes further, arguing that user-centred paradigms lead to design stagnation and perpetuate unsustainable economies (Verganti, 2010). He cites the example of the Toyota Prius, which when introduced in 1994 responded to no discernible market need. To the contrary, the market of 1994 was demanding production of large, gas-guzzling SUVs. Verganti writes that "the Prius was a proposal - a vision that came from a better understanding of the future evolution of the socio-cultural and economic scenario. Now, more than a decade after it was first launched, people like it, even if they did not ask for it when it was conceived. And thanks to its early start, Toyota is well ahead of its competitors" (Verganti, 2010). Like Norman, Verganti is adamant that innovation results from propositions which, like the Prius, may defy current market trends and user opinion.

The notion of design proposals is central to Verganti's concept of "design-driven innovation" (2009) (Verganti, 2010). Citing a group of highly successful Italian designer goods manufacturers (including Alessi, Flos, Kartell, Artemide), Verganti argues that their most renowned products were conceived speculatively rather than being based on any attempt to identify and satisfy an existing market need. Verganti suggests that the commercial success and longevity of the products lies in their "radical innovation of meaning". Citing the lighting of Artemide as an example, Verganti argues that their innovation was in proposing a "different and unexpected

meaning: a light that makes you feel better. This meaning, unsolicited, was what people were actually waiting for" (p.4, 2009).

Another area of design concerned with speculative proposals, albeit of an anti-commercial variety, is Critical Design. Critical Design is based upon the production of 'post-optimal objects', a form of conceptual product that seeks to provide "new experiences of everyday life, new poetic dimensions" (Antony Dunne, 2005b). Critical Design promotes the idea of design as a practice of imagining possible futures. As Dunne and Raby see it, design must "move beyond designing for the way things are now and begin to design for how things could be, imagining alternative possibilities and different ways of being, and giving tangible form to new values and priorities" (Dunne and Raby, 2010). Critical Design proposals need not be commercially viable, or even popular, as Antonelli writes:

> *"The Critical Design process does not immediately lead to useful objects, but rather to food for thought whose usefulness is revealed by its ability to help others prevent and direct future outcomes" (Antonelli, 2011).*

Design Fictions envisage a similarly speculative role for design:

> *"Design fiction is a way of exploring different approaches to making things, probing the material conclusions of your imagination, removing the usual constraints when designing for massive market commercialisation - the ones that people in blue shirts and yellow ties call "realistic." This is a different genre of design. Not realism, but a genre that is forward looking, beyond incremental and makes an effort to explore new kinds of social interaction rituals. As much as science fact tells you what is and is not possible, design fiction understands constraints differently. Design fiction is about creative provocation, raising questions, innovation, and exploration" (Bleecker, 2009)* [26].

It is worth stressing that the approaches outlined here are not anti-user, but anti-normalisation. The issue with user-centred design methods is not the users, but the way that user data is employed to normalise design outcomes. In their paper on "Aesthetics and Experience-Centred Design" (Wright et al., 2008), Wright et al cite the work and production practices of digital jeweller Jayne Wallace. Wallace conducts extensive consultation with clients commissioning her work, and then creatively responds to the information she collects which may include

[26] See also Bruce Sterling: Sterling, B. (2009). COVER STORY Design fiction. Interactions.

"personally meaningful stories, objects, relationships, and events" (Wright et al., 2008). The work that Wallace produces can be highly speculative, complex and poetic, but in her case, it is directly informed by her experiences with her client;

> *"User needs and requirements are not the focus of our enquiry. Rather the focus is an understanding of individuals, their concerns, desires, aspirations, values, and experiences. The relation between designer and "user" is not an objective one in which the designer stands outside of the user's situation. Instead, it is one in which the designer and user are in mutually influencing, empathic dialog" (Wright et al., 2008).*

3.14. Design as Discourse

Despite the difference in commercial/conceptual focus, all of these propositional approaches share a view of design as discourse, and the role of designers as proactive rather than reactive; leading the discourse with speculative propositions rather than assimilating with the status quo. Whether acknowledged explicitly or not, conventional approaches to software production have also conducted design as discourse. In web design for example, a central role of the designer is locating their work within a particular discourse in order to determine the most suitable design conventions for the given context. "Suitable" here means normal, and typically involves designers looking to the leading sites of the web to determine the best design conventions to adopt (Nielsen, 1999) (Krug, 2000).

An obvious difference of the speculative creative production epitomised by Yugop is that its engagement with the discourse of design is more questioning and critical. Rather than emulating conventional norms Yugop's works seek to question, provoke and entertain. It is in this critical regard that the speculative works of Yugop and the early Flash scene were so significant. Their unsolicited experiments initiated an important discourse around web interfaces as a creative cultural form. Through the subversion of dominant conventions and the proposition of novel forms, the Flash designers made original and important contributions. Their work served as a form of "material argument", a concept which Bardzell explains;

> *"the designer, in implementing the particulars of a design, implicitly states that this is how it should be done, and to make this (material) argument, the*

designer necessarily references and critiques earlier designs in similar categories" (Bardzell, 2008).

Of course, forming design propositions or arguments is an act of will which necessarily implies the presence of an author. This authorial role for the designer represents another split with conventional HCI practices which favour the designer as anonymous facilitator within a user-focused process of development and validation (Tedesco & Tullis, 2006). Authorship has been viewed suspiciously by HCI because it implies subjective decision-making whereas user validation methods imply scientific rigour [27]. Conversely, the authorial approaches discussed above are highly critical of design and validation methods which entrench norms and limit the creative, critical and aesthetic scope of design. We can simplify the argument between speculative and user-centred methods to be one of choice, ultimately governed by context; user-focused methods are ideal for hill climbing, whereas speculative approaches are better suited to invention and innovation.

3.15. Summary

At the outset, it was proposed that "we have much to gain from consideration of new creative cultures of practice and of innovative practitioners already engaging with the aesthetics and poetics of everyday computer technology". The early experimental Flash scene was proposed as a culture not only concerned with production for the everyday, but also as a culture that has emerged from the everyday - being enabled by the web and accessible production tools such as Flash. The question is, what does the early Flash scene show us about software production for the everyday?

The work and practices of the early Flash producers have given us much to consider with relation to computing in the everyday, but perhaps most fundamentally they demonstrated that aesthetic production is about questions and propositions, more than established proofs. Liberated of the history and obligations of traditional HCI practices, the Flash producers pursed software as an experiential medium and production as a creative practice. Accessible production tools in

[27] "While we may use scientific methods to do our evaluation, this does not necessarily mean we are always doing effective science." (Greenberg & Buxton, 2008)

combination with the web allowed the Flash producers to use the web as a kind of software incubator; a context where they could share their speculative experimental works. Their experiments focused on cultural rather than technical innovation, and unlike traditional development which focuses on “fit” and optimisation, their speculative approach was concerned with invention, provocation, and expanding the bounds of cultural possibility. From observing their approach to production, and the social and technical conditions from which it emerged, we gain a strong sense of the importance of speculative experimental practice and the need to cultivate spaces where questions and propositions can be formed in isolation from normalising forces. Ultimately, the early Flash producers focused our attention on the aesthetic significance of our daily interactions with technology, and flagged the possibilities for creating provocative and poetic forms for the everyday.

> *“Today the language of cultural interfaces is in its early stage, as was the language of cinema a hundred years ago. We don't know what the final result will be … But there is one thing we can be sure of. We are witnessing the emergence of a new cultural meta-language, something which will be at least as significant as the printed word and cinema before it” (Manovich, 2001).*

Methodology

4.1. Introduction

In the second chapter, the theory and research of the HCI field provides a formal context of scholarship concerned with the conceptions of computing in the everyday cultural context. In the third chapter, a creative community of practice provides a designer's view on the production of poetic computational forms for the everyday. The second chapter provides a theoretical context to which this research can refer and to which it can contribute. The third chapter demonstrates a methodology of creative practice that can be employed within this project and also indicates ways in which creative practice can influence the broader discourse of design.

Chapters two and three provide valuable context and reference for this research project but also give us an insight into the existing gaps in knowledge, as well as the ways in which new research can contribute. In many respects both the formal and creative endeavours demonstrate the same things; that the everyday provides opportunities for designers to engage with poetic forms of representation and interaction but that the realisation of that potential is complicated by the instrumental culture of computing. The gaps here are many. The task is

conceptual, technical and cultural. On a conceptual level, we need to adapt our fundamental conceptions of what computing is and develop theoretical frameworks that are attuned to the aesthetic priorities of the everyday. On a technical level, we need to begin filling in the details of how grand aspirations for poetic computational technologies can be realised in material form, in the here and now. On a cultural front, we have to develop both the awareness of producers and audience in order to elevate computing into a creative and critical medium.

In summary, it is clear that aesthetic production for the everyday is not a "problem" to be solved in a deterministic sense. It is a situation which invites contribution and which we should endeavour to better understand but it is not something that will be fixed. That said, there is great opportunity to improve our knowledge of software poetics and to contribute in meaningful ways on the various fronts listed above.

4.2. Method

The project draws inspiration from creative practitioners working in the field of interface aesthetics, but also borrows from their methodologies. As was outlined in chapter three, practice is framed as a form of material enquiry rather than an expository process of answering set questions or realising preconceived forms. It begins with questions and directions but remains open to the ideas and opportunities that emerge in the course of production (Frayling, 1997). This is very much a conception of production as an emergent activity. Drawing on Margolin, Frayling and Strand, among others, Dallow provides a succinct summary of research through practice as being

> *"where the process of making, producing or creating cultural presentations, and the exploration and transformation which occurs in the process, is taken as an act of research itself, where knowledge is gained in the creative act, and can be directly attributable to the creative process. This 'emergent' knowledge may be abstract or cognitive (theoretical) knowledge, as well as practical knowledge gained in and from the 'doing'" (Dallow, 2003).*

Following this line of thought, the project employs a reflective practice-based (Schön, 1983) form of enquiry in which interface poetics are explored through

critical reflection of creative production. This process produces research *in* the form and *about* the form (Strand, 1998), with the creative work contributing in a material sense to the design discourse and also acting as a catalyst for critical reflection and theoretical analysis.

4.3. The Project

For software, the everyday introduces creative opportunities and aspirations but at a practical level there is little detail of how such visions may be realised. Even where exemplars can be found, the solutions offered are not persistent truths - the everyday is a cultural context which is in constant flux, and as such we must continue to seek new solutions and understanding. Of particular interest to this project is the notion of poeticising the software representations of volatile dynamic cultural data sources. Much of the reference for aesthetics and poetics comes from art and literature that simply have not had to address the sorts of dynamics which characterise the digital networked media context. In addressing aesthetics in the computational setting, it is not simply a matter of transferring knowledge from one domain to another and patching over any incongruities. Exploring poetics in the computer context requires engaging with the pragmatics of the computer medium. The creative works produced within this project centre around the development of web-based speculative prototypes representing cultural data from the social media service Twitter; a context which provides dynamic complexity and the cultural scope for poetic realisation.

4.4. The Web

Why web based prototypes? The web has been the main vehicle in the everyday transformation of computing. It changes the way we use computers (why, where and how), and the production of computer media. Since its inception the web has been a fertile space for development. Its advantage lays not only in its easy and accessible set of coding languages but in its ability to serve as a platform for the publication/sharing of works. The web allows speculative production cultures to congregate and conduct their particular discourses but also fosters diffusion of innovative ideas and techniques from fringe cultures into larger market

contexts. In addition to being an ideal platform for speculative experimental works the web is the origin of the cultural data which serves as the source material for this project. On all counts, the web is an ideal development environment and publication platform.

4.5. Twitter

Twitter is a popular micro-blogging platform which enables its users to post short “status updates” and to receive an aggregated listing of statuses from users they elect to “follow”. Unlike legacy communication technology such as email, which was conceived in the office era of computing (as indicated by the term “mail” and reference to envelopes etc.), Twitter is a product of the everyday. Conceived in 2006 (Figure 14), it’s purpose was to enable networks of friends to keep abreast of each other in a very light, informal way. The proposition on the original Twitter site stated: “A global community of friends and strangers answering one simple question: What are you doing?”. A big part of its novelty was its resistance to the forces that were driving the internet/web of 2006-7. Where the order of the day was the increasing speed and power afforded by broadband networks and modern computers, Twitter offered a service that was pointedly minimal and lo-fidelity.

Figure 14. Twitter in 2006. (Source: http://www.webgeekly.com/)

It has been selected as a source for this project for two main reasons: culturally, it is a communication born of the everyday and for the everyday; and technically, its network dynamics introduce important additional complexities that must be reconciled with aesthetic ambitions.

The following description provides an overview of common terminology and modes of operation. Mashable's Brandon Smith provides definitions of key Twitter terms:

Tweet: A 140-character message.

Retweet (RT): Re-sharing or giving credit to someone else's tweet.

Feed: The stream of tweets you see on your homepage. It's comprised of updates from users you follow.

Handle: Your username.

Mention (@): A way to reference another user by his username in a tweet (e.g. @mashable). Users are notified when @mentioned. It's a way to conduct discussions with other users in a public realm.

Direct Message (DM): A private, 140-character message between two people. You may only DM a user who follows you.

Hashtag (#): A way to denote a topic of conversation or participate in a larger linked discussion (e.g. #AmericanIdol, #Obama). A hashtag is a discovery tool that allows others to find your tweets, based on topics. You can also click on a hashtag to see all the tweets that mention it in real time - even from people you don't follow.

(Smith, 2012)

(Source: http://mashable.com/2012/06/05/twitter-for-beginners/)

The fundamental usage of Twitter sees a user posting status updates and reading their timeline, which aggregates the status updates of all the users they have elected to follow. Despite its seemingly simple exterior, Twitter is actually a sophisticated system, with complex network structures arising from its system of friends and followers. Take for example "retweeting" which refers to the act of re-posting a status of another user. When a user retweets a status it is visible to all of their followers, irrespective of whether they follow the person being retweeted. In this way, a user is able to introduce their followers to users outside their existing networks. Retweets have the potential to be disseminated around the world

through these channels of retweeting, and these models of diffusion have inspired some interesting visualisations [28].

The complexities of the Twitter network are evidenced in the data that it supplies via its Application Programming Interface (API) which serves the source data for all of the creative works in this project. The following figure (Figure 15) is an extract from the data representing a user's home timeline.


```
1.  [
2.    {
3.      "coordinates": null,
4.      "truncated": false,
5.      "created_at": "Tue Aug 28 21:16:23 +0000 2012",
6.      "favorited": false,
7.      "id_str": "240558470661799936",
8.      "in_reply_to_user_id_str": null,
9.      "entities": {
10.       "urls": [
11.
12.       ],
13.       "hashtags": [
14.
15.       ],
16.       "user_mentions": [
17.
18.       ]
19.     },
20.     "text": "just another test",
21.     "contributors": null,
22.     "id": 240558470661799936,
23.     "retweet_count": 0,
24.     "in_reply_to_status_id_str": null,
25.     "geo": null,
26.     "retweeted": false,
27.     "in_reply_to_user_id": null,
28.     "place": null,
29.     "source": "<a href=\"http://realitytechnicians.com\" rel=\"nofollow\">OAuth Dancer
    Reborn</a>",
30.     "user": {
31.       "name": "OAuth Dancer",
32.       "profile_sidebar_fill_color": "DDEEF6",
33.       "profile_background_tile": true,
34.       "profile_sidebar_border_color": "C0DEED",
35.       "profile_image_url": "http://a0.twimg.com/profile_images/730275945/oauth-
    dancer_normal.jpg",
36.       "created_at": "Wed Mar 03 19:37:35 +0000 2010",
37.       "location": "San Francisco, CA",
38.       "follow_request_sent": false,
39.       "id_str": "119476949",
40.       "is_translator": false,
41.       "profile_link_color": "0084B4",
42.       "entities": {
43.         "url": {
44.           "urls": [
45.             {
46.               "expanded_url": null,
```


Figure 15. Twitter API data excerpt
(source: https://dev.twitter.com/docs/api/1.1/get/statuses/home_timeline)

[28] Twitter visualisations:
http://tweetreach.com/labs/rings#!/97357461811507202
http://www.technologyreview.com/view/415091/a-retweet-revolution-visualized/
http://moritz.stefaner.eu/projects/revisit-twitter-visualization/

Aspects and implications of data will be discussed in further detail elsewhere in this thesis. For our immediate discussion the example is shown as an indication of how Twitter's network structures come to bear in the structures of its data.

4.6. Summary

In a bid to provide a practice-based perspective on the current discourse in aesthetics for everyday computing, this research centres around the creative development of a series of novel realisations of dynamic cultural data generated by the Twitter social media service. As a web-based service that has emerged from the everyday shift in computing, Twitter serves as an ideal subject on both cultural and technical fronts with its informal personal communication mode inviting poetic representation, and its network dynamics bringing a challenging level of complexity to the aesthetic aims. Chapter 5 documents the creative works and includes discussion of how the works and their process of production have contributed to the research project and knowledge generation.

In addition to producing novel material forms, the practice-based methodology employs critical reflection on the creative production as a way to generate intimate technical and conceptual understanding of software aesthetics at the production coalface. To this end, chapters 6 and 7 present the formal theoretical output resulting from the reflective production process.

The Work

5.1. Introduction

As has already been extensively outlined in chapter two, the everyday cultural context allows, and some would argue that it demands, greater understanding of computing as a medium of aesthetic experience. As Dunne points out, when optimal levels of performance have been achieved, focus shifts to the post-optimal; to the problem of poeticising our everyday engagements with technology (Anthony Dunne, 2005a). As a result we need specific details of what poetics entails in different contexts and how such poetics might be achieved. The creative works completed for this project are an effort to contribute to these aims. The works seek to explore software as a creative form, as a vehicle for reflection and engagement as much as information and control. They are experimental propositions and each engages with the conceptual and technical aspects particular to its framing of the Twitter source.

In the course of their production and their final form, the works engage with some fundamental questions such as: What does it mean to poeticise software? How can volatile, dynamic information systems be poeticised? What is the significance of the computational context? The answers to such questions cannot

be categorical and the creative works are not proposed as conclusive results but as contributions to a broader discourse regarding the form and function of our everyday technologies. A benefit of the creative practice-based approach adopted in this project is that, in response to high level questions of aesthetics and poetics, it develops a perspective grounded in the specificities of craft. As a result, each of the creative works offers insight into contextualised practical aspects associated with concerns regarding aesthetics and poetics.

Many more works were completed in the course of the project, but not all can or need to be represented here. In most cases, ideas and techniques from the excluded works have been incorporated within the final selection of pieces. The descriptions that follow provide a general overview of each work and details of novel aspects, whether conceptual, aesthetic or technical.

5.2. Twitter Motion

http://gravitron.com.au/motion

Twitter Motion (Figure 16) is one of the earliest pieces completed as part of this project. In it, a person's home Timeline is realised as a motion graphics sequence. The most immediately striking feature of the piece is the scale and movement of the type - the proximity and speed of the letterforms makes the browser feel like a car or train window. The effect of the motion, rotating and scaling can be quite disorienting, which is precisely the point of the work. The dynamic composition of the text is reminiscent of motion sequences for movies or video clips and its association with a social media feed is incongruous, but it is an amusing prospect to see mundane tweets rendered with such intensity and drama.

In addition to the general novelty of rendering tweets in motion the interest of the work emerges from the deconstruction and reassembly of the original textual message. Breaking the text into discreet components introduces a particular phrasing unintended by the Tweet author. This new imposed phrasing is then performed to the viewer and when the combination of phrasing and choreography are complimentary, the effect can be quite amusing and occasionally poetic. Motion is a study in rules - rules for text dissection, rules for text display.

