Email: A History of Syntax

Isaac Quinn DuPont
Faculty of Information
University of Toronto
quinn.dupont@utoronto.ca

Abstract

Email is important. Email has been and remains a "killer app" for personal and corporate correspondence. To date, no academic or exhaustive history of email exists, and likewise, very few authors have attempted to understand critical issues of email. This paper *explores the history of email syntax*: from its origins in time-sharing computers through Request for Comments (RFCs) standardization. In this historical capacity, this paper addresses several prevalent historical mistakes, but does not attempt an exhaustive historiography. Further, as part of the rejection of "mainstream" historiographical methodologies this paper *explores a critical theory of email syntax.* It is argued that the ontology of email syntax is material, but contingent and obligatory—and in a techno–social assemblage. Email was instrumental in shifting computers from computation machines to text machines. Cryptography reappears throughout the theoretical and historical picture, as do love emails and postcards.

Introduction

This paper is an exploration of email technology,[1] which has received almost no academic interest. Some social scientists and management scientists have researched email, but these studies tend to take the technology as an unproblematic given. A central challenge of studying email technology is that it is many things and has grown and shifted through the forty years of its history. Proto–email performed some of the same functions email does today. The origin of email as the unintended application of file transfer protocols for communication (on time–sharing computers and across early networks) led to email technology that has a highly standardized syntax, for both communication protocols and header (or "routing") information.

The term "syntax" is especially problematic. Syntax has a relationship to order, but it is not clear how all of the uses in different fields articulate this orderliness. As I will use it, "email syntax" refers to the arrangement of word tokens in an appropriate (orderly) manner for processing by computers. Perhaps "computers" refers to syntactical processing, making my definition circular. So be it, I will hide behind the engineer's keystone of pragmatism. Email systems work (usually), because syntax is arranged such that messages can be passed.

This paper demonstrates the centrality of syntax to the history of email, and investigates interesting socio–technical issues that arise from the particular development of email syntax. Syntax is an important constraint for contemporary computers, perhaps even a definitional quality. Additionally, as machines, computers are physically constructed. Thus, email syntax is material. This is a radical view for the academy, but (I believe), unproblematic for the engineer. In fact, the methodology of this paper is radically empirical: it is historiography, and scarcely more.

Email is widely considered the original "killer app" and is of equal (or greater) importance to the most lauded computer technologies, such as networking, graphical user interfaces, or web browsers. Yet, despite the obvious importance, outside of software engineering email is poorly understood. Beyond anecdotes and a poorly–researched Masters thesis there is no email historiography. Email technology is ripe for critical theoretical research, like that being done on virtual reality, social networks, Web 2.0 and other topics.

In the past, ubiquitous (*i.e.,* important) technologies have succumbed to hagiography or falsehoods as neither "side" ("technical" or "critical") has managed to properly bridge the gap. On the one hand, part of the challenge of doing research on email technology has been to wrest control of technical domains from the technicians (engineers, designers, managers). Critical (social or philosophical) studies have often come from well outside of the technical field, and suffer from a lack of detail and technical clarity. Science Studies has arisen in recent decades as a considerable redress in this regard, and has sought to understand and problematize the issues. I see Science Studies as a methodological ally.

This paper is exploratory. There are limitations to the historiographical method employed, and the critical philosophy is speculative. Speculative, however, in the sense closer to that intended by the Speculative Realists. For example, F.W.J. Schelling sought to upend Fichte's transcendental philosophy

[1] There is no good term for this type of technology. I do not think that this analysis must be restricted to electrical computing and networks (fiber optics do not seem to obviate any of the claims), but simply "mail" is too broad (since I *do* want to distinguish between mail and email).

by seeing nature as producing the ideal. The appeal is that Schelling's position enjoys a kind of humility towards nature and the limits of human transcendental grounding. Technology is not just physics, but working out how we can sensibly talk about a theory of technology is very much the challenge here. As an empirical subject, there are limitations with a predominately documentary methodology. The proto–email history draws from a fairly wide documentary corpus (manuals, technical notes, and dubious secondary historical sources), while the later history draws almost exclusively from Request for Comments (RFCs).

My methodological commitments are: the acceptance of the explanatory power of exterior relations (and the rejection of interior relations), the recognition that technical decisions are (often) contingently obligatory, and a methodological reductionism to material reality. According to Michel Foucault, Gilles Deleuze and others, abstract concepts do not explain, but instead require explanation (*i.e.*, these thinkers espouse a form of nominalism). This is in contradistinction to Hegelian historiography, where any relations that happen to obtain between objects are extraneous and do not concern their nature. Hegel describes interior relations as such:

> This is what constitutes the character of mechanism, namely, that whatever relation obtains between the things combined, this relation is *extraneous* to them that does not concern their nature at all, and even if it is accompanied by a semblance of unity it remains nothing more than *composition, mixture, aggregation,* and the like (DeLanda, 2006, p. 9).

Instead, Deleuze's famous example of the wasp and orchid suggests that the assemblage of the wasp and orchid are obligatory but empirical. The wasp is accidently related to the orchid, but in a narrow Aristotelian sense. Parts may be exchanged between and among assemblages and change interactions, but the properties of the parts cannot explain the whole assemblage because the assemblage is not the sum of the interior relations, rather, it is the exercise of the parts. While some relations are occasional and circumstantial, some come to be obligatory through forces of coevolution or codevelopment.