Twitter Motion processes the text algorithmically rather than semantically, and uses some simple string analysis to determine basic word and sentence structures. Like DaDa poetry[29] it uses randomness to generate novelty. Text here is treated as an object to be dissected and rearranged as is the type; the letterforms are abstract blocks within a larger architecture of type. To allow the type to be used in this compositional way, I created an object for each letter, which means there are no conventional textfields in the piece. The computational treatment of type is a recurrent feature in my creative works and so too are the associated technical issues. These difficulties speak of a fundamental tension between conventions and practices developed for printing press technology and the new demands of the computational context.

---

[29] Take a newspaper. Take some scissors. Choose from this paper an article the length you want to make your poem. Cut out the article. Next carefully cut out each of the words that make up this article and put them all in a bag. Shake gently. Next take out each cutting one after the other. Copy conscientiously in the order in which they left the bag. The poem will resemble you. And there you are - an infinitely original author of charming sensibility, even though unappreciated by the vulgar herd.

"Looking at Dada" edited by Sarah Ganz Blythe, Edward D. Powers, 2006, Museum of Modern Art (New York, N.Y.)

Figure 16. Twitter Motion. Released October 2008.

Twitter Motion was featured by technology author Greg J Smith not long after its launch in 2008 (http://serialconsign.com/node/247 [30]):

> *"An example of what might be described as "transformational" design whimsy is at play in twitterMotion, a type focused viewer for examining recent activity for a twitter account. This application was obviously not developed as a full UI for twitter, but instead takes the content of the popular web service and turns it into the basis of a type-driven, animated adventure. It is fun, dizzying and completely alien compared to the scores of identical twitter applications out there. Since there is no input a user can only sit back and watch the content fly by reverse chronologically."* (Smith, 2012)

The piece was also utilised in various creative works by multimedia artist Mez Breeze, and included in her work at the Laguna Art Museum [31] California and Alternator Gallery [32] Canada in 2009. In 2009 I exhibited the work as part of the "This Is Not Art" festival in Newcastle.

Motion provided early and valuable insight regarding the relationship between theory and practice. The temptation in theoretical accounts of creative practice is to describe the process in convenient linear models in which practical works are cast as mere illustrations of preconceived concepts. Motion served as an early reminder that the creative process is not conveniently linear but iterative, dialogical and messy. Perhaps most importantly, the work emphasised the importance of questions rather than proofs. While Motion was initiated with particular questions in mind, its production raised additional questions which shaped not only its development but the trajectory of theoretical enquiry and ongoing creative production. For example, while it is largely focussed on motion graphic conventions and a cinematic approach to representation, its development raised many questions about fundamental graphic design concepts and practices, particularly with relation to typography and computational type setting. It also focussed my attention on the concept of whimsy as a device for critical resistance to the utilitarianism that is so dominant in HCI literature and web production cultures. Both of these themes are significant to the other creative works and the theoretical component of the research.

---

[30] Archived version: http://web.archive.org/web/20100410234859/http://serialconsign.com/node/247

[31] http://lagunaartmuseum.org

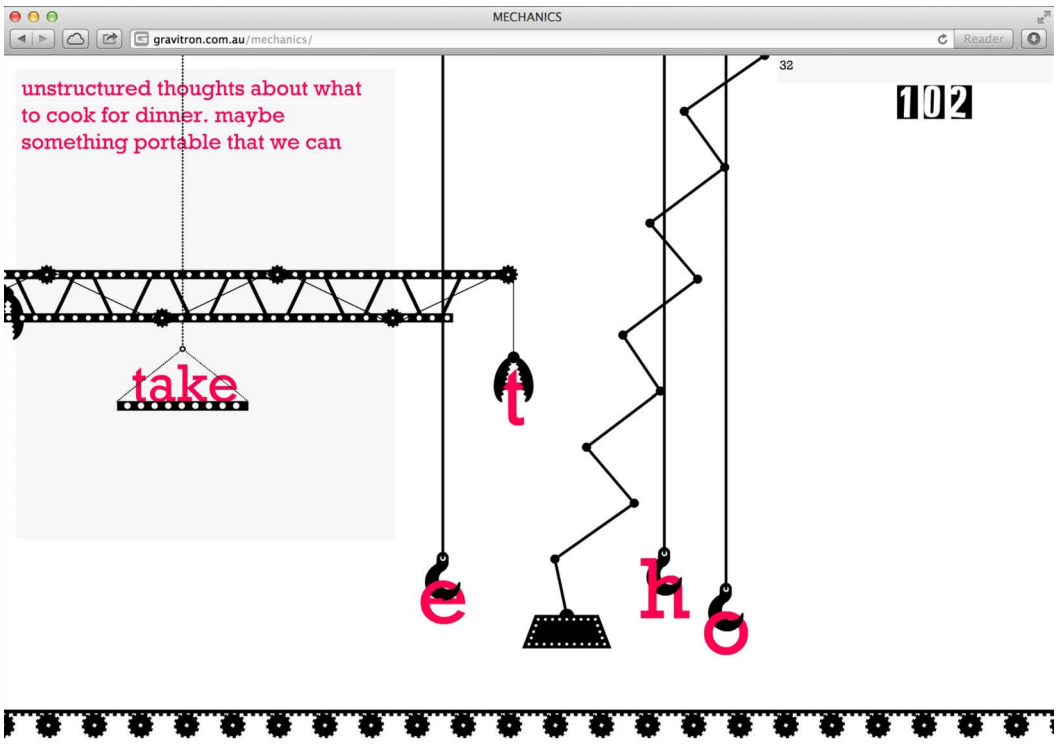
[32] http://alternatorcentre.com/home/

5.3. Mechanics

http://gravitron.com.au/mechanics

Manovich declares that all software is ultimately a performance (Manovich, 2008) and in this piece I exploit that fact to dramatic effect. Of all the creative works discussed here, Mechanics (Figure 17) most overtly encapsulates what I refer to as “critical whimsy”; using humour and nonsense as a critical resistance to the sobriety of the computational context. Obviously, play is also a central concern of this piece. It turns what is a very simple task, typing a Tweet message, into a remarkably elaborate performance for no other reason than the joy of participating in the spectacle.

Mechanics sat in stasis since late 2008 due to the mass abandonment of Flash incited by the rise of Apple’s iOS devices. There is no doubt that Flash is in demise, which is a loss to designers because it has been a uniquely useful development tool. In search of a replacement technology for the Mechanics piece I have have done some preliminary work using Scalable Vector Graphics (SVG) format, which is widely compatible with mobile and desktop browsers, but it is yet to be seen if it will be able to deliver adequate performance in animating the large numbers of complex vector objects in the Mechanics piece.



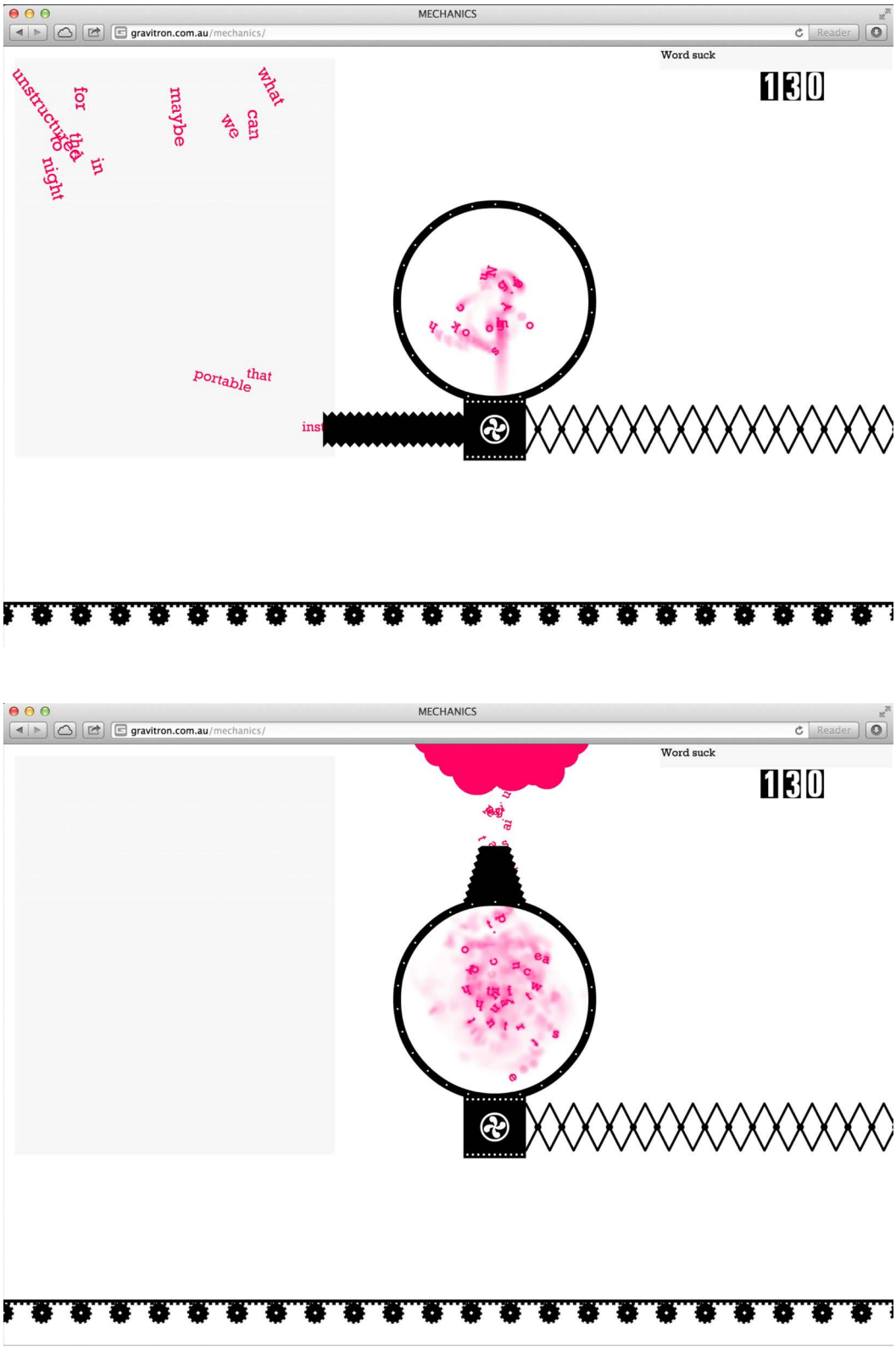


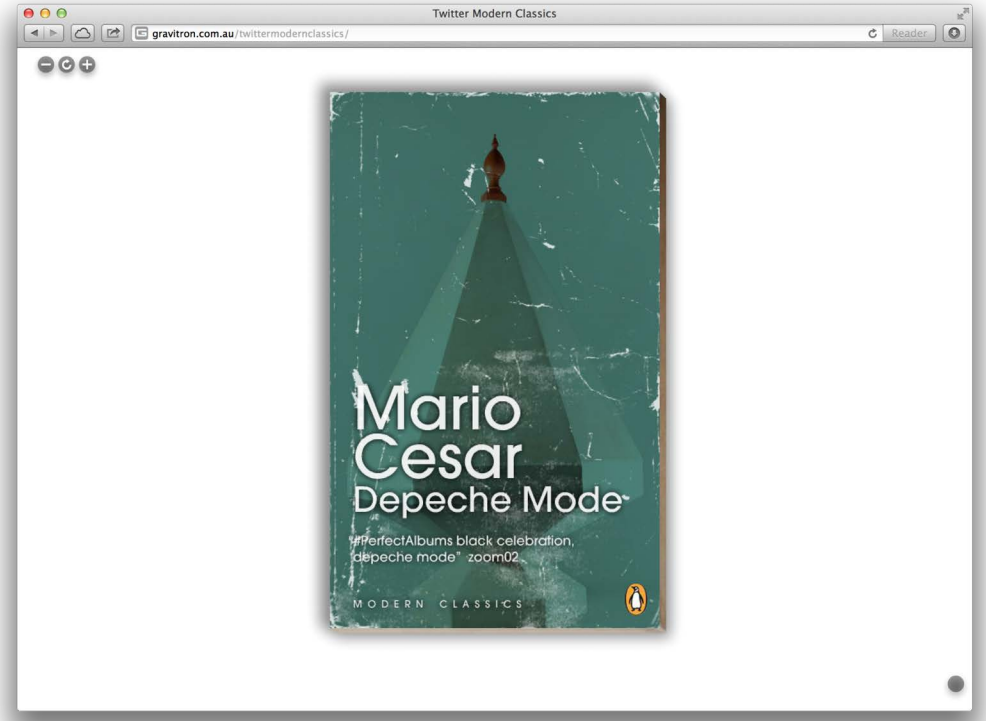
Figure 17. Mechanics. Released December 2008.

Mechanics explores the notion that the form of our software has a direct and important influence on our actions and output [33]. And as its title suggests, the work aims to bring a kinetic sense of mechanical process to computer interaction rather than disguise the interface or aim for “transparency”. As has been stated above, the primary contribution of the work was in developing the notion of “critical whimsy”, a concept that runs through all of the creative works. The work was also important in defining a boundary for the research. Whereas the majority of the creative works represent the Twitter source, Mechanics is concerned with user input. Interactivity is vital to all the creative works featured within this project but it is a subject that requires exclusive and in-depth attention rather than peripheral inclusion. Thus it was decided to focus more narrowly on representation of dynamic data rather than interactivity more broadly.

[33] See for example “Word Perhect” (Takahashi & Pollard, 2000)

5.4. Twitter Modern Classics

http://gravitron.com.au/twittermodernclassics



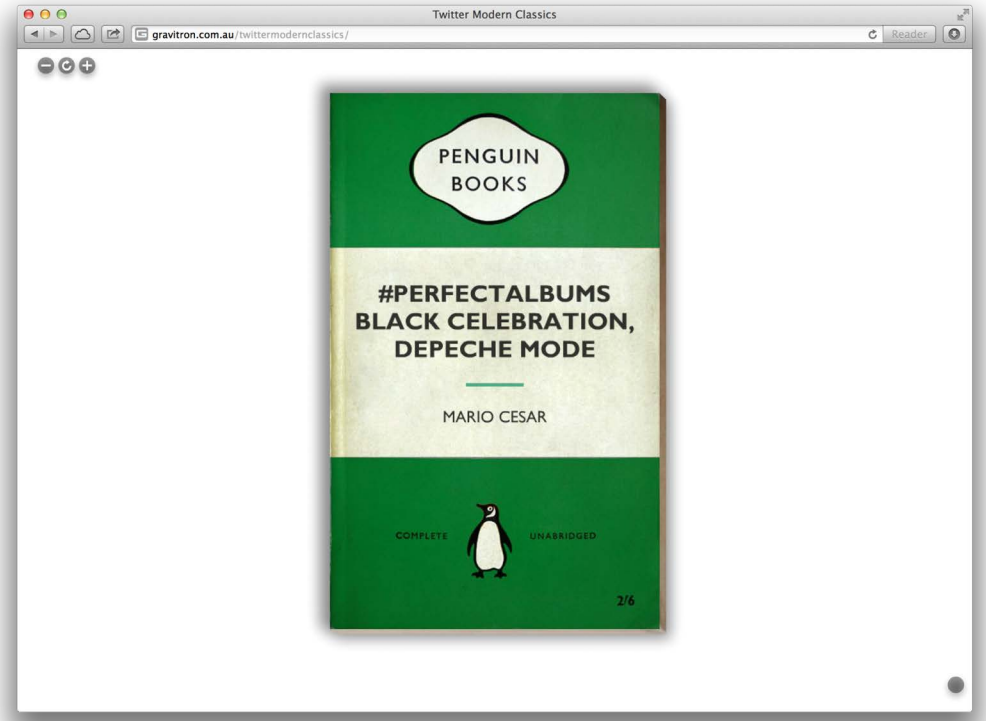


Figure 18. Twitter Modern Classics web version. Released January 2010.

The paperback cover is a quintessential graphic design form; a canvas within which typography and imagery are used to illuminate the title of a book. In Twitter Modern Classics (TMC) (Figure 18) regular tweets, typically lost amongst a Twitter timeline, are afforded unlikely gravity when re-imagined in the context of the classic paperback novel cover. As Whitelaw writes, TMC uses “dynamic design as a poetic strategy, a way to strike sparks of joy and surprise from the collision of form and content” (Whitelaw, 2011a). Another key aspect of the “poetic collision” within the piece is the generation of the title. A simple algorithm processes each tweet and identifies particular key words that it then assembles into a title between one and three words long, depending on the template. Juxtaposing words and image is a simple trick but an incredibly effective one that still underpins a vast quantity of print advertising.

To render each tweet involves a complex combination of processes. In addition to the tweet data, each cover must download relevant imagery and fonts, before rendering the tweet according the specific rules of the template. It is an elaborate dance but one that yields some very interesting results.

On a technical front the application involves some novel techniques involving interaction between client and server-side scripting. One process that involves a combination of these different interactions is the act of posting a cover to Flickr and Tumblr (Figure 19). The process sees the application first convert the screen-rendered cover to a bitmap version: which entails re-rendering the cover offscreen. The bitmap is packaged with the appropriate text and data required by the Flickr and Tumblr APIs before being posted to the gravitron server which caches the image and completes the post to the Flickr and Tumblr servers.

The piece has received considerable attention online and has also been exhibited in various forms. It has an automatic presentation mode which allows it to be projected at events or displayed on large LCD screens (for example Figure 21). There is also a stand-alone exhibit version (Figure 20) which houses a small printer that produces postcard-sized prints of selected covers. The box features four simple buttons, each with a single function: next, back, reset, print.

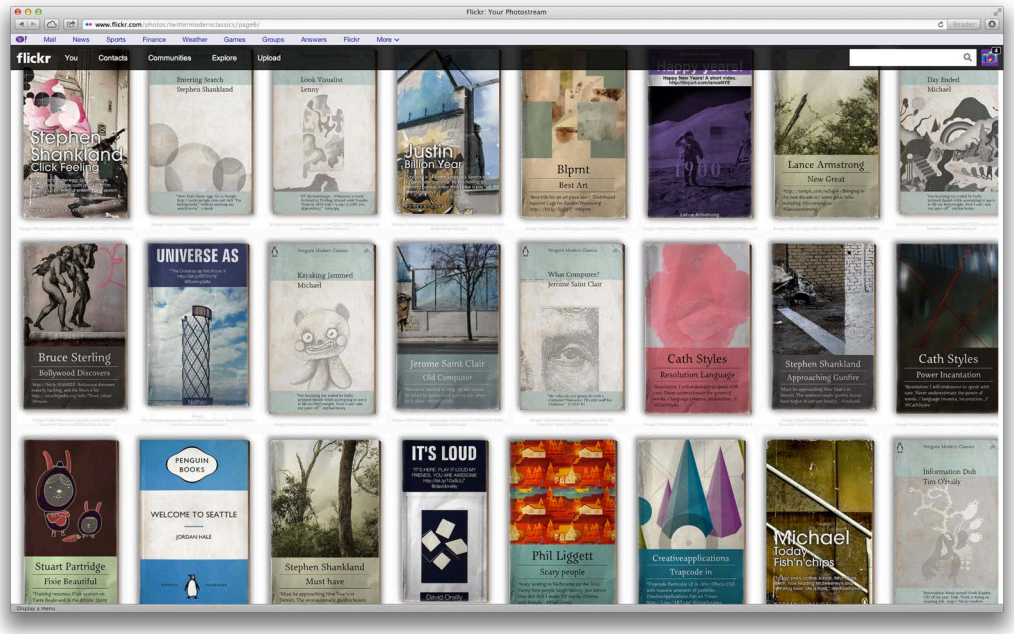


Figure 19. Twitter Modern Classics covers shared via Flickr.[34]

With its appropriation of print-based graphic design conventions, Twitter Modern Classics connected the research with established graphic design concepts and processes and demonstrated their relevance in a contemporary computer context. Reciprocally, through the process of adapting print-based forms and design approaches for a “live” interactive computer context the piece highlighted emerging opportunities and challenges in transforming graphic design to the computational context. These themes are amongst the most significant in the research project, and are evident in many of the subsequent creative works and in the thinking and research evident in chapters 6 and 7. The work also contributed on many technical fronts: working with APIs (Twitter, Flickr and Tumblr); programming for asynchronous network communication; managing security (cross-domain policies, proxies); image manipulation and analysis; font and resource loading. In addition to being useful, intimacy with such fine-grained technical details dramatically influenced the conceptual and theoretical frames of the research, as is most evident in Chapter 7.

[34] http://www.flickr.com/photos/twittermodernclassics

Figure 20. The “Cultural Interfaces” exhibition at CraftACT. 9/11/2010

Figure 21. Senator Kate Lundy opens the UC Research festival, University of Canberra. 18/10/2012

5.5. Tweet Report

http://gravitron.com.au/report

This work draws upon and contributes to the current design discourse around data graphics. It is particularly inspired by the work of Nicholas Felton, whose "Feltron Annual Report"[35] series are seminal works in the data graphics field. The Feltron Reports present the minutia of Felton's daily life as beautifully crafted info-graphic posters. My Tweet Report (Figure 22) draws some obvious cues from Felton's work and shares his focus on quantifying the unlikely and overlooked.

Instead of the usual focus on the textual content of the tweets, the Tweet Report treats the entire tweet stream as a dataset. Looking at an example of the json data that the Twitter API delivers, it is apparent that it already contains various classifications and values: statuses count, friends count, followers count, created date, etc. In addition to using these existing values, the Tweet Report performs a significant amount of data processing of the json object, creating totals for things like hashtags, mentions, retweets, replies, etc, and recording relationships between totals and those who contributed to them.

While at first it presents as a static poster, the Tweet Report is an interactive work that rewards inquisitive clicking. Clicking different elements reveals their relationships to the various polls, charts and lists. Click on a name and associated elements are highlighted. Click on a time band in a bar chart and associated content is highlighted. Click on a slice in a pie chart and again, all associated elements will be identified and highlighted. The austerity of its layout and typography is contrasted with the gentle humour in the titling of the statistics; verbosity values, wordiest tweet, tweep types, hyper-textual connectivity. The layout uses a horizontal and vertical grid, the latter being quite unusual in computational settings, the uniformity of which helps with ordering and positioning of elements. The use of the grids in combination with vector-based graphics[36] means that the report looks beautiful on high resolutions screens and also in print.

---

[35] http://feltron.com/ He is Felton - his site is FeltRon

[36] fonts, dom elements, and svg

The Tweet Report is entirely generative and does not use any external graphics[37]. The bar charts are made using HTML DOM elements and the pie charts are rendered in SVG with some help from the D3 javascript graphing library. It uses the Masonry javascript library to assist with transitioning the layout in response to changes in the browser width, and also makes extensive use of the Kernit[38] javascript library for the fine-grained kerning of heading text and the large numerals.

To implement the modular vertical grid requires additional processing as CSS cannot address the conditional rules of the grid. The default elastic boxes of the DOM expand as required unless given specific width/height values. In the case of a vertical grid we need to maintain elasticity to accommodate content but want to step any expansion to the minimum grid value. It is easy enough to achieve with some simple code but surprisingly such options are not currently possible in standard CSS.

The Tweet Report maintains the complex relationships between all of the different elements on the screen and targets elements based on CSS classes. For example, every element associated with a particular tweeter will have the tweeter's class associated with it. The same approach is maintained for elements such as tweets, time, words, hashtags, links etc. The result is a complex web of connections but remarkably the HTML DOM manages it without issue. The web of CSS classes is used in highlighting elements relationally, which is typically achieved with jQuery. However, because jQuery cannot style SVG elements identified by CSS class I instead use temporary global CSS styles, adding a style to the head of the html document, including styling attributes for a particular class and letting the browser do the highlighting work on both HTML and SVG elements.

Tweet Report continued development of the themes established in Twitter Modern Classics; exploring the transformation of established graphic design concepts and practices for a networked interactive context. On a technical front it investigated how the browser's Document Object Model (DOM) could be used as a canvas for sophisticated visual representation and produced a number of novel techniques[39] with significance for subsequent creative works. Importantly, with its aggregated perspectives on a common Twitter stream the work demonstrates how

---

[37] It does reference a web hosted font but all graphic imagery is generated with code.

[38] Kernit is a library developed as part of this project.

[39] Examples include collision detection for font kerning, vertical grid alignment, methods for dynamic CSS application.

data can be read and represented in unconventional and critical ways. As was argued in Chapter 3, users look to producers not only to make interfaces easy but increasingly as cultural translators capable of providing insightful mediations of the morass of information and media in everyday life. This line of enquiry is more fully explored in Chapter 7 and the notion of “Data as Script”.

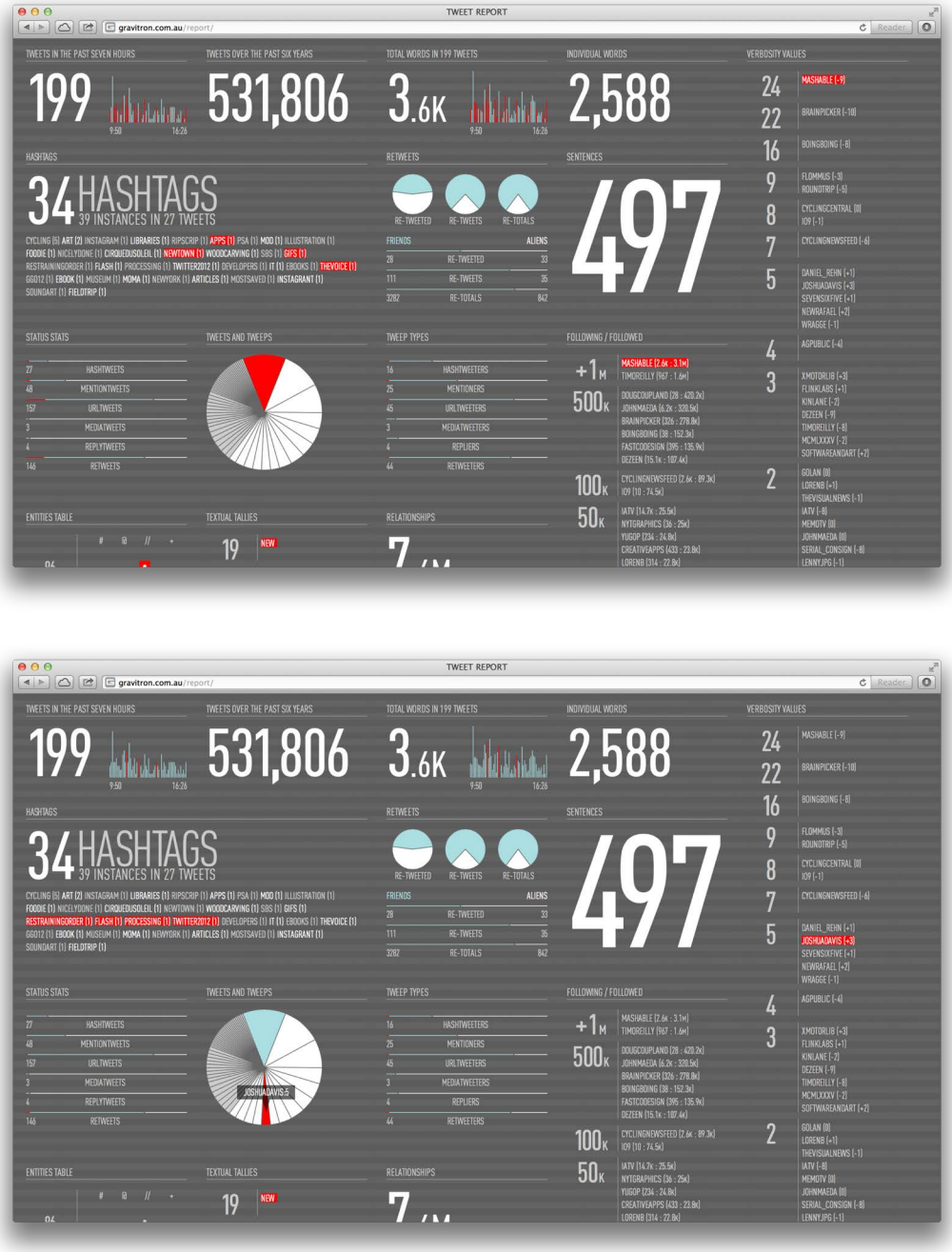


Figure 22. Tweet Report. Released October 2011.

5.6. 3D

http://gravitron.com.au/3d

In this work (Figure 23), the tweet stream is represented in a virtual space with older tweets in the far distance and more recent tweets in the near distance. Double clicking on a tweet shifts the focus and zooms the view to its proximity. Double clicking in space resets the view. Single clicking a tweet highlights it and others by the author.

While the piece has some practical logic governing its arrangement of tweets, its primary aim is one of affect, creating an atmosphere in which the viewer floats amongst their tweet stream. It is most effective at closer proximity, where the sense of scale and depth of field becomes exaggerated. The arrangement of tweets is far from practical, instead promoting an ambient form of browsing where tweets are discovered by chance as they float into view. The distance, opacity and depth of field are an effective way of communicating a sense of time.

Our understanding of typography is informed by its origins in print technology, yet here it is rendered into virtual space. It is an interesting evolution, made all the more remarkable by the fact that the depth of field and 3D effects were realised using web standard CSS; DOM elements are scaled and positioned according the CSS Z-attribute they have been supplied with. The depth of field effect is also achieved with CSS but unlike the 3D effect, CSS has no defined controller for depth of field. To achieve the effect I used the CSS “text-shadow” attribute for text blurring and the CSS “box-shadow” for blurring the box. The parallax scrolling, even with the blurred elements, is very smooth and precise, however changing the depth of field is relatively slow as it requires iterating through all of the html objects and applying different levels of blurring based on each object’s Z-depth. Technical limits aside, the combination of 3D scaling and text blurring effectively creates an atmosphere. Like the use of the Box2D physics with the DOM, the introduction of 3D space invites further experimentation.

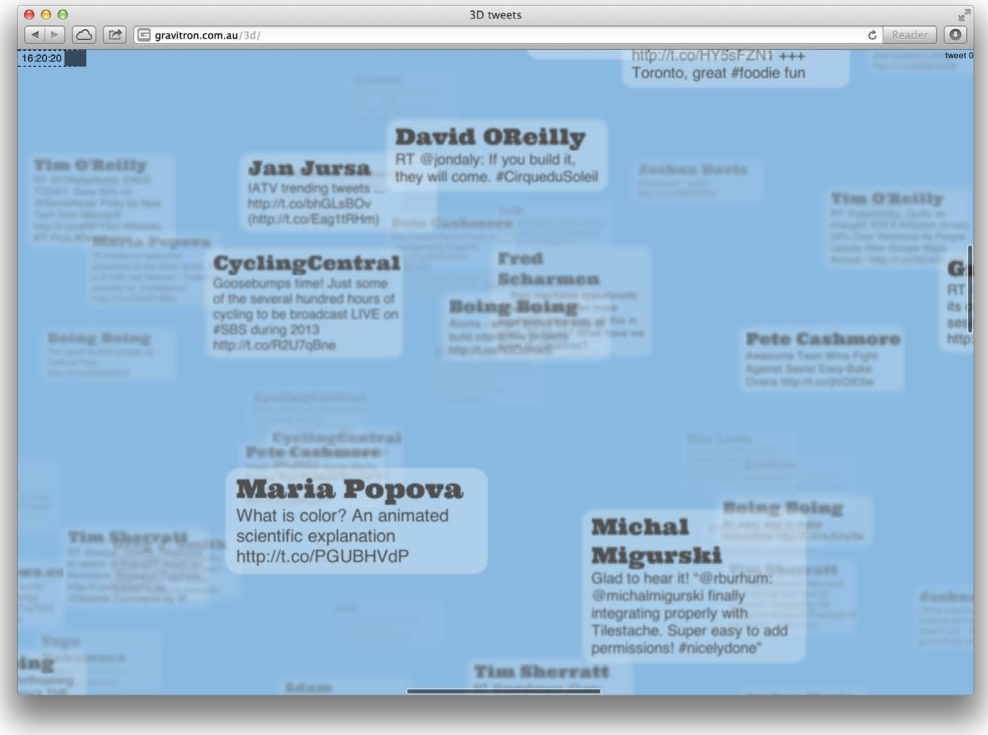


Figure 23. 3D. Released August 2011.

Evolutions in CSS like 3D, animation, and transitions are evidence of how established practices, like typography, are evolving in response to the computational context. It is fascinating to see attributes relating to time and space added as styling options within a display language that has evolved from page conventions. Continuing in this computational vane CSS already has limited support for variables which will allow programmatic assignment of values for colour, size, etc.

Like Tweet Report, 3D continued to explore the browser DOM as a creative canvas for sophisticated graphic representation. Its particular focus was the exploration of browser “space” as a medium for textual information that has conventionally been confined to the “page”. The novel adaptation of CSS in this work underlines the cultural nature of software production; technologies such as HTML and CSS are developed by organisations with particular implementations in mind, whereas producers appropriate and adapt those technologies in unexpected ways. Through such appropriation and misuse producers have the capacity to shape the future direction of technology. Production here is discourse.

5.7. Physics

http://gravitron.com.au/physics

Physics (Figure 24) on first appearance would seem to be one of the most abstract representations amongst my collection of creative works, however, it is a fairly straightforward data visualisation with a few twists. The basic logic of representation is much like a graph. Each tweeter appears as a disc featuring their avatar pic, the size of which is determined by the number of tweets made by the tweeter. Complexity grows with the addition of time; the Twitter timeline is replayed as a time-lapse sequence with discs arriving in the order that the tweets were made. A further increase in complexity occurs with the inclusion of physics; new tweeters fall from the top of the browser and bounce amongst the existing pool of tweeters, the discs rotating and jostling for position. The user can grab discs and hurl them about the browser, causing explosive collisions and rebounds.

Figure 24. Physics. Released July 2012.

While it is certainly satisfying to manipulate different tweeters about the screen and feel the forces of gravity and momentum at work, the physics simulator does also fulfil a more pragmatic role within the piece. The issue with visualising a large set of differently sized objects, like our collection of tweeters, is how to organise them in an economical way within the confines of the display area. There is much research and literature[40] dedicated to addressing organisational problems such as this but instead of following the stacking algorithms route I used the Box2D physics simulator as a layout engine. Not only can the physics simulator deal with the size and space issue, it also handles dynamic changes in the size and quantity of objects.

Box2D is an impressive physics engine which is widely employed in computer gaming[41]. Working with Box2D adds another layer of abstraction to what is already a fairly crowded context; the web "page" employs HTML to define objects, CSS to style the appearance of objects, Javascript to control the behaviour of objects, and Box2D to determine the positioning of objects. The Box2D engine makes no reference to browser windows or pages and is completely context-independent, working with its own virtual world and coordinates. To position objects on screen the Box2D coordinates need to be translated for the display context. In a web setting designers typically render a scene using the HTML canvas element but I saw the opportunity to work directly with the HTML Document Object Model (DOM). What is remarkable is how good the DOM rendering is, both with regards to frame rate and to image quality. When viewing the Physics work on a high resolution Retina screen, all of the objects are rendered with startling clarity and precision. It suggests exciting opportunities for further experimentation with Physics + DOM.

Some would assume that web-based works employing technologies such as HTML/CSS/Javascript and/or Flash should be considered "interface design" rather than "software". Whether using "real" programming languages such as C++ and Objective C or web languages such as Javascript or Actionscript does not determine whether a work is classified as software or interface. In this research project the term software is used because the works produced (and those cited

---

[40] See for example: Leutenegger, S.T.; Lopez, M.A.; Edgington, J., "STR: a simple and efficient algorithm for R-tree packing," Proceedings 13th International Conference on Data Engineering, pp. 497-506, 7-11 Apr 1997.

and: C. C. Lee and D. T. Lee. 1985. A simple on-line bin-packing algorithm. J. ACM 32, 3 (July 1985), 562-572.

[41] The mega successful Angry Birds series of games is entirely dependent on the Box2D engine.

elsewhere in the thesis) involve complex computational operations for data manipulation, on-screen presentation and interactivity. They are much more than graphic interface stylings. Physics sought to investigate this space between interface and software; between representation and adaptation. The exploration of these ideas was not only vital to informing the conceptual models fleshed out in Chapter 7 but also in developing the notions of creative code discussed in Chapter 6. Programming here is not a means of rendering a predefined form or system; programming is a means of sketching, of ideating, of exploration and discovery.

5.8. MONO

http://gravitron.com.au/mono

MONO (Figure 25) is essentially a conventional Twitter application. It brings together different concepts and techniques developed in the various creative works while also exploring new technical and aesthetic terrain. Stylistically, MONO is modelled using a minimalist range of elements; vertical and horizontal grids, Helvetica typeface, and monochrome colours with selective use of limited highlights. While it presents a relatively conventional representation of the Twitter source, like my other creative works in this project, MONO seeks to reveal more of the latent information contained within every data stream delivered by the Twitter API. In conventional Twitter apps this information is available on enquiry for those willing to click through to discover more about a tweeter. In MONO I bring more of this information to the surface and use it to adorn each tweet in the column. I use the term “adorn” intentionally to highlight the notion of data as an aesthetic embellishment. In the case of MONO this is reflected in the persistence of statistical numerical values and associated icons in an info-graphic style. As with Tweet Report, the presentation of data and information is integral to the visual aesthetic of the work but also provides a user with additional entry points for interaction and exploration of their Twitter stream. For example, clicking any of the five figures at top of a tweet reveals those lists: the author’s tweets; their favourites; who they follow; their network of followers; and finally their list of lists. Similarly, one can view conversations, a Tweeter’s profile details, retweeters or information about the application from which the Tweet was sent. All of the additional information and media is loaded into the one column rather than creating new views, thus the title MONO.

The piece works well on desktop browsers but is most at home on an iPhone or other mobile device; its geometry, monochrome palette and crisp vector graphics all being ideally suited to the high resolution displays of contemporary mobile devices. While still a conventional web-based work MONO behaves like a standalone app on iOS devices, employing the html “manifest”[42] and “local storage”[43] in order to provide users with a persistent application state. In simple terms, it means that unlike a typical web app which loads afresh each time a user

---

[42] http://www.w3.org/TR/2011/WD-html5-20110525/offline.html#manifests

[43] http://www.w3.org/TR/2013/REC-webstorage-20130730/#the-localstorage-attribute

returns, the MONO app preserves its previous state, providing a continuity between sessions as with a native iOS application.