These parts are not logically necessary (since interior relations have been rejected), instead, they are seen as *contingently obligatory* (DeLanda, 2006, p. 11). Contingently obligatory assemblages are empirical and historical (unlike logically necessary ones). According to Manuel DeLanda these relations of exteriority vacillate between a purely material role and a purely expressive role, where each part plays some role in the assemblage, aiding territorialisation or deterritorialisation (DeLanda, 2006, p. 12). This methodology is synthetic, but requires a *coding* process in which parts are (typically) held in hierarchies. The parts of the assemblage transform through steps adjusting to local conditions. Graham Harman argues that these parts conspire at each step to determine "where the possible variations can be addressed or ignored" (Harman, 2009, p. 15). Each step mediates non–neutral layers, what Bruno Latour calls a mediator. Harman describes the process of the mediator as such:

> A mediator is not some sycophantic eunuch fanning its masters with palm-leaves, but always does new work of its own to shape the translation of forces from one point of reality to the next. (Harman, 2009, p. 15)

The mediator contests. And, email *syntax* is, I will argue, one such mediator that has a will to power. Commitments to material reductionism are now brought to logical force, since nominalism, exterior relations, and coded layers of mediation do not require significant interaction of immaterial realities to function. Whether immaterial reality exists is *not* determined by these methodological commitments (I am agnostic on the existence of immateriality).

Hannah Arendt describes these technologies as a "loud voice for escape from earth." Not silent, these technologies scream as human and technology become one, cyborg–like. Mark Coeckelbergh argues that there is no longer an assembly of things distinct from an assembly of humans (Coeckelbergh, 2009, p. 3). Despite the scream, as technology goes through foldings with each step, the parts become more ubiquitous and banal.

Deriving a politics of artefacts is difficult. Revealing the politics of Patriot missiles or automobile seatbelts is the first step, but speaking politically about cupboards, stopwatches, alternating currents, or email is a much deeper challenge. There appears to be a gradation of politics with respect to artefacts, as Coeckelbergh admits,

> Consider companion robots, pet robots, household robots, care robots, sex robots, military robots, etc. Although such robots are only just emerging, they provide an interesting case, since they are more explicitly 'political' than many other artifacts. (Coeckelbergh, 2009, p. 3)

With no (necessary) distinction between humans and technology, what is it that permits gradations of the political? For the study of humans, it's a perennial question. More troubling still, how do ethics intersect with politics? A posthumanist answer, or at least an artefactual answer cannot involve claims to agency or intentionality. Coeckelbergh gives artefacts "speech", which he argues is sufficient for political engagement

but kills off the subject, leaving humans to wallow in *angst* (Coeckelbergh, 2009, p. 4).

But, discourse constructs categories and concepts. Social Constructivism is problematic because the constructed thing is not material or a subject, but rather, a concept. Ian Hacking argues that, for example, the social construction of women refugees is not women, but instead it is the category of women refugee (Hacking, 1999, p. 10). These categories, however, require explaining since they are not free of politics. Logically prior to these categories there exists a plane of immanence that, according to Deleuze, separates virtual and actual.

The plane of immanence is contested, for "before being there is politics" (Deleuze and Guattari quoted in Patton, *Deleuze and the Political*, 9). An assemblage may be the exercise of the parts, but politics comes before this exercise. We cannot study intentional practices to understand politics. Latour locates ethics or politics in the mediator, and calls all artefacts actants. Deleuze locates ethics or politics in lines of flight, and the interplay of territorialisation and deterritorialisation. Unlike a "scientific" analysis of the politics of email syntax, here, the reference and representation of email syntax is not important. Rather, on a plane of immanence social forces and natural or "machinic" forces stabilize identity, with each component of the assemblage working to do or undo actual identities. Both the molar and the molecular are written into the history, since it is necessary to understand both the military industrial complex that birthed email as well as the arbitrary decision to use the "commercial at" (@) symbol for dividing the username and hostname. The historical components interact as the assemblage called "email" permits and defines.

Writing Machines & Killer Apps

The proto–history of email testifies to the materiality of its syntax. From typewriters to computers to DARPA–funded networks, email formed a kind of 'fast text'. The problem is big, in fact, since "cyberculture cannot be understood without reference to the history of writing" (Milne, 2000, p. 100).

Email communication forms many assemblages, each territorialising or deterritorialising. Corporate email necessarily contains a legal appendage: a foot of legalese declaring privacy and confidentiality and non–culpability of the corporation is always included at bottom of the sent email. These corporate emails territorialise the chain letter or the link to a funny cat video getting passed on corporate time (and dime). Personal email also territorialises and deterritorialises, as different assemblages of technology get plugged in. Replace corporate legalese with a Google AdSense advertisement reading your love letters and you *feel* the territorialisation. It shocks you in to recognition of your capitalist consumption. Personal email deterritorialises through history, as email syntax changes to allow the sender to create new assemblages: first across time (time sharing computers), then eventually across global space and time. As standardization occurs, and new syntax is created to form new parts of email the process of territorialisation begins again. The material conditions of email are important, since, across time and space the collective assemblage of email technology *is* women, and war, and more. Esther Milne's argues that one ought to

> take seriously the work of theorists such as Friedrich Kittler, Katherine Hayles, and Donna Haraway, who focus attention on the material conditions of textual production and consumption by putting into question the idea of transparent unmediated communication. (Milne, 2000, p. 106)

Kittler argues that the technologizing of the body produced, initially, women as "the white sheet of nature or virginity onto which a very male stylus could then inscribe the glory of its authorship", then, women as *literally* "Type–Writer" (Kittler, 1999, pp. 186-187). This transubstantiation of woman to typewriter replaces sense perception and memory with inscription. Milne argues that Plato's *Phaedrus* was the first example of this techno–fear, soon followed by all literate societies (Milne, 2000, p. 101). With the typewriter's origins in war production by, among others, Remington and Son the "typewriter became a discursive machine–gun," ever quickly producing text (Kittler, 1999, p. 191). As women/typewriters were trained and made dexterous the speed of text increased.

In war, the speed of killing has increased: Roman's "decimated" (it's enough to kill $1/10^{th}$ the population), medieval city–dwellers outlived multi–year siege tactics, modern infantry sat in trenches dodging machine–gun fire, and today, smart bombs and drones nearly instantly vaporize the target. In text, speed has also increased. Speed is not a unique quality to email, yet it may occur that the speed of text and the speed of war continue in lock–step, as a couple tied to the same set of desires. Deleuze and Guattari state that "every machine, in the first place, is related to a continual material flow (*hyle*) that it cuts into" (Deleuze & Guattari, 2009, p. 36). For the typewriter, women are *hyle*. For war, cryptography is *hyle*. Underneath the *hyle*, as Aristotle realized in his own way, the changing 'substance' is the virtual, pushed along by desiring machines. Women and war are the desiring machines underneath email.

The Second World War moved text rapidly. Remington and Son's ersatz machine–guns were far

too slow for most communication, but the rapid medium of the day was broadcasted radio or too–easily tapped electrical communication cable (teleprinter). Like most wars, encryption was used.

Although the first encrypted messages were likely priestly, the first identifiably cryptologic system (kryptós) was, according to Thucydides, the *skytale* used by the Spartans for war–time messaging (Wrixon, 2005, p. 21). By World War Two encryption was using symmetric/secret–key algorithms that encoded messages on electrically wired rotor machines. The first of these rotor machines were invented in circa 1919, with the famous ENIGMA machine being commercially produced (but unsuccessful at first) in the early 1920s (Wrixon, 2005, p. 260). The American military begun encryption with the Electronic Cipher Machine (1925) that through many iterations was never subject to successful cryptanalysis. Conventional wisdom is that at Bletchley Park Alan Turing,[2] following Babbage's cryptanalysis techniques against polyalphabetic cyphers from the Crimean War, developed the "bombe" technique of cryptanalysis against the ENIGMA machine (Kittler, 1999, p. 255). The conventional story continues, that Turing was instrumental in the development of the modern computer, providing even faster text processing than the bombe drums developed in Bletchley Park.

Email is a war machine for many historical reasons. Remington and Son produced weapons and typewriters, and typewriters became computers (through cryptologic tools in war). Through cryptography war bodies (states and institutions) gain secret power. Deleuze and Guattari argue that

> it is the secret power (*puissance*), or strength of solidarity, and the corresponding genealogical mobility that determine its eminence in a war body. (Deleuze & Guattari, 1987, p. 366)

The tools of cryptography erase meaning (plaintext to cyphertext), but with private keys or cryptanalysis (exempting public–key cryptography for the moment) meaning can be re–inscribed. Political strength comes with the ability to create an *episteme* (in Foucault's sense) from cyphertext. The war machine is not external to the apparatus (Deleuze & Guattari, 1987, p. 354).

Text processing and communication are obligatory parts of war. The proto–history of email suggests that text processing was an odd twist to early 'computing', and propelled by the networks developed within the war efforts of DARPA. Deleuze and Guattari use a theory of games to understand war and the directionality of the game pieces neatly reflect the strategy of email. The coded pieces in chess and Go display relations of interiority and exteriority, respectively (Deleuze & Guattari, 1987, p. 353). Go functions as "pure strategy" in an open space, "without aim or destination". Go is a smooth space of *nomos*, while chess is striated like *polis* (Deleuze & Guattari, 1987, p. 353). In the end, chess codes and decodes, while Go territorialises and deterritorialises. Email has parts that function "without aim or destination" (such as Bayesian spam filtering), but most parts function like the coded pieces of chess, constantly territorialising.

To understand how the parts of email technology territorialise we must look at the development of the parts, starting with the proto–history of email as a form of communication on time–sharing computers. By the 1960s contemporary computers were available at military and university institutions (as well as private research organizations such as Bolt, Beranek and Newman). In 1965 Thomas Merrill and Lawrence Roberts at DARPA used Leonard Kleinrock's earlier packet–switching research to network computers using packets instead of switches (Leiner et al., 1997, p. 103). By 1967 the computers were being connected together under a DARPA initiative to create the ARPANET,[3] with BBN to supply the Interface Message Switchers (IMPs) (Leiner et al., 1997, p. 103).

The initial DARPA requirement for the ARPANET was to provide networking capabilities for resource sharing (Flichy, 2000, p. 3). A pioneering spirit for the ARPANET was Joseph Licklider who,

[2] It is quite well known that the "*bomba kryptologiczna*" technique for breaking ENIGMA encryption was developed in 1932 by Marian Rejewski, a Polish mathematician and cryptanalyst. The Polish Cipher Bureau (with the assistance of a French spy) kept their cryptanalysis current as the Germans changed rotors and introduced further complexity, until in 1939 when the Germans introduced two new rotors. The cryptanalysis problem was still qualitatively the same, but increased in difficulty substantially (jumping from 6 to 60 cryptanalysis drums). At this point the Polish shared their cryptanalysis techniques with the French and British allies. Turing and Welchman improved the techniques inherited by the Polish to break the new and more difficult ENIGMA machines. More complete histories exist, but this conventional wisdom with Turing as the candle in the wind pervades nonetheless.

[3] The Defense Advanced Research Projects Agency (DARPA) has been renamed several times; it started as ARPA but then in 1972 was renamed DARPA, then again renamed ARPA (1993), and DARPA (1996).

in 1962[4] argued that computers could be used for more than resource sharing. Licklider wanted "to improve man–machine interaction in teaching and learning, in planning and design, and in visualizing the internal processes of computers", in short, Licklider was a posthumanist in search of mind/brain augmentation through computing communication. Licklider later argued, "I wanted interactive computing, I wanted time-sharing. I wanted themes like: computers are as much for communication as they are for calculation" (Licklider quoted in Flichy, "Internet or the ideal scientific community," 3). Time–sharing for communication, not resource sharing, became the new computing prerogative.