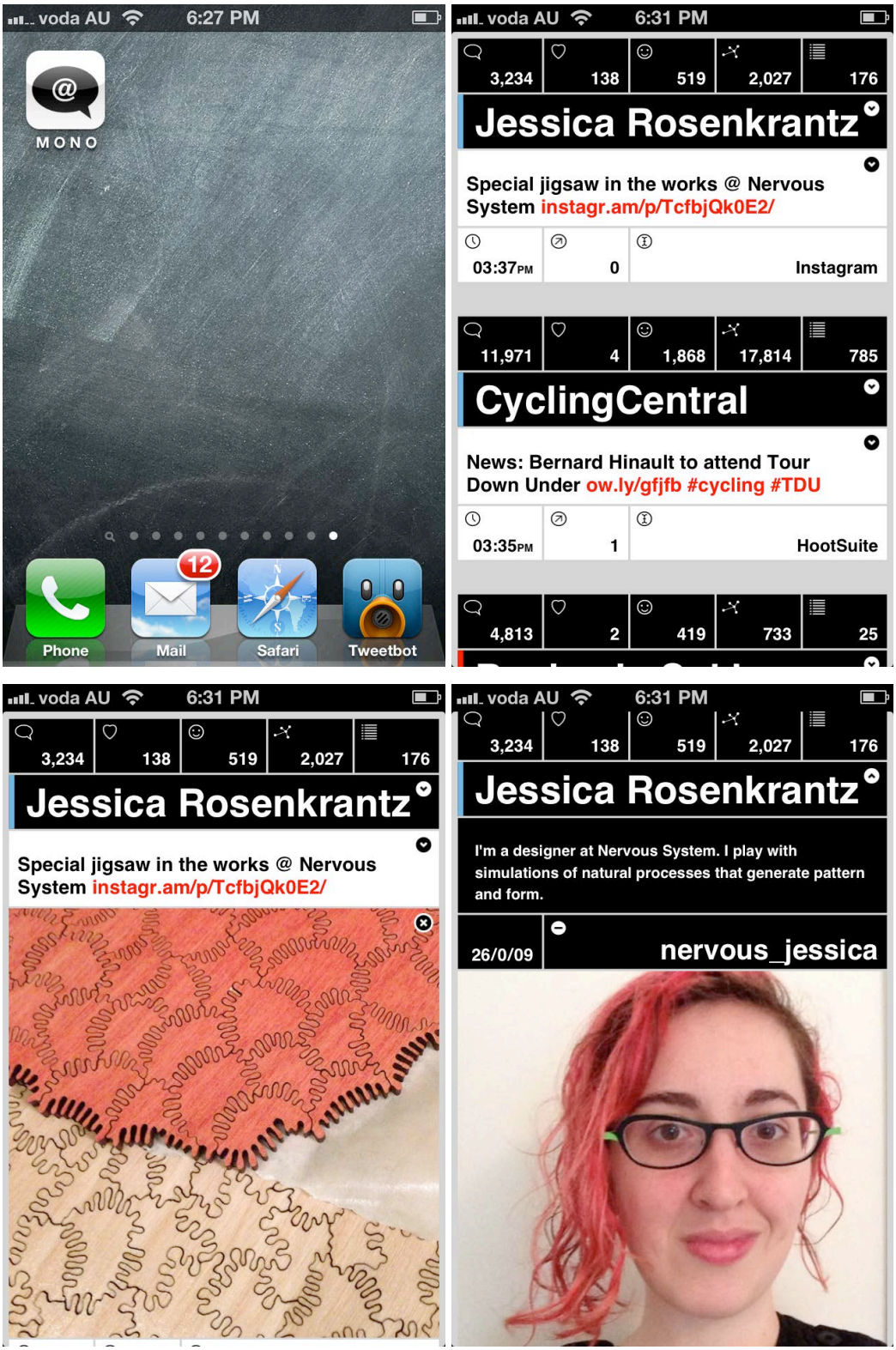


Figure 25. MONO. Released October 2012.

Technically, MONO continues to explore the possibilities of emerging browser technologies and in particular the “manifest” and “local storage” as a means to create a persistent state for each user of the application (especially relevant for mobile and tablet contexts). Conceptually, it explores generative graphic system design but also notions of data aesthetics as a way of revealing more of the content embedded within a regular Tweet stream. Despite the playful, whimsical nature of the creative works, MONO demonstrates that the concepts and techniques embedded within can be applied to more instrumental and prosaic ends. This notion is further explored in the following piece, Kernit.

5.9. Kernit

http://gravitron.com.au/kernit

A common concern in my creative works is the computational positioning of text. This seemingly simple concern turns out to be surprisingly complicated. In the case of Tweet Motion, I resorted to rendering each letter as a graphic object in order to get accurate boundary values. However, this is a very laborious solution and while it works reasonably well for Tweet Motion it also introduces new problems, particularly with regard to kerning. In a work like Tweet Report the positioning and kerning of the type is crucial to the piece; the tightly kerned sans-serif typeface is reminiscent of the utilitarian elegance of Swiss Design. To achieve the tight fit kerning evident in Tweet Report I developed a custom kerning script called Kernit (Figure 26).

Figure 26. Kernit. Released October 2011.

In a well designed typeface the letter spacings for every lettering pair are predetermined and their values included within the typeface file. But in many cases a font either does not include all of the detailed kerning information or the screen rendering of the font does not apply its kerning rules as it would for print. Kernit provides an alternative kerning solution. It works by breaking a string of text into

individual HTML span elements and setting custom margin positions to control the relative spacing between letter pairs. The problem with spacing letters in this way is that the bounds of the span are still quite crude. Fine kerning takes into account the specific contours of the letters. For example, in kerning the letters “T” and “o”, the “o” will be spaced to nest in under the cross-bar of the “T”. If we use the bounds of the span boxes no such nesting will occur. To solve this issue, the Kerning routine uses a collision detection algorithm[44] (Lenton, 2011) developed for computer gaming. The letters are rendered as bitmaps using HTML canvas and the distance between them reduced until they collide. At that point a value for their kerning is determined and used to space letters on screen.

Kernit also includes some custom procedures for checking the load progress of web-hosted fonts. CSS today supports the use of web-hosted fonts which are loaded with the other elements that constitute a web page. The problem for Kernit is that it needs the font to be loaded before it can determine letter spacings. As a result I developed a custom font checking routine which works by rendering two text strings to bitmap using canvas; one using the default font and one using the web-hosted font. It then compares their dimensions; if they match, the font is still loading, if not, the font is ready and the kerning process can commence.

To control the Kernit script a designer uses CSS. To declare an element as kernable, they apply the “kernit” class. CSS is also used to define the default kernit spacing and to set spacing for specific letter pairs. This use of CSS to program the Kernit code is an attempt to make the script more accessible to non-programming designers. It means that a web designer can implement the Kernit script without having to code any Javascript.

The primary focus of Kernit was a technical one; how to achieve finely tuned kerning of web-based text. The issue of kerning in a web setting is relevant to many of my creative works and ongoing creative production. In response to that problem, Kernit offers a novel and workable solution. But because it is a routine rather than a finished form, Kernit draws attention to code as creative output and as an instrument for discourse. Each of the works featured here is as much about code as it is the forms and behaviours that constitute their tangible interfaces. The different Figures featuring the rendered works could equally show the lines of code responsible. For example, the interface shown in Figure 26 is a demonstration, the

---

[44] Joseph Lenton - PlayMyCode.com

“work” is the code itself. The collision kerning algorithm of Kernit is but one of the various computational techniques and routines embedded within the different creative works. By extracting and exposing the kerning routine, Kernit highlights the potential of the creative work to produce generalizable outcomes with relevance outside the scope of this research project. In addition to providing such instrumental value, the work also serves as a contribution within a broader discourse concerned with web design and technology. Much of that discourse is focussed on how current technologies can be applied for creative purposes[45] but the discussion also includes technology adaptation[46], as well as speculation about future technical features[47]. Kernit contributes to this current discourse, demonstrating a working solution and promoting kerning as a topic of value, something that could ultimately be explicitly addressed in native CSS. As has already been stated above, production here is discourse.

---

[45] Example: http://css-tricks.com/all-about-floats/

[46] Example: http://css-tricks.com/viewport-sized-typography/

[47] Example: http://css-tricks.com/on-responsive-images/

5.10. Critical Whimsy

A distinct character of all of the works is their use of whimsy. All of the pieces resist conventional representation of the Twitter service, and in doing so, offer a critical perspective on it. Viewing a set of Tweets in 3D space, or as a collection of novel covers, or as an elaborate info graphic report, gives cause for reflection on the nature of tweeting, on the qualities of the discourse and a user's participation in it. However, the criticality of the works is always shrouded in a strong sense of whimsy. There can be joy in slowing down, in gaining a new perspective on a familiar form. This form of criticality is like that of Niedderer's "performative object" (Niedderer, 2007). In developing the concept, Niedderer created a set of tumblers pierced with a series of holes. The tumblers require a drinker to use their fingers to prevent liquid from leaking out. According to Niedderer, the disruption is intended to cause mindfulness: "in the "Libation Cups", the function of holding liquid is disrupted… However, the users can close the five holes with their fingers, thus restoring the function of the cup. The additional action required draws the user's attention to the cup, and causes reflection on its design and use, and on an additional level of meaning of this use. We therefore can conclude that we can cause mindfulness through materiality" (Niedderer, 2007).

Niedderer's cups possess a critical characteristic, in that they offer a critical perspective to the design discourse, questioning the values that designers embed within their artefacts, but her works are gently provocative - they do not intend to infuriate their audience but only to awaken them to the experience in which they are engaging. The playful nature of Niedderer's cups softens any critical element of the work. Similarly, in all of the Twitter pieces, there is a generous use of whimsy which serves to defuse any critical edge. In this way whimsy can be a strategic support for abnormal speculative works. The computer medium is dominated by the instrumentalism of its origins and the proposition of alternatives can be extremely difficult. It is much like the traffic culture that Monderman worked so hard to challenge but in the case of software, engaging with poetic forms is most often a side route rather than a major thoroughfare - people elect to travel it and can very easily bypass it. The simple point here is that propositions which are abnormal, which ask users to slow down and consider their actions, can be easily dismissed as annoying, or useless, or inefficient (which they may well be). Whimsy is a useful means of defusing the situation. It immediately dismisses the urgency of instrumentalism and instead invites an audience into play and gentle reflection.

The criticality discussed here is quite distinct from that discussed in Critical Design (Anthony Dunne, 2005a), Adversarial Design (DiSalvo, 2012) or Design Fiction (Bleecker, 2009). The issue with such approaches is that, like instrumental approaches, they can cast the production process as a means of exposition, a way to realise a set of preconceived objectives. It is akin to forms of conceptual art whose value does not depend on artisanal notions of craft - its value lies in the concepts it presents irrespective of who actually crafted it. My objective here is not to criticise such processes or lay claim that artisanal crafted works are superior, but only to stress that the conception of poetics advanced in this thesis is dependent on a material relationship with the medium, and as such places craft at the centre of practice.

When Dunne suggests that everyday interactions with technology can be rendered poetic, I enthusiastically agree. However, the term poetics suggests a form of expression which focuses on the aesthetic qualities of language rather than their typical prosaic function. This means that poetics is not simply about making works which are different or contentious, it means responding to the aesthetic qualities of the medium, and in the case of software, that means code.

Creative Code

6.1. Introduction

In its use of classic printed book covers, Twitter Modern Classics (TMC) makes explicit reference to a highly familiar graphic design form and, as importantly for this project, draws on the rich history of graphic design practice. In this section we consider some of the deeper conceptual parallels between design practices developed for print and those in the computational setting.

As a field of practice graphic design is expansive, and describes a broad set of approaches to the design and production of graphic media. Its immediate relevance to this project is as a practice based around giving form to an independent (and often dynamic) source text. Within this particular field of graphic design there are two distinct practices: one in which the designer works intimately with a specific text, and the other in which the designer works with a constantly changing text. In the first, the designer will typically seek to use graphic form to represent the character of the text, to illuminate its meaning through artful graphic composition. In the second, the designer will typically seek to develop a robust and flexible system capable of legibly displaying the required variety of content. In this

case, the designer does not have the same intimate access to a text and cannot therefore engage in the sorts of “illumination” suggested in the first scenario. That is not to say that a systemised solution cannot have an aesthetic character, but only that the relationship between form and content is not based on the close analysis of text used in the illumination approach.

In summary then, we can distinguish between two distinct graphic design approaches: customised and systemised. Where the customised approach seeks intimate understanding of its content and aims to poetically represent it in graphic form, the systemised approach is agnostic about the meaning of its content, working only with content taxonomies. Having established these two distinct approaches, our interest is in graphic design’s practices of systemised design, but we will return to the custom, particularly as the discussion draws focus on notions of poetics.

6.2. Systemised Graphic Design

Perhaps one of the best examples to introduce the concept of system-design is the newspaper. The typical newspaper is a matrix of text and image presented within defined containers; the printed pages. And while the content of a newspaper changes from day to day, we can observe in its different editions consistent formal elements governing the organisation and presentation of the dynamic content. It is this systemised approach to the graphic representation of a continuously changing content source which is our focus here.

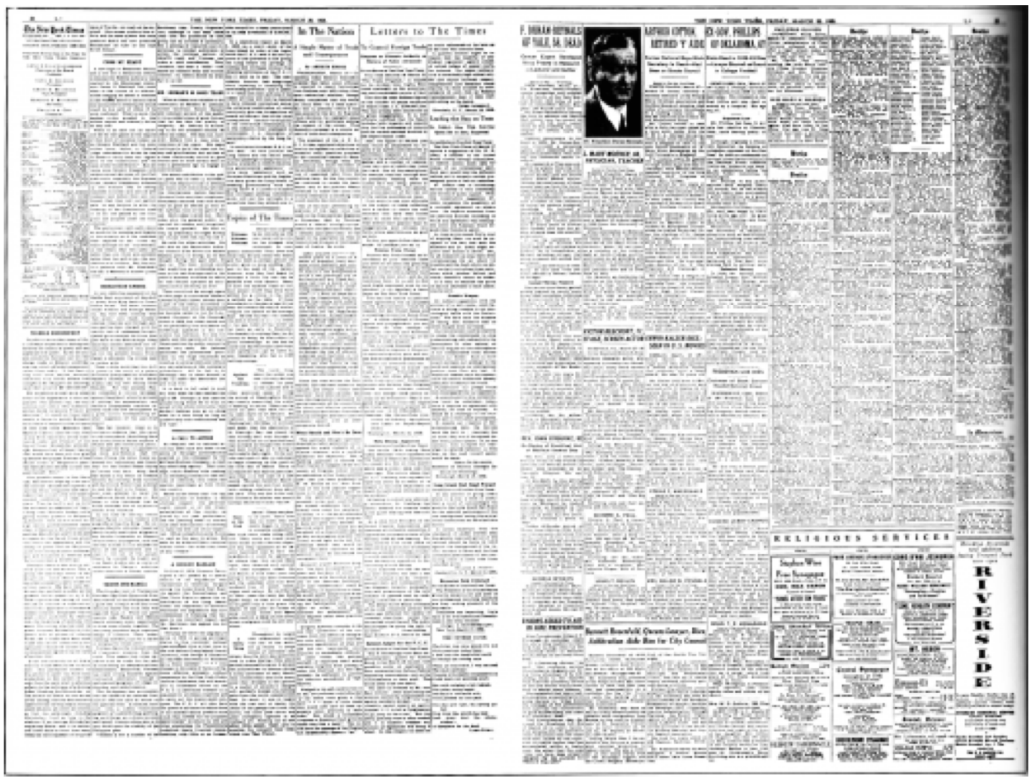


Figure 27. New York Times 1958

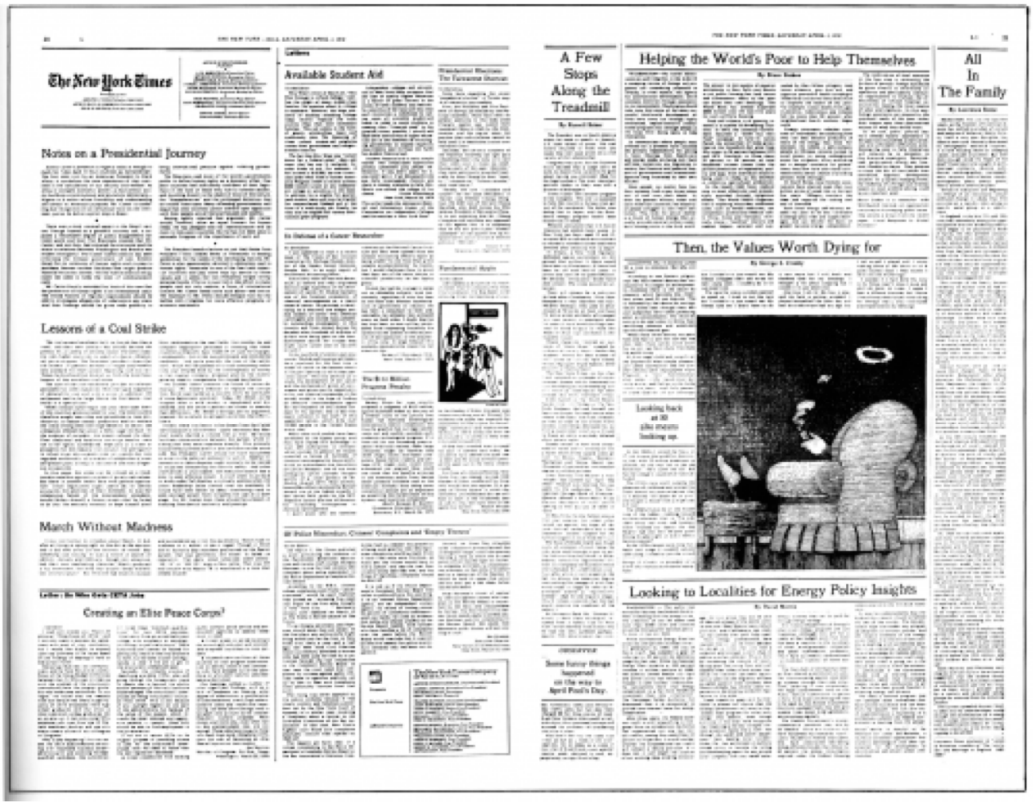


Figure 28. New York Times 1978

(Source: Newspaper Design for the Times, Van Nostrand Reinhold, 1990)

The newspaper as dynamic design framework is a feat of engineering but also one of interface aesthetics. Comparing Silverstein's editorial layout with the outgoing format it replaced, Roger Black reflects "It's hard to believe the before example (1958) (Figure 27) is not much more than 50 years old. It looks like the 19th century. Yet, the after (1978) (Figure 28) could have been printed yesterday" (Black, 2011). Both versions solve the practical problem of presenting a large volume of dynamic content within the confines of a page, yet Silverstein's editorial layout introduces an aesthetic elegance which is absent in the 1958 version. Put simply, his innovation was in creating a system that was both more practical and beautiful. His organisation of space and use of typographic hierarchies make the page more navigable and the articles easier to read, but also provide an aesthetic balance and rhythm which are absent in the suffocatingly dense 1958 version. What is remarkable about the example of the newspaper, and in particular a broadsheet like the New York Times, is the volume and variety of content that the system accommodates. By contrast, the iconic Penguin book cover was designed to accommodate only the most essential information, yet it stands as another exemplar of dynamic-design.

Originally designed by Alan Lane in 1935, the Penguin covers employed a powerful, yet extremely minimal system: a text-only design using a sans-serif typeface of two weights and a set background colour denoting the series to which a book belonged (orange for fiction, blue for biography, green for crime). Lane's initial system of colours and typography were further formalised by Jan Tschichold in 1946 (Doubleday, 2006). Tschichold had gained renown for his 1928 text "The New Typography" in which he outlined a disciplined and ordered approach to typography and page layout based around the fundamentals of the grid. It is widely regarded as the most important foundational theory of modern typography and book design (Kinross, 2006). Prior to Tschichold's arrival, Lane's system was relatively cohesive, particularly in comparison with other publishers of the day, but had been subject to variation at the behest of different editors and printers. Tschichold, who had a proven passion for typographic systems, set about refining the Penguin design; unifying the front, spine and back, and redrawing "Edward Young's endearingly amateurish Penguin symbol in eight variations" (DesignMuseum, 2005). Tschichold also defined a strict set of Composition Rules which he insisted that Penguin's typographers and printers follow. The result was the book series which is today an icon of 20th century design.

In both of these iconic examples we have robust design frameworks developed for particular dynamic source content, and in both cases there are clear practical motives for those frameworks. The distinctive columns of dense text which characterise late 19th and early 20th century newspapers developed out of the need to represent a vast volume of information in an affordable daily publication using lead letterpress printing technology. With advancements in technology and an increased appreciation of visual aesthetics, designers like the New York Times' Lou Silverstein were able to innovate and develop graphic solutions that were both practical and aesthetically pleasing. The Penguin covers also emerged in response to a practical problem, that of bringing affordable publishing to the masses, particularly the commuting workers of London (Baines, 2006). As with the example of the newspaper, the design framework of the Penguin covers expedited the design process but also created a bold visual brand that was instrumental in the success of the Penguin publishing model.

Figure 29. Design by Robert Büchler for Typographische Monatsblätter 1960 Issue 11 (Source: http://www.tm-research-archive.ch/issue/1960-11/)

Another exemplar of a systemised and programmatic approach is Karl Gerstner's book Designing Programmes (1964) which Lupton describes as "a manifesto for systems-oriented design" (Lupton, 2004). Gerstner was principal in establishing the rationalist ideals that defined "Swiss design", also known as "Swiss style". A signature of their approach was the extensive and disciplined use of grids, which they used as a programmatic system. While suggestive of a conservative approach to layout, much of the their grid-based design was highly experimental as Lupton describes; "Constructing ever more elaborate grids, the Swiss designers used the confines of a repeated structure to generate variation and surprise. Such grids could be activated in numerous ways within a single publication, always referring back to the root structure." (Lupton, 2004) The "Typographische Monatsblätter" (Figure 29) journal was perhaps the most important vehicle for disseminating the approaches that defined the "Swiss Style".

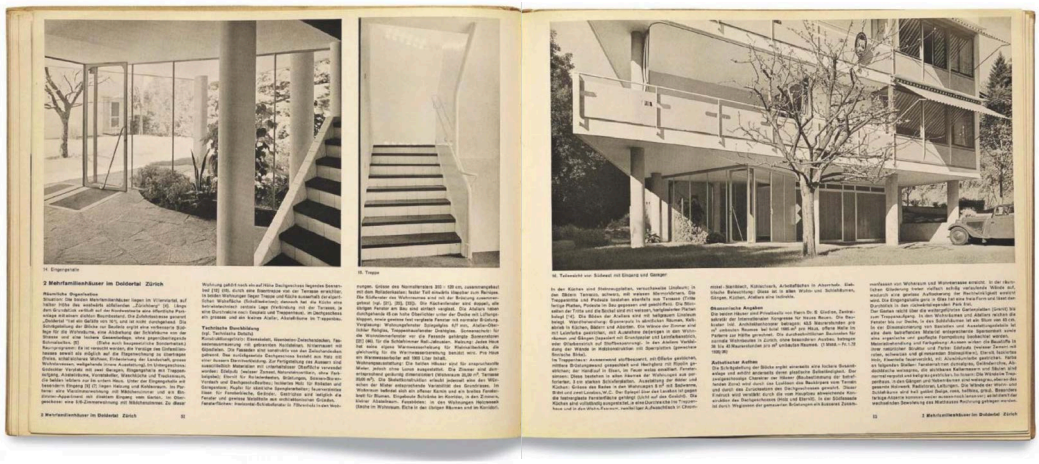


Figure 30. The New Architecture, 1940, Max Roth. Designed by Max Bill. From Thinking With Type, Lupton, 2004.

The ideas of Tschichold's "New Typography", which Gerstner and the Swiss designers further developed, are today evident in a wide variety of contemporary graphic media, in both print and computer forms. Examining a page layout from "The New Architecture" from 1940 (Figure 30), the resemblance to modern graphic forms is striking; the only indicator of its history being the vintage car in the bottom right of the photo.

6.3. Aesthetics of Illumination

The Swiss designers explored the formal aesthetic of the grid as an alternative to the approach of the aesthetics of illumination, in which graphic form is shaped by the meaning of the content. In illuminated design, the relationship between form and content may be harmonious, ambiguous or even contradictory, but the value of a work resides in an appreciation of the dialogue between form and content. It is interesting to note, in Roger Black's generous obituary for Louis Silverstein, the reverence paid to the notion of graphic design as an art of illumination. The title of his article, a quote from Silverstein, encapsulates this sentiment: "Read the story before you lay out the page" (Black, 2011). The same reverence can be found in countless design awards. It is the celebration of an expertly crafted gestalt formed through arrangement of verbal and visual elements; where the graphic form of a design perfectly compliments the communication intent of the verbal message. It is perhaps at its purest in typesetting, where designers pore over the aesthetic qualities of letterforms in an effort to find the perfect typeface for a particular message and context.

Silverstein's advice, "Read the story before you lay out the page" , speaks to the core proposition of graphic design; that through artful graphic means the designer can bring deeper meaning to a text. His statement implies that creating a complimentary graphic representation requires knowledge of the meaning in a text and that without that knowledge a layout may be superfluous, or potentially unsympathetic to the intent of the text. It is a reasonable assertion to make, and it is an attitude that is common in the rhetoric of graphic design literature and practice. Polish designer Jacek Utko extolls a similar philosophy when describing his remarkably successful redesigns of various Baltic newspapers[48]. At a time of global decline in newspaper circulation, Utko's redesigns inspired impressive increases in newspaper sales (Holston, 2011). For Utko, the design was not to be considered a "look" but the outcome of a ground-up reconception of the newspaper, including its editorial content. And as with the work of Silverstein, it is Utko's highly customised layouts which garner the most attention, particularly his poster-like covers (Society of News Design, 2006). For Utko, the persuasive power of the cover relies on the skill of the designer to translate a storyline into an arresting graphic form.

---

[48] See newspaper design Jacek Utko http://www.utko.com/

Ironically, despite his emphasis on the artful illumination of a source text, Silverstein's work is cited here for almost the opposite reason. While it is possible to single out exceptional examples of insightful interpretation from the New York Times we can also reflect on and pay homage to the system that Silverstein was principal in creating.

6.4. System Aesthetics

At the heart of the illuminated approach is the notion of a poetic form/content relationship resulting from artful graphic representation of a text. With a system approach, a close textual analysis may not be possible but we cannot therefore conclude that the system approach is devoid of aesthetic quality. As we see in the work of the Swiss designers, and in that of Silverstein, Lane and Tschichold, there is an aesthetic of the system. The suggestion here is that the system framework itself is central to our aesthetic evaluation of a dynamic design artefact; that a satisfying design can be achieved through adherence to, or contrast with the rules of a framework. For example, we see within the spreads evidence of particular adjustments and customisation but all within the bounds of the system.

Figure 31. New York Times 1984
(Source: Newspaper Design for the Times, Van Nostrand Reinhold, 1990)

Silverstein's spread for the coverage of Indira Ghandi's assassination (Figure 31) features both the conventional and the contrasting. The editorial content on the right page conforms to the fundamentals of the system, whereas the left page of graphs and charts works outside the conventional system to create a complimentary a-symmetry. The conventional 6 column layout of the right page is contrasted with a 5 column grid on the left (created by combining a 4 column and 6 column grid), with elements such as the circular illustrations defying the grid completely.

In this example, we see Silverstein "riffing" on the internal rules of the system, playing one page of the spread against the other, breaking uniformity to create an arresting visual dynamism. But in addition to the aesthetic qualities of an individual artefact there is an aesthetic appreciation of the collective; the set. To that end, the more products of a system that a viewer experiences the greater their appreciation of the system aesthetic. For example, the long time reader of the New York Times who witnessed the design changes made by Silverstein would have an entirely different appreciation of his designs than the first-time reader who experiences Silverstein's spreads with no other reference. The "appreciation" here is of the system rather than an individual spread or edition. Even more than the example of the newspaper, the Penguin covers represent this system aesthetic. Our appreciation of the Penguin covers is, in part, based on an understanding of the set, with each new cover expanding our understanding of the series system. The goal here is not the pursuit of a Platonic Ideal Form (Shorey, 1938) but a recognition of an aesthetic of the set and the system, both of which have obvious value in a networked computer context where dynamic data is the norm and there are few opportunities for illuminated design based on intimate analysis of a text.

Of all of the creative works completed within this research project, TMC makes the most direct reference to the aesthetics of the set. Viewing the collections of TMC covers in Flickr or Tumblr (Figure 32) makes apparent the qualities of the set. Seeing a tweet rendered multiple times in different configurations is an explicit expression of the rules of the system. Within the TMC application viewers can re-render a cover. Through repeatedly re-rendering a cover a viewer gets to read not only each render but also the system, gaining an appreciation for the qualities of the set.

Figure 32. A collection of Twitter Modern Classic covers shared via Tumblr. [49]

6.5. Computational Context

By making reference to print design, TMC draws our attention to a beautiful cultural form but also to the programmatic design principles which shape familiar graphic forms like the paperback novel cover. Examining examples of system design from print publishing makes clear the common ground between print and screen with regard to the graphic presentation of dynamic information. For this project's focus on software aesthetics, graphic design's well established interest in both the practicalities and aesthetics of visual systems is highly relevant. Clearly there is some conceptual parity between the practice of page layout and that of screen design; both involve the systematic organisation and styling of text and media within the confines of a 2D space - page or screen. But despite the conceptual continuity between systemised print design and computer interface design, they are far from identical, and developing design aesthetics in the computer context is about much more than simply transitioning concepts from print to screen. That said, the benefit of considering historical precedents of programmatic design is that they help to throw into focus the unique characteristics of the computational context. For example, comparing the production of the

[49] http://twittermodernclassics.tumblr.com

original printed covers to those generated by TMC highlights commonalities and key distinctions between the two mediums.

Both print and computer renderings are based on a system of rules, however print and computer have different ways of encoding and executing the rules. In the case of Tschichold's Penguin covers, his design program was codified as a set of written instructions, a design "Style Guide". In the computer setting, those instructions are recorded as computer executable code. But this difference in encoding formats is relatively insignificant; primarily a difference of syntax. A more important distinction is that of automation. In the case of the original Penguin covers, Tschichold's design instructions were implemented by human operators, who could use their aesthetic judgement in executing the rules (even if discouraged by Tschichold to do so). In the computer setting, the execution of the design rules becomes fully automated, placing all responsibility for the outcome of the production process with the author of the software. The consequence is that the computational instructions need to be far more explicit, not only in defining the ideals but in addressing the edge-cases - the exceptions to the rules. Much effort and energy is dedicated to identifying and accommodating edge-cases and anomalies.

As with all of the creative works included in the project, TMC features an ever expanding set of elaborate checks to determine that each cover displays within the an acceptable range. The networked context adds the asynchronicity of API transactions and complex datasets that need to be verified and processed. In the case of TMC, the execution of the rules and related checks and balances is displayed as a form of dialogue with the audience. It begins by counting the loading of the image feeds, then announcing the loading of tweets, and then the rendering of the covers. Each cover is initially mapped with a dotted line indicating its contour. If images are required, their size must first be checked. This negotiation is conveyed with brief statements within the cover frame: Trying Image 1, Trying Image 2, Loading… Looking to alternate images means that the previous version was rejected for some technical reason. Of course, there are many more specific rule sets for each of the individual cover templates: rules for the text, for layout, for colour, and image manipulation.

Another significant difference is not only who executes the instructions, but when and where. In the case of the original Penguin books, their production would be carried out at an industrial site days, weeks, months or years before appearing

on a bookstore shelf. In the case of the computer program, it is executed at the time it is viewed; it is a performance that may occur in as little as a matter of milliseconds. The producer of the software program knows neither where their program will be executed or by whom (the whom being the computer hardware running the program rather than the user interacting with it). “We no longer deal with “documents,” ”works,” “messages” or “media” in 20th century terms. Instead of fixed documents whose contents and meaning could be fully determined by examining their structure… we now interact with dynamic “software performances.”… what we are experiencing is constructed by software in real time.” (Manovich, 2008)

The concept of a “performance” is naturally associated with characteristics such as time, interactivity and dynamics. In contexts like the web, software must respond to constant changes; things as simple as a shift in the width of a browser window, or as complex as detecting and processing new streams of data. All of which underlines the presence of time. Interactivity also emphasises the significance of time, as well as that of the performativity of software. A software work may engage very lightly with concepts of dynamism, interactivity and time, instead seeking what appears as a relatively static state, but the temporality of software means the characteristics of dynamics, interactivity and time cannot be ignored by designers. As software works engage more fully with these factors, their connection to established graphic print forms is further detached.

The final differentiating characteristic to be considered, is that of computation. Computation provides opportunities for the construction of elaborate rulesets that would not be feasible for human operators. The exercise of computationally rendering existing graphic forms, such as the Penguin book covers, provided a valuable insight into the relative benefits and limitations of human and machine execution of design rules. In many instances, the process of reverse-engineering the covers highlighted practices as well as rules. For example, a common practice was (and still is) to selectively sample a highlight colour from a background image to use on the title text being overlaid. In a computational version, that practice must be codified like any other design rule.

In attempting to decipher the inherent graphic systems, great complexity and subtlety were often revealed. In many cases there was a strong sense that the reason rules are often not formally codified (in style guides, templates, or computer code) is because of the sheer difficulty of the task - firstly, in defining the rules and

their parameters, and secondly in articulating those rules in an accessible form. Instead, the common practice is to let the works themselves serve as the expression of the rules upon which they are based. One benefit of this approach is that the works are capable of representing incredible complexity and subtlety. Another is that they can represent incomplete rule sets, setting a foundation to be built upon by future designers. Attempting to computationally represent even some of the most obvious design formulas gives new appreciation of the sophistication of the rules and the capabilities of the humans that enact them.

6.6. Creative Code

In considering the aesthetics of the everyday there is obvious relevance in established graphic design forms and practices; graphic design is a discipline expert in aesthetic representation and has a long history of association with the everyday cultural context. However, when borrowing from the established practices of graphic design we have to be mindful that while the medium of print and that of computing have some commonalities they are ultimately very different. These differences can be informed by acknowledging the historical origins of current design forms and practices. Those origins involve radical experimentation with print as a cultural medium and sophisticated technology. Artists and designers such as van Doesburg, Itten, Merz, Bayer and El Lissitzky produced seminal design works through their radical experimentation in print (Hollis, 1994). Take for example El Lissitzky's famous children's book "Of Two Squares" which tells the tale of a red and a black square who unite to join the revolutionary cause (Eskilson, 2007). The illustrations, like the story itself, are abstract, with the composition of type and graphics being directly inspired by the phonetics of the words and the narrative of the tale. Considered one of the first works of the "new typography", its Modern aesthetic is tightly connected to the technology of its manufacture.

In looking to develop aesthetic sensibilities in the computer setting we have much to learn from graphic design's print legacy but equally, we need to adopt the spirit of the pioneers who developed the graphic language through ambitious practice-based works. Those designers engaged with print as a cultural and technological medium and today's designers need to approach computing in much the same way. That means engaging with the computer not only as a tool, but also as a medium (Maeda, 2004). This is an approach which is evident in all of my

creative works. It sees me working directly with code as a creative means, as a way to explore new aesthetic possibilities that emerge from the pragmatics of the computational context. I see it as an opportunity to extend the rich legacy of graphic design and this approach has lead to some novel outcomes, where the pragmatics of print and computer collide. In particular, works such as Physics, 3D, Kernit and Motion all represent an intersection of traditional graphic design conventions and the new pragmatics of the computational context.

Of course my computational approach is not unique and can be situated within a growing field of practice. Sometimes referred to as the "Creative Code" scene, it is comprised of practitioners from diverse art and design disciplines whose point of union is their use of code as a creative means. The recent text Form+Code (Reas & McWilliams, 2010) describes the Creative Code view: "Once the exclusive domain of programmers, code is now being used by a new generation of designers, artists, and architects eager to explore how software can enable innovative ways of generating form and translating ideas." (Reas & McWilliams, n.d.) If the "aesthetic turn" represents HCI's efforts to transform itself into an aesthetic practice, "creative code" represents the "computational turn"; the efforts of creative practices to incorporate the computational.

Whether for instrumental or creative aims, producers choose to engage in creation of their own software in order to gain sufficient control over the artefact being produced. Put another way, crafting their own code frees a producer from the constraints of existing production software. The irony is that the "control" they most often seek is a form of non-control - it is the ability to use the computer in a generative way to produce new and unexpected outcomes rather than within the confines to the tool-like software packages which are today a staple in art and design practices. Take for example Field's work for the paper manufacturer GFSmith.[50] Field generated 10,000 unique "digital paintings" for inclusion in GFSmith's paper sample package. Each page featured a different view of a "hypercomplex sculpture" - a vividly coloured composition of intersecting voids, each facetted with intricate folds and fissures. The novelty of this project is Field's system for generating unique images. Inherent in that system is a relinquishing of control - the exact form of the images cannot be predetermined. As Marcus Wendt of Field explains: "Instead of working towards a single image, you start to think in the possibilities of a system. Designing a process rather than the end result forces

[50] 10,000 Digital Paintings http://www.field.io/project/digitalpaintings

you to be open for and work with unexpected results, and sometimes surprisedly embrace the outcome." (Reas & McWilliams, 2012)

Wendt's suggestion of embracing the unexpected refers to a defining characteristic of the Creative Code scene. Whether for open-ended experimentation or for a commercial product, programming is viewed as a reflective iterative investigation, as Erik van Blokland describes: "Write, generate. Evaluate. Tweak (code, parameters or both), generate again, repeat. Design iterations are code iterations, the code is as open ended as the design." Van Blokland and Just van Rossum of Letterror have employed computational methods in their type design since the early 1990s. The pair came to prominence with their 1990 typeface "Beowolf" - a randomised font which is rendered uniquely for every print. The Beowolf font family has varied levels of randomness where a conventional font would have weights. Beowolf 20 is relatively conventional looking, whereas Beowolf 24 distorted and jagged. The font was discovered through chance while attempting to develop a smooth rounded typeface[51]. For van Blokland, programming is akin to sketching, with the designer constantly evaluating the qualities of their output in a rapid iterative cycle[52].

Complimenting van Blokland's view, Julia Laub and Cedric Kiefer of "onformative" design studio describe the centrality of programming to their practice:

> *"The design process is no longer divided into concept, design, and production, but rather the design process and the product are created in many small iteration steps in which idea, design, and programming are always closely entwined. When writing one's own software, the creative work and the implementation of these are mutually dependent, and the separation of design and production is abolished. Because one has very different insight into the working methods and detailed processes of one's own software, there is far more room for experimentation at one's disposal, which has a direct effect on the quality of the work."* (Reas & McWilliams, 2012)

---

[51] http://www.wired.com/wired/archive/3.07/letterror.html

[52] http://www.typotheque.com/articles/letterror_designers_and_programmers

This process is especially evident in onformative's software works. For pharmaceutical company Actelion they produced an "Imagery Wizard"[53] which demonstrated Onformative's philosophy of embracing software's variability. Rather than a set of fixed brand images, the onformative's Imagery Wizard enabled the generation of countless variations, all within the bounds of the defined corporate styling.