While Licklider was laying the groundwork at DARPA for what would eventually become the ARPANET (which email would function across), Douglas Engelbart was developing the On Line System (NLS) Teleconferencing System at the Augmentation Research Center (ARC) in Stanford Research Institute (SRI). NLS was a very early implementation of networked computers that, in 1971, joined the ARPANET. Before the existence of the ARPANET, NLS was a system of networked communication that, unlike later ARPANET implementations used closed–circuit television to display terminals remotely (Englebart & English, n d, p. 5c3a). The Journal subsystem made NLS a unique and important precursor to email. The Journal subsystem was conceived in 1966 for the purposes of keeping a "log" of events, and performing a document–oriented communication system, described as "direct distribution". Direct distribution could send documents (memos, messages, data records, *etc*.) directly to invited participants through the use of a personal IDENT code (Engelbart's code was his initials, DCE) (Engelbart, 1975, p. 7c). IDENT codes were stored in a directory for lookup and were organized by group memberships (with multiple memberships possible). Documents were sent to a "mail box" and marked with a status, such as "For Action" or "For Information" (Engelbart, 1975, p. 7d). Depending on the length of the document, either a "citation" was displayed to the recipient (for later retrieval of the full document), or the entire document was displayed. The Journal began in 1966, a full five years before the accepted "official" start–date of email,[5] yet the Journal remained, co–developing alongside other systems of email. Many parts of the system were shared with proto–email systems, such as the IDENT codes, directory lookup and mailing lists (mirroring similar functionality developed later in email).

In 1961 Programmed Logic for Automated Teaching Operations (PLATO) II and Compatible Time-Sharing System (CTSS) introduced time–shared computing. Time–sharing quickly became popular and through the 1960s it was common to pass notes to other users by leaving a file for another user by placing it in a common directory. Tom Van Vleck suggests that it was common to title the file left in the common directory with a person's name, such as to tom (Vleck, n d). The first system to formalize a mail command occurred on CTSS running on an IBM 7094 at Massachusetts Institute of Technology (MIT). Between December 14, 1964 and January 8, 1965 the undated Programming Staff Note 39 for CTSS was written and distributed by Crisman, Schroeder, and Pouzin (Saltzer, 2010). In February 1965 Van Vleck joined the programming staff at MIT, along with Noel Morris shortly thereafter, and read Programming Staff Note 39. Programming Staff Notes did not describe implemented functionality for CTSS, instead they contained directives or ideas for future implementation.

Over the spring of 1965 Van Vleck and Morris read Programming Staff Note 39 and over the weekend of July 4th, 1965 they implemented the MAIL subsystem for CTSS using privileged commands on the problem number M1416 (Vleck, 2010). In December 1969, in the CTSS Programmer's Guide MAIL functionality is officially described, mirroring the syntax suggested in Programming Staff Note 39.[6]

[4] In 1962 Licklider joined two ARPA departments, *Behavioral Sciences* and *Command and Control Research Department*. In 1964 Licklider left ARPA, after *Command and Control Research Department* was renamed *Information Processing Techniques Office* (IPTO), reflecting Licklider's influence on time-sharing computers and communication processing. Flichy incorrectly argues that Licklider went from the publication of "Man-Computer Symbiosis" in 1962 (at ARPA, but previously at BBN since 1957, although Flichy does not mention this) to IPTO in 1964.

[5] This date, as we will see, is incorrect (or at least requires some subtlety to understand). Most people place the start of email with Ray Tomlinson's enhancements to SNDMSG in late 1971.

[6] Programming Staff Note 39 MAIL syntax is: MAIL LETTER FILE USER1 USER2 USER3 … . CTSS Programmer's Guide MAIL syntax is: MAIL NAME1 NAME2 PROB1 PROG1 … –PROBn– –PROGn–. NAME is the name of the file to be mailed, and PROB and PROG are, according to the CTSS Programmer's Guide, the "users to which mail will be sent", while the 1969 CTSS source code describes PROB and PROG as "DIRECTORIES TO WHICH IT IS TO BE SENT". The CTSS Programmer's Guide also includes the LIST option as well as * for recipients, meaning "all".

On COM5 of the 1969 CTSS source code listing the MAIL subsystem is in place, last modified by R. Roach on March 17, 1969. At this point no to: syntax had been developed, but the combination of PROB and PROG (the recipient's problem number and program number) mark a destination. The delivered mail includes FROM syntax in the form of FROM USRPB USRPG DATE TIME on the first line, thus identifying the sender's problem number, programmer number, and the date and time of transmission. In circa 1969 Vleck re–implemented MAIL for the Multiplexed Information and Computing Service (Multics) time–sharing operating system (Vleck, n d). The Multics MAIL syntax was slightly different (*e.g.*, mail VanVleck.Multics), and growing closer to the familiar username@host identification system developed by Ray Tomlinson in late 1971. These systems were not networked, so while they shared some of the features of later systems, they cannot be called email in the sense used today.

In many ways, networked email systems originated simultaneous to the formation of ARPANET and the RFC documentation structure formed to document ARPANET. The critical function and popularity of email ensured simultaneous development with ARPANET. RFCs are an interesting (and seldom studied) historical source: they are immutable, published in completed form with co–citations, and obsolete or update each other. RFCs also have varying statuses, and tend to describe completed (working) systems, rather than document "standards" to be developed. Finally, RFCs were developed somewhat organically and fell into a discernable style with rules only as they developed. The first 30 years of RFCs were "edited" by Jon Postel, where "edited" means shepherd, alter, develop, limit, and generally (benevolently) rule over. Since Postel's death (in 1998) the RFCs have been managed by a more democratic and formal body (under the auspices of the Internet Engineering Task Force).[7] Most of this paper is, in a sense, medieval. I start the "modern era" of email with the rupture at RFC 821 and RFC 822, that splits email systems in to two logically separate (but technologically inseparable) systems.

On April 16, 1971 RFC 114 was published to specify the File Transfer Protocol (FTP). FTP was instantly used to send email across a network using a mechanism very similar to mail passed on time–shared computers prior to Van Vleck's CTSS MAIL subsystem implementation. FTP relied on the early HOST protocols developed for the ARPANET–connected computers and if a user wanted to send email to a user of another networked computer he or she would log into the remote computer and leave a file for the user, just as the time–sharing users did previously.[8] As described in RFC 414, by November 29, 1972 "User–FTP" had come to encompass mail features including SNDMSG and a CALICO subsystem.

On July 20, 1971 Richard W. Watson proposed a networked email system in RFC 196, but this system was never developed. The significant advance in networked email came with the development of SNDMSG and READMAIL for the TENEX system on Digital Equipment Corporation's (DEC) Programmed Data Processor (PDP) 10 machine. The TENEX system was developed by BBN starting in 1969, and was made commercially available in 1973 (Murphy, 1989). According to M.A. Padlipsky, before Ray Tomlinson augmented SNDMSG in late 1971 some programmers had already "done a TENEX to TENEX mail hack"(Padlipsky, 2000). By the summer of 1971 Tomlinson had begun work on incorporating CYPNET[9] code in to SNDMSG, an existing non–networked mail program (Tomlinson, n d). Previously, SNDMSG was used to send local messages, or even used to send local messages from a remote Telnet connection. It is unclear when the corresponding email viewer READMSG was developed. RFC 369 "Evaluation of ARPANET Services: January through March, 1972" specifically mentions the use of SNDMSG for "Inter–personal communication", presumably across the ARPANET.

For the first five years TENEX machines and its header syntax dominated email traffic on the ARPANET (Crocker, Pogran, Vittal, & D. A. Henderson, 1977). RFC 524 proposed a networked and direct system of mail delivery, not dissimilar to Telnet (*i.e.*, remote) or FTP mail delivery. Although the system described in RFC 524 was almost certainly never developed, it was proposed that a series of commands would be invoked to facilitate direct login and delivery of email (as command and response). There was no logical separation between header syntax (destination and origin, *etc*.) and communication syntax (encoding and technical capabilities, *etc*.). Like the NLS Journal system, an IDENT code identi-

[7] A fuller study would be required to properly understand RFCs. This is a task for future research.

[8] M.A. Padlipsky argues that a decision was made in 1971 to "add mail to the [FTP] protocol". RFC 114, published on April 16, 1971 first describes FTP but makes no mention of any mail capabilities. RFC 171, published June 23, 1971, makes reference to mail systems using HOST capabilities, and thereafter references to FTP and the MAIL command become frequent throughout the RFCs.

[9] CYPNET appears to be an experimental FTP implementation, although I have been unable to locate any solid evidence on its construction or use.

fied the recipient, but using RECI syntax (for "recipient"). RFC 543 specified a mechanism to send email directly to the NLS Journal system (using either SNDMSG or FTP or Telnet). The SNDMSG syntax was "author(s), slash, recipient(s), optional semicolon and conversion algorithm," for example jew/mdk rww cr (Meyer, 1973, p. 2).

RFC 561 was published on September 5, 1973 as a stopgap measure to bring some order and interconnectivity to heterogeneous email systems, and to address obvious problems with the proposal suggested in RFC 524. Again, the proposed system was similar to Telnet or FTP mail delivery, even suggesting that existing MAIL commands or MLFL commands should be used to handle the data and login requirements. A header, or envelope metaphor was introduced, including FROM: DATE: SUBJECT: syntax, and room for a miscellaneous keyword.

There has been much debate over Tomlinson's decision to use the "commercial at" symbol (@) to divide username and host for his networked version of SNDMSG, but this decision was a trivial, although very visible, delimiter to distinguish local SNDMSG mail from networked SNDMSG mail. At this early stage in email's history the system was monolithic; SNDMSG was a basically a network application designed to send a specific type of file, not much different from the FTP MAIL command developed shortly thereafter. Later, as described in RFCs 821 and 822 the email system was split in to two logically distinct pieces.

RFC 821 and RFC 822 are arguably the most important RFCs for the history of email, marking a virtual schism. Prior to these two RFCs email is hodge–podge and entrenched in implementation, after these two RFCs email was *abstracted*. With abstraction, however, considerable documentary (and technological) complexity arose. While the "modern" era of email (after RFCs 821 and RFC 822) is characterized by two interconnected technologies (MIME and SMTP), the "medieval" era saw warring factions setting up fiefdoms. It took a more sophisticated documentary system (and various institutional organizations behind it) to set up an administration sufficiently robust to tie together all the pieces of email technology. To be sure, there are "medieval" attempts at the RFC 821 and RFC 822 split (MIME extensions were conceived in 1977), but the assemblage of socio–technical parts were not ready for the split (D. A. J. Henderson & Myer, 1977, p. 1).

By 1971–72 the 'envelope and letter' metaphor was still nascent, and email was conceptualized more like Engelbart's Journal system, taking its cue from libraries and publishing. Email had directionality due to the network communications systems, but little syntax beyond its destination. The to: header would finally be standardized with RFC 561, published September 5, 1973, although the syntax was almost certainly in use prior to this. The @ symbol to separate IDENT codes from host names (and signal a 'networked' email) was in use long before the to: syntax, destinations being specified interactively using MAIL commands, FTP, Telnet, or other mechanisms. Computers had finally come to mean much more than 'computation'.

Co–developing with email, the shift from 'number cruncher' that simply 'computes' to a 'text' machine was initially made possible by the invention of symbolic programming languages in 1947 (instead of 'direct programming') that allowed programmers to forget the materiality of code as well as the sense of 'instructing' for computation (Chun, 2005, p. 28). The new form was not just email, simultaneously it became letter writing in a foreign language.