Clearly for these practitioners production is not expository, it is a process including discovery and surprise. This production process requires producers to remain open to the possibility of happy accidents as Pierre Rodière and Joël Rodière of design company Trafik[54] explain: "the code, by the complexity and diversity of what it can produce, manages to surprise us and to go beyond what we imagined at the outset." (Reas & McWilliams, 2012) Whereas the members of Letterror and onformative are designers who code, Trafik's staff are dedicated designers or programmers. Despite this delineation in roles, their process is based on a tight integration of the two domains: "By using programming, the creative process seems to us to be more complete… The project builds itself throughout the development, through a permanent exchange between suggestions from the graphic designer and those of the programmer." (Reas & McWilliams, 2012)

While there is a recognition and celebration of the generative and performative nature of software, there is a strong sense of the producers' need to retain an authorial control over the quality of the outcomes. This level of aesthetic control is clearly evident in the work of Nicholas Felton. Felton is renowned for his Feltron Annual Report, in which he transforms the banality of daily life into an exquisitely designed series of info-graphic posters. The reports collate data representing a selection of arbitrary and often unexpected subjects - sick days, average temperature, pedometers purchased, least expected menu item, etc. Felton has embraced programming as an important part of his production process as it allows him to work with more complex data and to experiment freely with different visualisations, but he also emphasises that the code is "accountable", meaning that he is the ultimate arbitrator of an output and that he will mould his code to achieve his aesthetic objectives. This need for discernment is summed up by Boris Müller: "beautiful ideas do not necessarily generate beautiful images". Like Felton, Müller is perhaps best known as the producer of an annual design - which in Müller's case

---

[53] http://www.onformative.com/work/actelion-imagery-wizard/

[54] http://www.lavitrinedetrafik.fr/index.php?wid=1372&lang=en

is the visual theme for a literature festival called “Poetry on the Road”[55]. Since 2002 Müller has followed the same strategy for generating the theme: he creates a computer program which generates the imagery based on the texts of the festival. Müller produces striking visuals with a strong graphic sensibility - minimal colour and bold forms, and perhaps of equal importance is the narrative of his imagery. Each visual theme is accompanied by the rationale of the generative system which produced it. Often, in addition to the printed materials for the festival, Müller will produce a software version which allows users to play with the system to gain a greater insight into the relationship between form and text.

While most of the Creative Code examples cited here focus on graphic practice, the computational turn is radically reshaping all forms of creative practice, whether screen or material based. And practitioners within those different fields have the same challenge as graphic designers; to adapt and evolve their established practice for the computational context.

From the various opinions expressed here there is an implicit understanding that creating software is a material practice conducted through code. This is certainly the view that I derive from my creative practice and with it, there is an understanding of an aesthetic that is informed by and bound to the computational; that emerges from working with code.

6.7. Summary

With its explicit reference to paperback novel covers, TMC motivated a deeper consideration of the connections between system design practices for print and those for software. Looking to established print forms and practices provided a valuable historical pretext to contemporary computational design practices. The print examples also gave some insight into the nature of system aesthetics, in which the structures of the system and the qualities of the sets it produces serve as aesthetic frameworks.

[55] http://www.esono.com/boris/projects/

Having shown some of the benefits of connecting traditional practices with the new contexts of computing, it was argued that designers must also look to the future and engage with the computer, not only as a tool, but as a creative medium. Looking to the field of Creative Code and my own production practice, software was shown to be viewed as a material medium to be creatively explored through code.

We have two different but compatible aesthetic perspectives; graphic design gives us a foundation for system aesthetics while the creative practitioners (including this author) refer to an aesthetic which emerges from a creative engagement with code.

Data as Script

7.1. Introduction

Some of the key barriers to producing new poetic forms of software relate to fundamental conceptions of what software is; what it does, how it does it, and why. These fundamental conceptions are derived from, and are evident in, the conceptual models used to define the contexts, artefacts and production procedures for software. In the process of conducting the creative practice for this research project, the benefits and limits of conventional conceptual models have become apparent. The aim of this chapter is to respond to these limitations and offer an alternative conceptual framing which promotes a creative and aesthetic engagement with software production.

The chapter examines two different models and considers the implications of their respective conceptual framings. In an effort to better represent the network context of my creative works, I extend the second model to develop a third. As with the previous models, the analysis considers both the practical and conceptual implications of the framing, and in particular, their impact on production.

Given that this research has argued strongly against reductive and deterministic approaches to the production of computer technology, it would be reasonable to wonder why it is now dedicating significant attention to the crudely simplified models presented here. The simple answer as Blythe et al explain: "Representations are always reductions; but they are not necessarily dangerous… Models, theories, reductions can be helpful and necessary, but they are neither the reality nor are they neutral or objective. Being aware of this is crucial" (Blythe et al., 2009).

The goal is not to provide a comprehensive picture of experience with technology, but only to devise some simple instruments for helping to conceptualise the situation. As Schön explains, models and metaphors are instrumental in this process: "Each story constructs its view of social reality through a complementary process of *naming* and *framing*. Things are selected for attention and named in such a way as to fit the frame constructed for the situation. Together, the two processes construct a problem out of the vague and indeterminate reality which Jon Dewey called the "problematic situation." They carry out the essential problem-setting functions. They select for attention a few salient features and relations from what would otherwise be an overwhelmingly complex reality. They give these elements a coherent organisation, and they describe what is wrong with the present situation in such a way as to set the direction for its future transformation" (Schön, 1979).

The danger of reductive models is in applying them prescriptively rather than suggestively. The models outlined here, including any advanced by this research, are not static truths but artefacts of an ongoing effort to respond to new understandings and contexts. Agre argues that model-building should be understood as a discursive and "ongoing enterprise, proceeding through the incremental, generative application of metaphors to new issues and questions" (Agre, 1997).

It is in this spirit that we observe existing models and metaphors, and offer adaptations and new framings. Understanding the models as provisional and propositional, we can appreciate that while crudely simplified, the models serve their basic function; to outline a conceptual model for the production of computer software.

7.2. Mental Models

The first model to be examined (Figure 33) is widely cited in HCI research and literature and describes the concept of “mental models”. The concept was popularised in Norman’s text, “The Design of Everyday Things” (1988). Within usability/HCI fields, the concept describes a computer artefact as a form of semiotic system in which the designer is author, the user is reader, and the “system image” is the text.

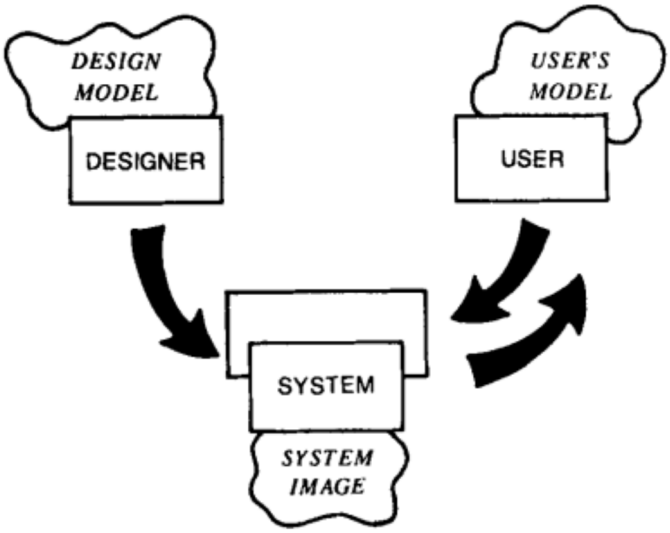


Figure 33. A diagram of the System Model model (Norman, 1988)

In the System Model (Figure 33), as in any text, the user’s reading of the System Image is open to interpretation, which in HCI terms is problematic. In the case of a written text, a reader may form an understanding that is different to the understanding the author intended which, depending on the nature of the text, may not be an issue, or may have resulted from the author using ambiguous prose for intended effect. In the computing context, the concern of authors such as Norman is that because a user’s understanding of a system directly impacts their ability to operate it effectively, an alternate reading will likely result in detrimental outcomes. To combat this possibility, the System Model emphasises the need to ensure an appropriate match between the designer’s and user’s understanding of a system; their respective “mental models”. The designer’s challenge is to create a “System Image” (typically a graphic user interface) that conveys to the user an appropriate “mental model” of the system - what it does, how it functions, etc.

When completed successfully, the act of translating an abstract computer system into tangible form is a remarkable technical and cultural feat. Perhaps the greatest example of this is the original desktop GUI, Doug Engelbart's "NLS" ("oN Line System"). Developed at Stanford Research Institute's "Augmentation Research Centre", Doug Engelbart's famous 1968 demonstration has come to be known as the "Mother of all Demos"[56]. The extraordinary demo introduced many of the core concepts of modern computing including the mouse, cursor, point-and-click operations, hypertext, dynamic file linking, networking, email, and live tele-conferencing. As Johnson writes; "Historians a hundred years from now will probably accord it the same weight and significance we now bestow on Ben Franklin's kite-flying experiments or Alexander Graham Bell's accidental phone conversation with Watson. Engelbart's thirty-minute demo was our first public glimpse of information- space, and we are still living in its shadow" (Johnson 1997, pg. 11).

The propositions of Engelbart's team were so revolutionary, on both technical and cultural fronts, that after its 1968 debut it would take another 15 years and the dedication of Xerox and Apple to bring the ideas to the marketplace.

Engelbart's system involved development of hardware and software, but as computing has become standardised (CPU, mouse, keyboard, screen), the role of the designer portrayed in the System Model has focused on development of the software that constitutes the System Image. As already stated, the difficulty in producing the System Image is in translating an abstract conceptual model of a computer system into appropriate graphic and interactive form. If the System Image is too literal a representation, it will be too complex for the user to conceive and operate. If on the other hand, the System Image is over-simplistic, it will prevent the user from comprehending the full functionality of the system. Norman's "Design of Everyday Things" (1988) elaborates on many principles and strategies informing decisions about what a System Image should reveal, how and when. It portrays an explicit tension between ease and complexity; between conveying a simplified but accessible model versus a detailed but difficult form.

As the medium of computing matures, and the literacy of both designers and audiences improves, the burden on the designer is reduced. This evolution is evident in all other major forms of communication media; print, cinema, TV and to

---

[56] http://sloan.stanford.edu/mousesite/1968Demo.html

some extent even radio (Manovich, 2001). What we find in those media, and increasingly in computing, is that as the technical sophistication of the medium improves and its cultural conventions stabilise, producers and audiences develop a shared understanding of the medium (Wright, Wallace, & McCarthy, 2008). In TV and cinema for example, genre frameworks are common conventions recognised and used by both producers and audiences. Producers use genres as a framework to guide their production and can combine conventions from different genres for innovative effect. Audiences use genres to inform their expectations and as a framework for reading a work. In some sense, genre conventions serve as predefined mental models. In a computer context, familiar conventions and genres are also developing. Take for example the desktop operating system; when Engelbart's team first demoed their system the concepts were utterly alien but today are familiar to the point of ubiquity. With regards to the System Model (Figure 33) conventions, genres and other frameworks for shared understanding greatly simplify the problems of communication for producers and users alike.

7.3. The Improvisational Model

The second model to be considered is Ishizaki's Improvisational Model (Figures 34 - 38). Ishizaki offers us a more complete model which accommodates various additional components to better represent a contemporary networked context. Starting with a model (Figure 34) resembling Norman's classic System Model, Ishizaki progressively adds complexity. The single user is transformed first into a mass audience (Figure 35) and then into multiple individuals (Figures 36 & 37). The content that the system displays is recognised to be dynamic, indicative of a networked context. "Media" is defined as a discreet entity, separating it from the "Design System". This distinction recognises that a single Design System needs to output an increasing variety of media forms. Media in this model may indicate a variety of types (print, tv, web, etc) or just a variety of forms within one type (desktop, mobile, tablet, etc).

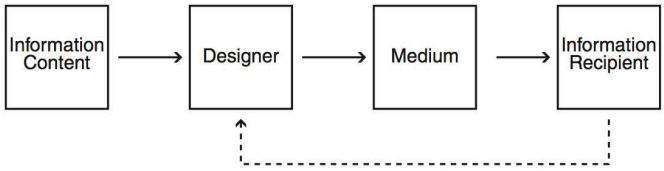


Figure 34. Ishizaki, 2003

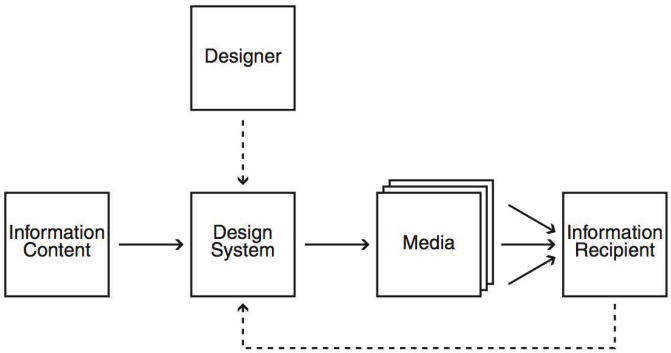


Figure 35. Ishizaki, 2003

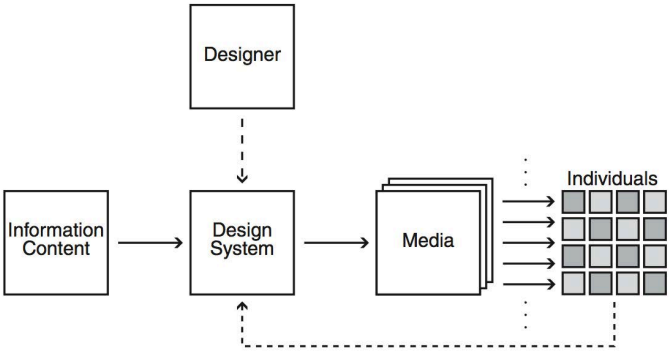


Figure 36. Ishizaki, 2003

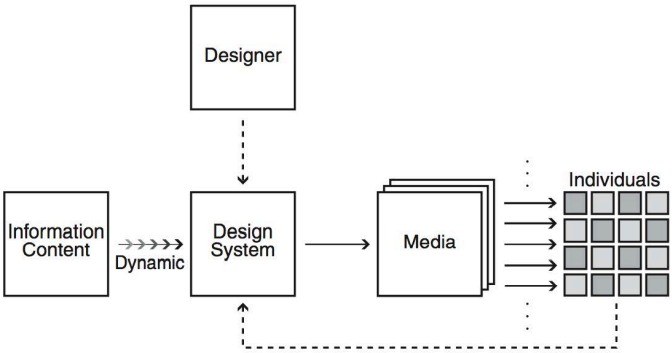


Figure 37. Ishizaki, 2003

Ishizaki's final model (Figure 38) also recognises that the individual audience members can customise the choice of content they receive. In the example of a news service, this would see readers choosing feeds; latest headlines, sport, politics, entertainment, etc. In a final iteration (not show), Ishizaki replaces Media with "Situations" to indicate new forms of ubiquitous computing such as smart objects and environments. Ishizaki's version emphasises dynamism; content is dynamic but so too is the Media context and the design system has to be adaptable to both.

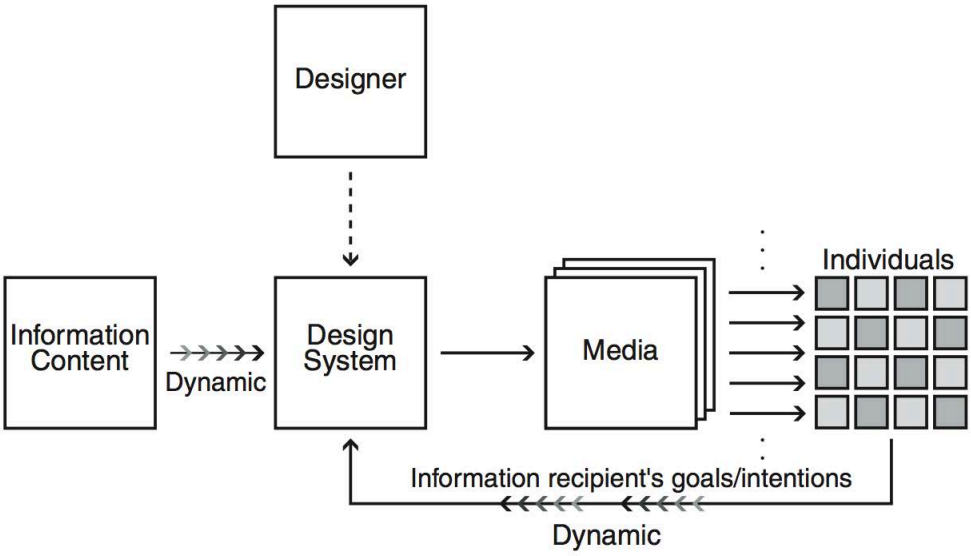


Figure 38. Ishizaki's Improvisational Model

7.4. The API Model

By introducing an increasing degree of dynamism, Ishizaki's model makes a truer representation of a conventional networked media context, however, it can be extended even further when we adapt it to the network context being considered in my creative works for this project; as represented in Figure 39 where I have adapted Ishizaki's Improvisational Model (Figure 38). The API Model (Figures 39 & 40) is the third and final model to be considered.

The API Model (Figure 39) has some distinct variations on Ishizaki's Improv Model (Figure 38). In the social media context, individuals act as both information recipients and as content authors. For example, in social media systems people contribute to the Information Content as well as retrieving it. Another critical addition is the API; with the advent of the networked "application programming interface" (API), the Content System is separated from the Design System. In Ishizaki's Improvisational Model (Figure 38), the Design System had direct access to the Information Content. This is typical of a situation where the Information Content and Design System are the property of a single entity. However, in a social media context, access to the Information Content is routed via the API, which dictates the conditions of access to the Information Content. The consequence of the addition of the API is the potential for multiplication of Designer and Design System; third party designers are able to construct their own Design Systems.

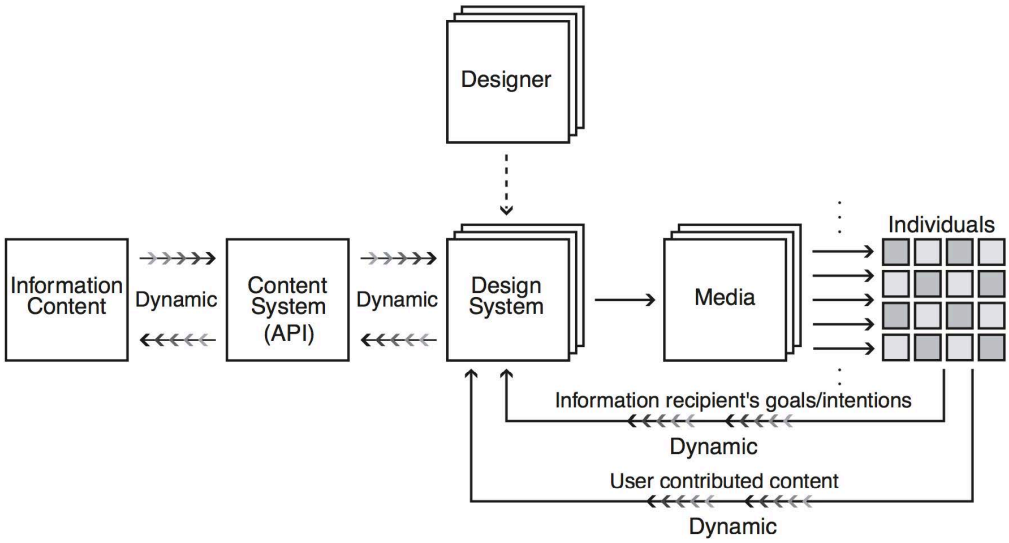


Figure 39. An augmented version of Ishizaki's model (Figure 38) which separates the Content System (API) from the Design System, permitting multiple Designers and Design Systems.