Soon, the new model became string.h. Data typing reflects the shift from 'number crunchers' to 'text' machines (and eventually networked text machines). No historiography of data types exists, but of the first symbolic programming language (Fortran [1958], Lisp [1958], ALGOL [1958] and IBM RPG [1959]), quite significantly, none contained direct means for manipulating character or string data. Fortran contained Hollerith constants that were typeless, but the original *Fortran: Automatic Coding System for the IBM 704* manual omits mention of these constants, yet provides *two* numerical constants: fixed point and floating point.[10] Iterations of character and string data types (and functions) were to follow, including char, character, 'write text', printf and so on. These early machines were 'number crunchers', but by the late 1960s the model had shifted to fast text, formed as letters in foreign languages, and eventually in native languages as interactive programs replaced punch cards.

With war and rapid text machines, word processing became processors of sexuality, but not romance. The exclusion of women from "discursive technologies" prevents the "romantic love" of word processing, instead, "it is the business of couples who write, instead of sleep [sexually] with one another"(Kittler, 1999, p. 214). Mirroring the desiring machines of war, Deleuze and Guattari describe the desiring–production machines of email precisely:

> It is at work everywhere, functioning smoothly at times, at other times in fits and starts. It breathes, it heats, it eats. It shits and fucks. (Deleuze & Guattari, 2009, p. 1)

[10] The syntax for Hollerith constants was made explicit by Fortran 66, and is as follows: <*number of characters*>h<*characters*>.

The result of word processing is that "typed love letters… aren't love letters" (Kittler, 1999, p. 214). Word processors do not love, they fuck.

Email syntax is part of larger assemblages, both molecular and molar, interacting with extensional relations. The relations can be seen on the right side of the to: delimiter—sending an email to your boss, your lover, your friend. As Deleuze and Guattari note, "something on the order of a *subject* can be discerned on the recording surface," that is, email syntax is to:, from:, cc:, or Authentication-Results: (marking for spam), *etc*. (Deleuze & Guattari, 2009, p. 16). The recording surface of email is not like a stenograph, sent to anyone, like the game of Go "without aim or destination". Instead, email has an inescapable syntax that codes communication even when not desired. The coding is sexual:

> Just as a part of the libido as energy or production was transformed into energy of recording (Numen), a part of this energy of recording is transformed into energy of consummation (Voluptas). (Deleuze & Guattari, 2009, p. 17)

Email sent in love is not mere transcription, but a form of communication more procedural than love, perhaps something closer to consummation. As Kafka noted about his first love letter, the I, or the "nothingness that I am," disappeared under deletions or abbreviations (Elias, 2005, p. 5). The polished mirror of prose remains, only the to: or from: but not the subject. With so much lucidity in love email the Google AdSense algorithm that parses it scarcely seems out of place. Eventually, all love email turns in to AdSense, as the human relationship changes (from erotic to missionary) but the medium of communication does not; sexting is replaced by "remember the milk".

Territorializing & Deterritorializing: MIME & SMTP

History is dynamic and non–linear; this is especially the case with the history of technology. The history of catapults, for example, shows that rather than linear development from simple to complex, from small to large (or large to small), the catapult developed according to an assemblage of historical needs and accidents. Despite what older scholarship suggests, the catapult did not develop from non–torsion 'arrow–throwers' to more sophisticated torsion 'stone–throwers', and nor did the invention of a 'technologically superior' design obsolete older forms. For catapults, heavy stone throwers required administrative and labour organization to sort appropriate sized stones, and with the invention of the *trace italienne* and its low walls the trebuchet was useful for hurling fetid materials into enclosed water supplies, as the gunpowder cannon was for directly attacking low, strong walls. Social, political, and material relations—and *sheer chance*—all contributed to the assemblage. When reading early RFCs I was struck with how much of the development was backwards–looking and accommodating to interconnection with existing systems. It was widely recognized that email was an important system for any networked computers, but with ARPANET, BITNET, FidoNet, X.25 or even X.400 all offering competing technologies (at various times)—some including email replacements, some offering interconnectivity, and some completely foreign—we see historical traces of differing email technologies interacting on many different technical and communication substrates.

N. Katherine Hayles has imported from archaeology two useful concepts for understanding technological development: skeuomorphs and seriation charts. As Hayles describes it, a skeuomorph is "is a design feature, no longer functional in itself, that refers back to an avatar that was functional at an earlier time" (N. Katherine Hayles, 1994, p. 446). Further,

> skeuomorphs visibly testify to the social or psychological necessity for innovation to be tempered by replication. Like anachronisms, their pejorative first cousins, skeuomorphs are not unusual. (N. Katherine Hayles, 1994, p. 446)

In fact, once you are made aware of skeuomorphs their existence is pretty boring. Skeuomorphs act as cognitive crutches for humans, or as Hayles puts it, "skeuomorphs act… as threshold devices, smoothing the transition between one conceptual constellation and another" (N. Katherine Hayles, 1994, p. 447). In this sense, they are similar to seriation charts, which "depict… changes in an artifact's attributes [that] reveal patterns of overlapping innovation and replication" (N. Katherine Hayles, 1994, p. 445). A skeuomorph is a physical artefact testifying to an earlier design requirement (Hayles gives the example of the fake stitching on her car's vinyl dashboard), and a seriation chart is the dynamic morphology of the artefact, or, "overlapping innovation and replication" when a seemingly necessary design arises out of contingency (N. Katherine Hayles, 1994, p. 446). Seriation charts are the archeological term for those artefacts that come to be "contingently obligatory".

For email, many early design decisions seem obligatory, but to completely different systems: for example, the SOML command that maintained appearances of direct "instant messaging" (common when multiple terminals were in a centralized and local system). Likewise, email syntax was frequently delimited by special keys (CRLFs for ending lines, or

the @ symbol being repurposed from the kill command in Multics), this points back to time when control of computers was more direct—if not quite direct programming—and before the widespread use of daemons and store and forward systems. A pervasive seriation is the reoccurring use of the keyword syntax, from the NLS Journal system's model of library and publication systems, which came and went in different forms as email syntax developed. keyword syntax points to the cognitive and political challenges regarding the use of email—its widespread use as a personal communication system but its funding as a corporate or research memorandum and document system. The challenge was that research required categorization for information retrieval, yet categorization hardly made sense for personal communication. Many of the email skeuomorphs signal a pre–computer era, such as cc: (carbon–copy), bcc: (blind carbon–copy), POSTMASTER as a reserved name, and the HELO command for initiating a new MAIL connection.