The API split makes explicit the distinction between application and interface. In traditional desktop software this delineation has not been apparent, with graphic interface, application logic and data all created by the same author/company and presented as a single definitive view. By providing access to application data and logic, APIs enable and encourage independent designers to create their own custom representations (Design Systems). Of course, it is also possible for a

Design System to access and aggregate content from multiple APIs. Representing this additional evolution, the following diagram features a Design System interacting with multiple APIs. Note that one of the APIs is a “scrape” of a Design System. Scraping refers to the process of extracting information and data from a Design System (typically a webpage) instead of an API. Scraping was actively pursued in the mid 2000s but is today discouraged and is largely unnecessary due to the widespread availability of RSS feeds which act as a simple API, offering access to streamlined content.

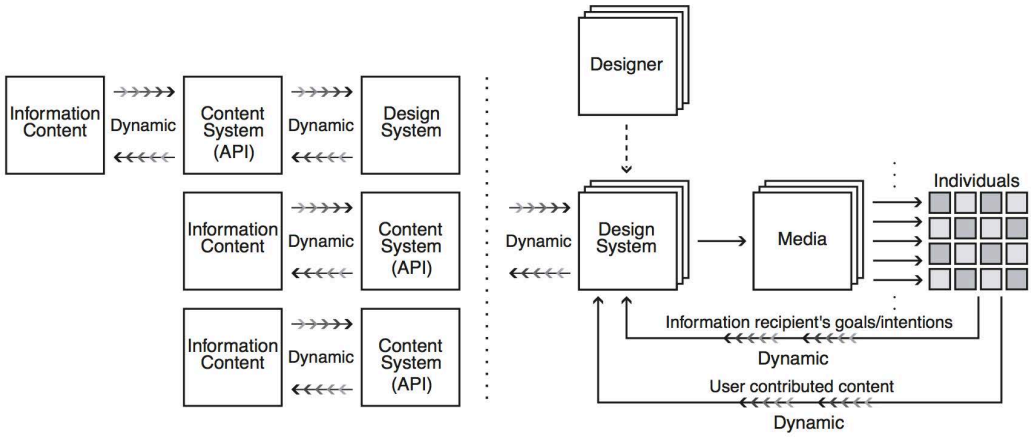


Figure 40. The API model. Created by the author based on Ishizaki’s Improvisational Model (Figure 38)

7.5. Three Models

The System Model (Figure 34) presents an intentionally stripped down version of computer interaction in order to emphasise its agenda of prosaic communication: ensuring a user comprehends a system in the intended way. This clear and simple objective is also reflected in the production and evaluation methodologies favoured in HCI, in which user evaluation (checking that the user did indeed get it “right”) is the dominant form. Ishizaki’s concerns in his Improvisational Model (Figures 33 - 37) are more logistical than semiotic. The Improvisational Model introduces additional complexity, with the components and relationships within the system more accurately reflecting the network model employed by many contemporary publishing entities, particularly within the news media space. Rather than the emphasis on matching Designer and User’s mental models, the focus of Ishizaki’s model is primarily logistical; how can a designer accommodate the

increasing dynamics of the network context? His answer is 'improvisational design systems' with the ability to adapt to continuously changing information content and individual user customisations. A further interest of his study, which is not clearly evident in the model diagram, is the capability of temporal presentation, which he sees as an integral component in any attempt to address the dynamics of content and customisation (Ishizaki, 2003).

Ultimately Ishizaki's model gives us a more detailed representation of networked computing, but with similarly prosaic ambitions as the System Model. Certainly if our focus is on designing interfaces for complex utilitarian software applications, or aggregating and representing information such as news, then prosaic aims are appropriate. However, as we consider the everyday, the nature of the content and context cannot be assumed to be prosaically oriented. "Cultural data" (Manovich, 2001) includes reference to all forms of culturally generated data, from websites to YouTube videos, email, text messages, music services like Rdio and Spotify, and social media services like Twitter and FaceBook. In a technical sense, digital data is generic, but the information that it represents will have specific cultural values. The simple point here is that increasingly there will be instances where more aesthetic and poetic representations will be appropriate, and even required.

In my proposition of the API model (Figures 39 & 40), I have attempted to better define the networked context in which my creative works are located. The model introduces the additional complexity afforded by networked APIs, recognising the data they produce as being discreet from any realisation of it. The model can be regarded as a form of network documentation but, as with the other models discussed, it serves as a conceptual base, directly shaping the fundamental conceptions that dictate the aims, objectives and final forms that software takes. In that regard, it shares some of the pragmatic concerns of Ishizaki's Improvisational Model but I will argue that its inherent structures bring a very different emphasis to the role of designer and to the form and function of the artefacts they produce.

7.6. Data as Script

The System Model (Figure 33) suggests an analogy of "system as text" which is suitable for its focus on the user's "reading" of the system. In the second model, Ishizaki's Improvisational Model (Figures 34-38), the interest in the user is with regard to their actions; what customisations they make to the Design System. The model pays equal attention to the characteristics of the network, emphasising the scope for two-way, dynamic flows of information. Ishizaki adopts an analogy of "improvisation" based on "the nature of performing arts, such as dance and music" (Ishizaki, 2003) and writes;

> *"In the model of improvisational design, a design solution is considered a performance consisting of a number of design agents (performers), each responsible for its own role in the design. A design solution as a whole emerges from the collaboration of design agents, just as a jazz improvisation does. The individual design agent is also sensitive to the change in its immediate context (information and the reader's intention) and can respond to it accordingly" (Pg. 21).*

In the social media context of my creative work, the analogy of improvisation offered by Ishizaki continues to resonate and is possibly intensified with the growing number and increasingly complex nature of the information sources, designers and design systems. However, the issue with the analogy is that the sole intent of the designer is still focused on a facilitation of user access to the required information content. The argument presented repeatedly throughout this thesis is that transparent facilitation is not always enough, but also, in the case of the data presented by an API, it is not necessarily in a suitable form for a direct and transparent transmission. Typically, the data returned by an API will suggest a variety of applications and possibilities for integration and presentation. Another factor that strongly influences any analogy or metaphor is the independence of API and Design System, as well as the fact that the API is in itself a designed artefact. To account for these two distinct components, and their levels of authorial autonomy, I propose the analogy of "data as script".

Crucially, the script analogy recognises that the data produced by an API is a designed artefact in its own right and as such presents its audience (software developers) with propositions of what it is and how it might be used. In order to

illustrate this point, let us examine a typical dataset produced by the Twitter API (Figure 41).


```
20.      "text": "just another test",
21.      "contributors": null,
22.      "id": 240558470661799936,
23.      "retweet_count": 0,
24.      "in_reply_to_status_id_str": null,
25.      "geo": null,
26.      "retweeted": false,
27.      "in_reply_to_user_id": null,
28.      "place": null,
29.      "source": "<a href=\"http://realitytechnicians.com\" rel=\"nofollow\">OAuth Dancer
     Reborn</a>",
30.      "user": {
31.        "name": "OAuth Dancer",
32.        "profile_sidebar_fill_color": "DDEEF6",
33.        "profile_background_tile": true,
34.        "profile_sidebar_border_color": "C0DEED",
35.        "profile_image_url": "http://a0.twimg.com/profile_images/730275945/oauth-
     dancer_normal.jpg",
36.        "created_at": "Wed Mar 03 19:37:35 +0000 2010",
37.        "location": "San Francisco, CA",
38.        "follow_request_sent": false,
39.        "id_str": "119476949",
40.        "is_translator": false,
41.        "profile_link_color": "0084B4",
42.        "entities": {
```


Figure 41. Extract of Twitter API data.
(source: https://dev.twitter.com/docs/api/1.1/get/statuses/home_timeline)

To give some context to this dataset requires a very brief description of the Twitter API. Twitter is a popular micro-blogging platform in which its users post short status updates of 140 characters or less. In addition to posting status updates, users follow other users and receive an aggregated feed of tweets from their network. In simple terms the Twitter API is a web-based application that receives requests and returns relevant data. It provides a non-graphical interface to almost all of Twitter's data and functionality. A typical API transaction would see a client application requesting the 20 latest tweets of a user and their network of friends. The Twitter web servers would process the request and return any matching data as a structured list. The client application would then parse the data, format it, and present it to the user. The image above is an example of the data returned by the Twitter API.

As can be seen in Figure 41, the API data can be read in its raw form but the intention is that it will be interpreted, reformatted and presented to the user. The notion of creating a work to be mediated by independent producers rather than published for a general audience aligns the API conceptually with the script (screenplay/play).

"Script: the working text, manuscript, or the like, of a play, film, television program, etc." (ed. Yallop, 2004).

While both script and API are created for producers rather than as a finished product for a general audience, neither provides elaborate specifications for how a production should be realised. Typically, the script for theatre or cinema will include only rudimentary suggestions regarding its production. However, it is not entirely without context, and will draw on established conventions implicitly. For example, even if not explicitly stated, a theatre script will assume the existence of actors and audience. Instead of detailed descriptive prose the script writer uses subtext, action and symbolism to indicate a character's motives and emotions. Michael Holt describes the theatre designer's reading of a script as a forensic process in which they must identify significant settings, objects and costumes whether specified or implied:

> *"Some details are hidden in the text. They may be revealed only in the dialogue, when a character refers to them. Sometimes they are there solely by inference, not actually mentioned, but nonetheless needed for the dialogue to work in action. Great care is needed to identify these items and decide how important they are" (1988).*

In applying the script analogy to Twitter's API, it is tempting to correlate Twitter users with characters, and status updates with dialogue. Certainly this correlation could yield some interesting results, but it may not hold when considering non-social and non-verbal data, such as times/dates, urls, location data, etc. In any case, as is evident in the API Model, the source generated by Twitter's API is dynamic. Each successive set of tweets contains different status updates, as well as different users with different interrelations. Producers for theatre or cinema will drill down to the semantics of individual words in the process of analysing a script, and clearly this is not possible when the script is being generated dynamically. Instead, designers must focus on the constant elements within the script: its structure and data types. Within the data produced by the Twitter API (Figure 41) the structure and data types are clearly identifiable nodes with corresponding data: the "text" node contains the status update; the "created_at" node contains the creation date; the "source" node lists the application used to post the status; the "user" node contains a subset of nodes with information about the author of the update; etc. So while the exact data that each node will contain is unknown to the

producer, the structure and node types are consistent, and form the basis of the script.

In suggesting the script/screenplay analogy the aim is not to dwell on the mapping between script and API data; actors, props, lights, curtains. The emphasis here is on the API as an artefact designed to be interpreted and realised by another party. The analogy could therefore just as easily be “system as score”, in the spirit of the musical composition. And while both score and script have a strong performative inference, the script analogy implies a sense of material production that resonates with software production.

7.7. Creative Interpretation

The concept of “data as script” proposes the script as a foundation structure rather than a complete blueprint, requiring interpretation and adaption in order to be realised in a mediated form. In addition to shaping production, the notions of interpretation and adaptation suggested by the “script” analogy are increasingly evident in the critical evaluation of software. To explore this proposition, we examine reviews for a leading Twitter client application[57] and make some general comparisons to reviews for a theatrical stage production where the tradition of interpretation in production is widely recognised and celebrated.

In his production of Shakespeare’s “Richard III” for London’s Old Vic Theatre, Sam Mendes’ realisation “steers a clear course through the internecine drama. It is in modern dress but unspecific about its exact setting, which allows Spacey to channel touches of dictators through the ages” (Lawless, 2011). The setting “reminds us how today's dictators seek spurious constitutional legitimacy and become skilful media manipulators” (Billington, 2011). Alternatively, the Telegraph reports that while the production is “fluent and lucid, it lacks the striking invention and disconcerting dreamlike atmosphere of Edward Hall’s production now running at Hampstead. Like Tom Piper’s architectural design of receding walls and doors, Mendes’s staging feels a touch obvious and over-deliberate, leaving little room for the audience to let their own imaginations soar” (Spencer, 2011).

[57] TweetBot application by TapBots http://tapbots.com/software/tweetbot/

From these commentaries we are able to appreciate the practical and poetic significance of the producer's interpretation of the script. With its unspecific, yet modern setting, Mendes' production amplifies themes within the script and allows his audience to connect a centuries old text with contemporary characters and events. And while it may seem incongruous (if not dangerous) to relate a Twitter interface to a Shakespeare production, reviews of the TweetBot app have commonalities with theatre reviews listed above, particularly in their framing of the interface as a novel realisation of a familiar source, as in Lofte's review entitled "Tweetbot once again redefines Twitter for iPhone". Consistent amongst the many TweetBot reviews is a strong appreciation of its interface aesthetics:

> *"The interface is what stands out the most about the application. This is something we have come to expect from Tapbots, who have a reputation for crafting gorgeous looking applications" (Naik, 2011).*

> *"From Tapbots – the creators of Convertbot, Pastebot and Weightbot – Tweetbot promises to be a full-featured Twitter client with a lot of personality; boasting a meticulously crafted user interface, smart gestures, and customizable navigation" (Bell, 2011).*

> *"There is this addictive cleverness and playful uniqueness to the way Mark and Paul (the app designers) build their apps. The sounds, the animations, and graphics don't feel or act like a standard app, they feel more like a toy. A toy you get to use for work" (Blanc, 2011).*

The positive appraisal of the interface's playful antics is typical, as is the identification of the designers of the work. Another key aspect of the reviews for both Richard III and TweetBot is the highly contextual nature of the evaluation. In the reviews of Mendes' Richard III the production is understood within a London Shakespearean theatrical context, while the reviewers of TweetBot write for an audience already familiar with a host of existing mobile Twitter applications. In both cases, the works are assessed as independent productions and as critical contributors in their respective fields. The reviews not only celebrate the works, but also explicitly recognise the agency of the producers and their creative abilities in interpreting and adapting a source.

7.8. Summary

This chapter began with the discussion of three different models of computer systems. In the initial model (Figure 33), the focus was on a highly prosaic communication between designer and user, with the computer interface acting as the “text”. In the second (Figures 34 - 38), Ishizaki progressively elaborated on the first model and ultimately built it into a more complex system based on dynamic networked communication. Ishizaki’s focus was on the logistics of designing in a highly dynamic networked context. In the final model (Figures 39 & 40), I added further complexity, representing the presence of multiple APIs on the network. My aim in extending the model was to consider both the logistical and the conceptual implications. On a logistical level the addition of APIs introduced greater complexity with options to integrate and aggregate sources. On a conceptual front its separation of application and interface created a situation which invites and rewards creative interpretation and adaption of a source.

Suggestions of interpretation and adaptation are unremarkable in most creative disciplines but are novel within the computing context because of the historically deep seeded assumptions of prosaic transmission. This emphasis on the prosaic was understandable in the early years of computing, when the focus of both developers and users was attaining basic competency of operation. It was prioritised in that context because the dominant software form was the desktop application. But the prosaic approach makes less sense today, when a fundamental computer literacy has been achieved and the nature of computing is no longer dominated by the application as a tool. The computer is no longer the sole domain of engineers, it is today the medium of film makers, musicians, writers, graphic designers and countless other disciplines. It is what Manovich calls a meta-medium (Manovich, 2009).

Looking at reviews for a popular Twitter client application (TweetBot), it was shown that notions of interpretation and adaptation suggested by the “data as script” analogy are widely understood and used as evaluative criteria. The concept of “data as script” attempts to challenge presumptions that the interface need be an entirely prosaic translation of a computer system. Instead, the “data as script” analogy promotes the active and creative interpretation of a source, and its imaginative realisation through the process of production.

Conclusion

In this final chapter I reflect on some of the most significant contributions of the research, particularly with regard to understanding the poetics of everyday technology.

The first and most obvious way that the research has contributed to understanding of software poetics is through the production and publication of the body of creative work. The works represent a material investigation into software aesthetics and bring a production perspective to the discourse, providing fine-grained details of how broad, abstract concepts such as aesthetics and poetics may be reconciled within the highly specified and technical context of software production. Each of the creative works presents a novel conception of the Twitter service; interpreting and adapting the common Twitter source for poetic, comic and critical effect. The works have been widely viewed via the web and some exhibited formally. As importantly, they have instigated discussion and reflection not

only about Twitter but about our everyday engagement with software[58] (Figure 42). As a collection of creative works they represent an original contribution to the design discourse regarding the current and crucial topic of everyday aesthetics.

Figure 42. A sample of tweets responding to an early iteration of Twitter Modern Classics.

In addition to their novel aesthetic forms, the works include some innovative technical approaches and processes which constitute another form of knowledge contribution. Innovations such as using a physics engine with html elements; rendering browser elements to bitmap for uploading/downloading; developing vertical grid techniques; using css styles for 3D depth of field; using collision detection for optical kerning of lettering; etc. All of these technical solutions constitute knowledge arising from the creative practice. Additionally, software code can be published[59], representing another means of knowledge dissemination.

---

[58] Much of that discussion has occurred via Twitter itself as evidenced in the Twitter excerpt pictured in Figure 42. Other examples include Greg. J Smith's discussion of whimsy: http://web.archive.org/web/20100410234859/http://serialconsign.com/node/247 Whitelaw's discussion of dynamic design: http://teemingvoid.blogspot.com.au/2011/03/dynamic-design-three-systems.html

Desktop magazine's interview with the author: http://desktopmag.com.au/features/interview-mitchell-whitelaw-geoff-hinchcliffe-part-one/ and http://desktopmag.com.au/features/interview-mitchell-whitelaw-geoff-hinchcliffe-part-two/

[59] The Kernit class is an example of code being packaged and distributed independently. http://gravitron.com.au/kernit

The body of creative works was not developed as a categorical solution but as a playful enquiry into an enormously promising proposition; that of software as a medium of aesthetic experience. However, as the outcomes of the research demonstrate, to capitalise on this promise we need to look beyond norms to what Dunne refers to as the “post-optimal”. Rather than constantly reaffirming the familiar, designers need to build on an audience’s existing cultural knowledge of computing, and invite them into new engagements. This will require designers and developers to work speculatively, to follow hunches and seek out unconventional solutions. Perhaps most importantly they will need to design for a discerning and critical audience rather than passive users seeking the easiest option.

The speculative is quite alien to traditional HCI research but ideally suited for poetic aims. Rather than the normalising effects of standard user-testing, the speculative approach centres on the design discourse and aims to develop novel propositions as a form of critical contribution, albeit in material form. The speculative approach is particularly relevant in the web context where innovative ideas can be easily disseminated and diffused into larger markets.

Having demonstrated contribution in aesthetic and technical forms, and methodologically through the advancement of the speculative approach, the research contributes to greater understanding of poetics in software production. Through the process of creative and critical enquiry the research developed a conception of software as a material medium to be creatively explored through code.

The creative development gave rise to many ideas and issues but perhaps most fundamentally it suggested the invention of a different analogy for the production of software as an aesthetic and poetic form. The model of “data as script” was proposed as a more fitting description for the role and responsibilities of the producer in a creative production context. In that context, data is not a “text” to be transparently transmitted but a source to be analysed, interpreted and imaginatively realised. The “data as script” analogy foregrounds the concepts of interpretation and adaptation that were integral to the production of the creative works completed for this project.