Conceptualizing email systems as material helps understand the importance of skeuomorphs and seriation charts. If email developed in the noumenal world, as immaterial bits, the seriation chart no longer has the gravitas of the contingently obligatory. In fact, as Hayles argues with respect to Foucault's panopticon—that the panopticon abstracts power out of the bodies of disciplinarians to give the panopticon its force—the perception that email is an immaterial and dematerializing system is what gives email its force (N. Katherine Hayles, 1993, p. 153). The materiality of email syntax exposes lines of flight and robs email of its power. According to Hayles, a media–specific analysis is required, for failing to recognize the electronic materiality of digital texts "impedes the development of theoretical frameworks capable of understanding electronic literature as media–specific practices that require new modes of analysis and criticism" (N. Katherine Hayles, 2004, p. 71).

Yet, when Hayles calls for a "media–specific analysis" she appears to have in mind the new sense of media, from the Oxford English Dictionary: "The main means of mass communication, *esp.* newspapers, radio, and television, regarded collectively; the reporters, journalists, etc., working for organizations engaged in such communication." But, a much older sense of the word highlights a different specificity, "An intervening substance through which a force acts on objects at a distance or through which impressions are conveyed to the senses." It is this latter sense that I think highlights materially specific, radically empirical, historical aspects. Hayles' (1993) position *blocks* the possibility of (strong) posthumanism, since she rejects the material combination of human body and technological artefact. Of the phrase "He is *into* computers" she argues that it "implies that the body can flow into and occupy objects or even concepts as if they were spaces—a feat hard to imagine if the body is a material structure, but commonsensical if it is an informational pattern" (N. Katherine Hayles, 1993, p. 167). Hayles reduces technological systems to information systems. Hayles seems to have been seduced by the power of electronic systems, thinking that they dematerialize everything, leading her to postulate that

> With word processing, the touch grows lighter and the friction of textuality decreases almost to zero. The smallest keystroke can completely reformat the text, move it to a new location, or erase it altogether. (N. Katherine Hayles, 1993, p. 165)

If only the development of computer systems was so easy! From the perspective of the end–user the system does seem immaterial, as Clark's Third Law suggests: "any sufficiently advanced technology is indistinguishable from magic." Yet, a more subtle analysis reveals an assemblage of the techno–social, requiring considerable effort to stabilize identities that are contingently obligatory.

Hayles is partially correct—email is an information and communication system. But, rather than being so light to the touch as to appear immaterial, email is closer to the appearance of a postcard. Header syntax is always exposed, even in the case of encrypted messages, so unlike an anonymous letter (or even an enveloped letter), email shares qualities of mass communication. Godard compares television and film, for example, to "the sending of 25 postcards per second" (Elias, 2005, p. 5). Email is a kind of rapid postcard, like film. As an electronic postcard, email might offer an updated version of the Beat Poets' project of "open secrecy". Said to "declassify the secrets of the human body and soul" open secrecy was associated with the "the Romantic idealization of spontaneity, the letter promised to extend the originally oral, intimate, and mutual confessions of the early Beat circle" (Harris, 2006, p. 59). As communication technologies become more cyborg–like communication is likely to become less "private" in the increasingly outmoded sense of "secret", but instead more like a postcard or an open secret, simultaneously subversive and informative. Truly secret communication is more like terrorism (by the state or otherwise), whereas an open secret has political relevancy and potency.

Derrida recognized that postcards are especially open–ended communication media, since, without a stamp the postcard will never reach its destination, and words that never arrive are "rendered unreliable" (Derrida, 1987). Yet, truth can transcend media, since it does not require a stamp to arrive.

Email, like the postcard, is a media with "double potential": it can transmit a message or it can interrupt a message (if the card never reaches its destination). This is unlike the telephone or instant messaging, which is interactive and supplies contextual clues to the sender when the message has been or has not been received. Elias writes, "when the message is original, it has a textual structure; when the message is potential, it has hermeneutic structure" (Elias, 2005, p. 9). Love emails, more than any other kind,[11] are hermeneutical because context has been robbed. The material substrate is clean, almost hermetic, and directional, since to: and from: sit right at the top—perhaps spoofed, but never "your secret admirer."

The codings of technological relationships are more than just signification and discourse. Hayles argues that the "relation between assembly and compiler languages is specified by a coding arrangement, as is the relation of the compiler language to the programming commands that the user manipulates" (N. Katherine Hayles, 1993, p. 166). Indeed, the coding is a techno–social assemblage.

Cryptanalysis broke the mechanical coding of war, just as some future technology will break the coding of global networks. Already, sophisticated search engines seem poised to render seemingly intractable topological relationships visible. The black box of technology never completely opens, however, just as cryptanalysis today is exceedingly difficult against well designed algorithms of sufficient key length. With the correct key, however, the meaning is always available, just obfuscated.

Obfuscation is available by other means. The existence of spam helps hide messages from network analysis, and when combined with other stenographic techniques email can facilitate anonymous communication. New stenography techniques, such as "chaffing and winnowing" allow information that has been packetized (by TCP/IP) to hide; it takes a secret key and the complete transmission to be able to sniff the message. All stenography must be indistinguishable from the "noise floor" of the carrier, so there must be redundancy in the carrier message for stenography to hide. Cryptography and stenography may offer lines of flight, but "open secrets" might also. Any privacy enhancing technologies should be viewed with suspicion, since they enclose meaning and politics.