In conclusion, the research contributes to greater understanding of the poetics of everyday computing in a number of significant ways: it presents a body of creative work that makes original contribution on both aesthetic and technical fronts; it advances understanding of the speculative production processes that are so closely associated with innovative poetic forms of software; and it develops a novel conceptual model to emphasise the practices of interpretation and adaptation which are integral to poetic software forms.

References

Agre, P, E. (1997). Computation and Human Experience. Cambridge University Press.

Akrich, M. (1992). The De-scription of Technological Objects in Shaping Technology /Building Society, Weibe E. Bijker and John Law, eds. MIT Press.

Antonelli, P. (2011, July 31). States of Design 04: Critical Design - Design. *domus*. Retrieved November 22, 2012, from http://www.domusweb.it/en/design/states-of-design-04-critical-design-

ArsElectronica. (2012, June 5). “ZEITRAUM” AT VIENNA INTERNATIONAL AIRPORT’S NEW TERMINAL. *aec.at*. Retrieved December 1, 2012, from http://bit.ly/K8PT1v

Baines, P. (2006). Penguin by Design: A Cover Story 1935-2005. Penguin Books.

Bardzell, J. (2012). Commentary on: Tractinsky, Noam (2012). Visual Aesthetics: in human-computer interaction and interaction design. (M. Soegaard & R. F. Dam, Eds.) *Encyclopedia of Human-Computer Interaction*. Retrieved June 28, 2012, from http://www.interaction-design.org/encyclopedia/visual_aesthetics.html#jeffrey+bardzell

Bardzell, J. (2008, July 21). Criticism = Design = Criticism | Interaction Culture. Interactionculture.Wordpress.com. Retrieved December 13, 2010, from http://interactionculture.wordpress.com/2008/08/21/criticism-design-criticism/

Battarbee, K. (2003). Defining co-experience (pp. 109–113). Presented at the DPPI '03 Proceedings of the 2003 international conference on Designing pleasurable products and interfaces, ACM. doi:10.1145/782896.782923

Bell, K. (2011). Tweetbot for iPhone Deserves its Release Day Hype, Cult of Mac, http://www.cultofmac.com/tweetbot-for-iphone-deserves-its-release-day-hype-review/90259

Ben-Bassat, T., Meyer, J., & Tractinsky, N. (2006). Economic and subjective measures of the perceived value of aesthetics and usability. *Transactions on Computer-Human Interaction (TOCHI*, *13*(2). doi: 10.1145/1165734.1165737

Bertelsen, O. W., & Pold, S. (2002). Towards the aesthetics of Human-computer interaction (pp. 11–12). Presented at the Proceedings of the Second Danish Human-Computer Interaction Research Symposium. Retrieved from http://www.diku.dk/forskning/Publikationer/tekniske_rapporter/2002/02-19.pdf/

Betts, T. (2002, March 10). WEB WIZARDS AT THE DESIGN MUSEUM. *Mute*. Retrieved April 25, 2011, from http://www.metamute.org/editorial/articles/web-wizards-design-museum

Billington, M. (2011). Richard III - review, The Guardian, http://www.guardian.co.uk/stage/2011/jun/29/richard-iii-review-kevin-spacey

Black, R. (2011, December 10). “Read the story before you lay out the page” . . . And other things I learned from Lou Silverstein. *rogerblack.com*. Retrieved April 3, 2012, from http://rogerblack.com/blog/post/learning_from_lou

Blanc, S. (2011). Tweetbot’s Got Personality, http://shawnblanc.net/2011/04/tweetbot-review/

Bleecker, J. (2009, March 19). Design Fiction: A Short Essay on Design, Science, Fact and Fiction. *nearfuturelaboratory.com*. Retrieved November 26, 2012, from http://nearfuturelaboratory.com/2009/03/17/design-fiction-a-short-essay-on-design-science-fact-and-fiction/

Blythe, M., Bardzell, J., Bardzell, S., & Blackwell, A. (2008). Critical issues in interaction design (Vol. 2, pp. 183–184). Presented at the BCS-HCI '08: Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction, British Computer Society.

Blythe, M., Hassenzahl, M., & Law, E. (2009). Now with Added Experience? *New Review of Hypermedia and Multimedia*, *15*(2), 119–128. doi: 10.1080/13614560903251100

Bolter, J., & Gromala, D. (2003). Introduction. In *Windows and mirrors: interaction design, digital art, and the myth of transparency* (pp. 2–7). MIT Press.

Bødker, M. (2004). Position Paper for the Aesthetic HCI Workshop. In O. W. Bertelsen, M. G. Petersen, & S. Pold (Eds.), *Aesthetic Approaches to HCI - Proceedings* (pp. 5–8).

Bødker, S. (2006). When second wave HCI meets third wave challenges. Presented at the NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles, ACM Request Permissions. doi:10.1145/1182475.1182476

Bødker, S. & Kammersgaard, J. (1984) Interaktionsbegreber, internt arbejdsnotat, version 2.

Dallow, P. (2003). Representing creativeness: practice-based approaches to research in creative arts. *Art, Design & Communication in Higher Education*, *2*(1/2), 49–66.

Darrough, S. (2010, February 19). MoMA | A Few Words with Legendary Web Designer Yugo Nakamura. *moma.org*. Retrieved November 30, 2012, from http://www.moma.org/explore/inside_out/2010/02/19/a-few-words-with-yugo-nakamura

DesignMuseum. (2005). A History of Penguin Books. *designmuseum.org*. Retrieved April 4, 2012, from http://designmuseum.org/design/penguin-books?print=1

Dewey, J., (1980/1934) Art as Experience, Perigee Books, New York.

DiSalvo, C. (2012) Adversarial Design. MIT Press.

Doubleday, Richard B. (2006) Jan Tschichold, designer: the Penguin years. Lund Humphries Publishers.

Dunne, Anthony. (2005a). *Hertzian Tales: Electronic Products, Aesthetic Experience, and Critical Design*. MIT Press.

Dunne, Antony. (2005b). The Electronic as Post-Optimal Object. In *Hertzian Tales: Electronic Products, Aesthetic Experience, and Critical Design*. MIT Press.

Dunne, Anthony., & Raby, F. (2010). Between Reality and the Impossible (Entre la Réalité et l'impossible), in Téléportation, catalogue of the 2010 SaintÉtienne International Design Biennal (p.105).

Dunne, Anthony, & Raby, F. (n.d.). Critical Design FAQ. http://www.dunneandraby.co.uk/content/bydandr/13/0. Retrieved January 20, 2012, from http://www.dunneandraby.co.uk/content/bydandr/13/0

Engeström, Y., & Escalante, V. (1995). Mundane Tool or Object of Affection The Rise and Fall of the Postal Buddy. In *Context and consciousness* (pp. 325–373).

Eskilson, S. (2007). Graphic Design - A New History. Yale University Press

Fallman, D. (2007). Persuade Into What? Why Human-Computer Interaction Needs a Philosophy of Technology. In Y. Kort, W. IJsselsteijn, C. Midden, B. Eggen, & B. J. Fogg (Eds.), Persuasive Technology

Fallman, D. (2011). The New Good: Exploring the Potential of Philosophy of Technology to Contribute to Human-Computer Interaction (p. 1051). Presented at the 2011 annual conference, New York, New York, USA: ACM Press. doi:10.1145/1978942.1979099

Fishwick, P. (2004). An introduction to aesthetic computing. Aesthetic computing.

Fitzpatrick, M. (2008, February 4). Yugo Nakamura - The Craftsman. *http://www.creativereview.co.uk/cr-blog/2008/february/yugo- nakamura-the-craftsman*. Retrieved January 20, 2012, from http://www.creativereview.co.uk/cr-blog/2008/february/yugo- nakamura-the-craftsman

Ford, R. (2006, September 6). 10th Anniversary of Flash. *thefwa*. Retrieved September 10, 2011, from http://www.thefwa.com/article/10th-anniversary-of-flash

Frayling, C. (1997). Practice-Based Doctorates in The Creative and Performing Arts and Design, Coventry: UK Council for Graduate Education.

Fukushi, N. (2004, August). SHIFT | PEOPLE | YUGO NAKAMURA. *shift.jp.org*. Retrieved December 1, 2012, from http://www.shift.jp.org/en/archives/2004/08/yugo_nakamura.html

Geertz, C. (1977). The Interpretation of Cultures. Basic Books.

Greenberg, S., & Buxton, B. (2008). Usability evaluation considered harmful (some of the time) (pp. 111–120). Presented at the Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems. doi: 10.1145/1357054.1357074

Hackos, J., & Redish, J. (1998) User and task analysis for interface design. John Wiley and Sons Inc., New York, USA

Hallnäs, L., & Redström, J. (2001). Slow Technology – Designing for Reflection. *Personal and Ubiquitous Computing*, *5*(3), 201–212. doi:10.1007/PL00000019

Hallnäs, L., & Redström, J. (2002). From use to presence: on the expressions and aesthetics of everyday computational things. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *9*(2), 106–124.

Harrison, S., Sengers, P., & Tatar, D. (2007). The Three Paradigms of HCI. Presented at the ALTCHI 2007.

Heidegger, M. (2008). Being and Time. Harper Perennial Modern Classics.

Hollis, R. (1994). Graphic Design: A Concise History, Thames & Hudson.

Holston, D. (2011). The Strategic Designer: Tools & Techniques for Managing the Design Process, HOW Books.

Holt, M. (1988). Stage Design and Stage Properties, Phaidon Press Limited, London. Pg.37

Hutchins, E. (1995). Cognition in the Wild. MIT Press.

Ishizaki, S. (2003). Chapter 1. In *Improvisational Design - Continuous, Responsive Digital Communication*. The MIT Press.

Johnson, S, A. (1999). *Interface Culture*. Basic Books.

Kauffman, S, A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution.*

Lawless, J. (2011) *Kevin Spacey Shines as Shakespeare's Richard III,* Associated Press, http://abcnews.go.com/Entertainment/wireStory?id=13968815

Lofte, L. (2011). *Tweetbot once again redefines Twitter for iPhone.* TiPb, http://www.tipb.com/2011/04/13/tweetbot-iphone-app-review/

Lupton, E. (2010). *Thinking With Type.* Princeton Architectural Press.

Maeda, J. (2004). *Creative Code.* Thames & Hudson.

Manovich, L. (2001). *The language of new media*. The MIT Press.

Manovich, L. (2005). Generation flash. *Total Interaction*, 67–77.

Manovich, L. (2006). Interaction as an aesthetic event. *Vodafone Receiver, Vodafone*, *17*(1).

Manovich, L. (2008). *Software Studies: SOFTWARE TAKES COMMAND (book draft)*. lab.softwarestudies.com.

Manovich, L. (2009). Import/Export: Workflow and Aesthetics of Graphic Design. In H. Armstrong (Ed.), *Graphic Design Theory* (pp. 127–132). Princeton Architectural Press. Retrieved from http://manovich.net/DOCS/workflow.doc

Matthews, B., Stienstra, M., & Djajadiningrat, T. (2008). Emergent interaction: creating spaces for play. *Design Issues*, *24*(3), 58–71.

McCarthy, J, J., & Wright, P. (2007). *Technology as Experience.* The MIT Press

McKeich, M., & Dziekan, V. (2005). Software Democracy and the Quiet Revolution of Practice. In L. Jones, P. Anastasiou, R. Smithies, & K. Trist (Eds.), (pp. 1–7). Presented at the Vital Signs: Creative Practice & New Media Now. Retrieved from http://search.informit.com.au/documentSummary;dn=040898779124914;res=IELHSS

McNichol, T. (2004, December). Roads Gone Wild. *WIRED Magazine*, *12*(12), 1–3.

Millward, D. (2006, November 4). Is this the end of the road for traffic lights? *The Telegraph*, pp. 1–3. Retrieved from http://www.telegraph.co.uk/news/uknews/1533248/Is-this-the-end-of-the-road-for-traffic-lights.html

Nakamura, Y. (2010, August 17). FITC SF 2010 - Long Time No See - Part I on Vimeo. *vimeo*. Retrieved June 26, 2012, from http://vimeo.com/14397973

Niedderer, K. (2007). Designing Mindful Interaction: The Category of Performative Object. *Design Issues*, *23*(1), 3–17. Retrieved from http://www.mitpressjournals.org/doi/abs/10.1162/desi.2007.23.1.3

Nielsen, J. (1999). When Bad Design Elements Become the Standard. *Alertbox*, 1–5.

Nielsen, J. (2000, October 29). Flash: 99% Bad (Jakob Nielsen's Alertbox). *useit.com*. Retrieved December 11, 2012, from http://www.useit.com/alertbox/20001029.html

Norman, D. (2003). Attractive Things Work Better. In *Emotional Design: Why We Love (or Hate) Everyday Things* (pp. 1–12). Basic Books.

Norman, D. A. (2010). The way I see it: Technology first, needs last. *interactions*, *17*(2), 38. doi:10.1145/1699775.1699784

Ortony, A. (1979). *Metaphor and Thought.* Cambridge University Press

Petersen, M. G., Hallnäs, L., & Jacob, R. J. K. (2008). Introduction to special issue on the aesthetics of interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *15*(3), 1–5. doi:10.1145/1453152.1453153

Petersen, M. G., Iversen, O. S., Krogh, P. G., & Ludvigsen, M. (2004). Aesthetic Interaction — A Pragmatist's Aesthetics of Interactive Systems. Presented at the DIS2004.

Prasad, N. (2011). *Tweetbot Review,* Techtree, http://www.techtree.com/India/Reviews/Tweetbot_Review/551-115093-609.html

Reas, C., & McWilliams, C. (2010). *Form+Code in Design, Art, and Architecture*. Princeton Architectural Press.

Reas, C., & McWilliams, C. (2012, January 23). Designing Programs. (C. Reas & C. McWilliams, Eds.) *Creative Applications*. Retrieved June 8, 2012, from http://www.creativeapplications.net/theory/designing-programs-theory/

Reas, C., & McWilliams, C. (Eds.). (n.d.). *Form+Code in Design, Art, and Architecture*. Princeton Architectural Press. Retrieved January 19, 2012, from http://formandcode.com

Redström, J. (2001). *Designing everyday computational things*. *Thesis*. Göteborg University.

Redström, J. (2006). Towards user design? On the shift from object to user as the subject of design. *Design Studies*, *27*(2), 123–139.

Schön, D. (1983). *The Reflective Practitioner.* Basic Books.

Sengers, P. (2004). The Engineering of Experience. In M. A. Blythe, K. Overbeeke, A. F. Monk, & P. C. Wright (Eds.), Funology: From Usability to Enjoyment (Human-Computer Interaction Series). Springer.

Shorey, P. (1938). *Platonism: Ancient and Modern,* Cambridge University Press

Shusterman, R. (1992) *Pragmatist Aesthetics.* Living Beauty, Rethinking Art. Blackwell.

Simpson, A. (2008, February 12). Grandmasters Of Flash: An Interview With The Creators Of Flash. *http://coldhardflash.com/2008/02/grandmasters-of-flash-an-interview-with-the-creators-of- flash.html*. Retrieved January 22, 2012, from http://coldhardflash.com/2008/02/grandmasters-of-flash-an-interview-with-the-creators-of- flash.html

Smith, B. (2012, June 5). The Beginner's Guide to Twitter. *mashable.com*. Retrieved July 19, 2012, from http://mashable.com/2012/06/05/twitter-for-beginners/

Society of news design. (2006). *The Best of Newspaper Design: 27th Edition,* Rockport Publishers.

Spencer, C. (2011). Richard III, Old Vic, review, *The Telegraph,* http://www.telegraph.co.uk/culture/theatre/theatre-reviews/8606884/Richard-III-Old-Vic-review.html

Sterling, B. (2012, April 2). An Essay on the New Aesthetic | Beyond The Beyond | Wired.com. *wired.com*. Retrieved December 3, 2012, from http://www.wired.com/beyond_the_beyond/2012/04/an-essay-on-the-new-aesthetic/

Strand, D. (1998), *Research in the Creative Arts, Canberra: Department of Employment, Education,* Training and Youth Affairs.

Suchman, L. (1987). *Plans and Situated Actions: The Problem of Human/Machine Communication.* Cambridge University Press.

Tedesco, D., & Tullis, T. (2006). A comparison of methods for eliciting post-task subjective ratings in usability testing. *Usability Professionals Association (UPA)*. (pg.1–9).

Tractinsky, N. (2012). Visual Aesthetics: in human-computer interaction and interaction design. (M. Soegaard & R. F. Dam, Eds.)*Encyclopedia of Human-Computer Interaction*. Retrieved April 27, 2012, from http://www.interaction-design.org/encyclopedia/visual_aesthetics.html

Tractinsky, N., Katz, A. S., & Ikar, D. (2000). What is beautiful is usable. *Interacting with Computers*, *13*, 127–145.

Udsen, L. E., & Jørgensen, A. H. (2005). The aesthetic turn: unravelling recent aesthetic approaches to human-computer interaction. *Digital Creativity*, *16*(4), 205–216. doi:10.1080/14626260500476564

Verganti, R. (2009). *Design Driven Innovation.* Harvard Business Press.

Verganti, R. (2010, March 19). User-Centered Innovation Is Not Sustainable. *HBR Blog Network*. Retrieved December 7, 2011, from http://blogs.hbr.org/cs/2010/03/user-centered_innovation_is_no.html

Wajcman, J. (1991). *Feminism Confronts Technology,* Allen & Unwin, Sydney.

Walton, J. (2001, December 3). A web of visions. *BBC News - Dot Life*. Retrieved May 22, 2011, from http://news.bbc.co.uk/2/hi/in_depth/sci_tech/2000/dot_life/1684640.stm

Whitelaw, M. (2012). Transmateriality: Presence Aesthetics and the Media Arts. In U. Ekman & M. Fuller (Eds.), *Throughout: Art and Culture Emerging With Ubiquitous Computing*. MIT Press. Retrieved from http://tinyurl.com/transmateriality

Whitelaw, M. (2011a, March 12). Dynamic Design - Three Systems. *teemingvoid.blogspot.com.au*. Retrieved March 16, 2011, from http://teemingvoid.blogspot.com.au/2011/03/dynamic-design-three-systems.html

Whitelaw, M. (2011b, April 10). After the Screen: Array Aesthetics and Transmateriality. *teemingvoid.blogspot.com.au*. Retrieved December 3, 2012, from http://teemingvoid.blogspot.com.au/2011/04/after-screen-array-aesthetics-and.html

Wright, P., McCarthy, J., & Meekison, L. (2004). Making Sense of Experience. In M. A. Blythe, K. Overbeeke, A. F. Monk, & P. C. Wright (Eds.), *Funology: From Usability to Enjoyment (Human-Computer Interaction Series)* (pp. 43–53). Springer.

Wright, P., Wallace, J., & McCarthy, J. (2008). Aesthetics and experience-centered design. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *15*(4), 18–21. doi:10.1145/1460355.1460360

Yallop, C. (ed.) (2004). *Macquarie Concise Dictionary,* The Macquarie Library, Macquarie University, Australia.

Yugo Nakamara Interview. (2001, May). Yugo Nakamara Interview. *http://www.designboom.com/portrait/flash.html*. Retrieved January 20, 2012, from http://www.designboom.com/portrait/flash.html

Yugo Nakamura - Creativity 50 - Creativity Online. (2008, February 19). Yugo Nakamura - Creativity 50 - Creativity Online. *creativity-online.com*. Retrieved November 30, 2012, from http://creativity-online.com/news/yugo-nakamura/125202

Yugo Nakamura - interview. (n.d.). YUGO NAKAMURA - interview. *http://designmuseum.org/design/yugo-nakamura*. Retrieved January 21, 2012, from http://designmuseum.org/design/yugo-nakamura