Summary & Future Work

This paper set out to explore a critical theory of email syntax using a historical methodology. This paper showed that from cryptography and women and war, email became a dominant communication technology. Email has its origins in time–shared computers (notably the TENEX operating system) and the NLS Journal system. Email co–developed with other computing technologies to shift from "number crunching" to a fast text machine. These machines are not just discursive, but are desiring: that is, they make love and war. Skeuomorphs and seriations identified the materiality of email syntax. Email is metaphorically associated with postcards, which were used by Beat poets as "open secrets" to subversive and political ends. Cryptography, stenography, and other codings can provide communication subversion and lines of flight, but since these communications eradicate meaning (even temporarily), we should be suspicious of the political effects of these technologies.

Further research on this topic is required. The history of computing technology in general is still very poorly understood. This paper did not trace the history into the "modern" era (past RFC 821 and RFC 822), but interesting developments have been made and are worth studying. Although figures such as Latour have been developing capable ethical and political understandings of artefacts, further research is required, especially as becoming cyborg is increasingly a real possibility.

[11] Second to love email in terms of hermeneutical structure is surely the sending of emails to bosses or PhD advisors.

References

Chun, W. H. K. (2005). On Software, or the Persistence of Visual Knowledge. *Grey Room*, 26–51.

Coeckelbergh, M. (2009). The Public Thing: On the Idea of a Politics of Artefacts. *Techné: Journal of the Society for Philosophy and Technology*, *13*(3). Retrieved from http://scholar.lib.vt.edu/ejournals/SPT/v13n3/coeckelbergh.html

Crocker, D., Pogran, K. T., Vittal, J., & Henderson, D. A. (1977). *Proposed official standard for the format of ARPA network messages*. Request for Comments. IETF. Retrieved from http://www.ietf.org/rfc/rfc724.txt

DeLanda, M. (2006). *A new philosophy of society: Assemblage theory and social complexity*. Continuum Intl Pub Group.

Deleuze, G., & Guattari, F. (1987). *A Thousand Plateaus: Capitalism and Schizophrenia*. University of Minnesota Press.

Deleuze, G., & Guattari, F. (2009). *Anti-Oedipus: Capitalism and Schizophrenia*. Penguin Classics.

Derrida, J. (1987). *The post card: from Socrates to Freud and beyond*. University of Chicago Press.

Elias, C. (2005). Transmitting (to) Derrida. In B. Sørensen (Ed.), *5 Faces of Derrida*, Working Papers (pp. 5-10). Aalborg University: Aalborg University Department of Languages and Intercultural Studies.

Engelbart, D. C. (1975). NLS Teleconferencing features: The Journal, and Shared-Screen Telephoning. *Proceedings of the CompCon75 Conference September 9-11, 1975* (pp. 173-176). Presented at the CompCon74, IEEE.

Englebart, D. C., & English, W. K. (n.d.). A research center for augmenting human intellect. *AFIPS Conference Proceedings of the 1968 Fall Joint Computer Conference* (Vol. 33, pp. 395-410). San Francisco, California: Stanford Research Institute.

Flichy, P. (2000). Internet or the ideal scientific community. *Réseaux*, *7*(2), 155–182.

Hacking, I. (1999). *The Social Construction of What?* Cambridge, Mass: Harvard University Press.

Harman, G. (2009). *Prince of Networks: Bruno Latour and Metaphysics*. Melbourne, Australia: re.press.

Harris, O. (2006). *William Burroughs and the Secret of Fascination*. SIU Press.

Hayles, N. Katherine (1993). The Materiality of Informatics. *Configurations*, *1*(1), 147-170.

Hayles, N. Katherine. (2004). Print is flat, code is deep: The importance of media-specific analysis. *Poetics Today*, *25*(1), 67–90.

Hayles, N. Katherine. (1994). Boundary Disputes: Homeostasis, Reflexivity, and the Foundations of Cybernetics. *Configurations*, *2*(3), 441-467.

Henderson, D. A. J., & Myer, T. H. (1977). Issues in message technology. *Proceedings of the fifth symposium on Data communications* (pp. 6.1-6.9). Snowbird, Utah, United States: ACM.

Kittler, F. (1999). *Gramophone, Film, Typewriter* (1st ed.). Stanford University Press.

Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Postel, J., et al. (1997). The past and future history of the Internet. *Communications of the ACM*, *40*(2), 102–108.

Meyer, N. D. (1973). *Network journal submission and delivery*. Request for Comments. IETF. Retrieved from http://www.ietf.org/rfc/rfc543.txt

Milne, E. (2000). Vicious Circles. Metaphor and the Historiography of Cyberspace. *Social Semiotics*, *10*(1), 99–108.

Murphy, D. L. (1989). Origins and Development of TOPS-20. Retrieved 22:59:25, from http://tenex.opost.com/hbook.html

Padlipsky, M. A. (2000). And They Argued All Night... Retrieved 23:20:02, from http://www.lafn.org/~ba213/allnight.html

Patton, P. (2000). *Deleuze and the Political*. Thinking the Political. London; New York: Routledge.

Saltzer, J. (2010, December 20). Source materials for CTSS Programming Staff Note 49.

Tomlinson, R. (n.d.). The first network email. Retrieved 23:30:19, from http://openmap.bbn.com/~tomlinso/ray/firstemailframe.html

Vleck, T. V. (2010, December 20). Source materials for CTSS Programming Staff Note 49.

Vleck, T. V. (n.d.). The History of Electronic Mail. *The History of Electonic Mail*. Retrieved 20:38:57, from http://www.multicians.org/thvv/mail-history.html

Wrixon, F. B. (2005). *Codes, Ciphers, Secrets and Cryptic Communication: Making and Breaking Secret Messages from Hieroglyphs to the Internet*. New York N.Y. Black Dog & Leventhal Publishers.