INFXXXX: Major Paper

Social Semiotics as Theory and Practice in Library and Information Science

Matthew Wells

#XXXXXXXX

December 14, 2012
The fields of LIS and IS are awash in "conceptual imports", to borrow the term used by Frohmann when discussing his own advocacy work for discourse analysis (1994, p. 123). Such imports necessarily bring their own discursive practices and biases into the information disciplines, as Frohmann explains it, and the changes they impose, or fail to impose, on existing power relationships within these disciplines are worth considering just as much as any methodological or epistemological benefits that they might have on offer. It is with these issues in mind that I carefully make a case in this essay for the incorporation of social semiotics paradigms and practices into information research. Social semiotics, to put it broadly, blends "traditional" semiotics – the study of signs as delineated by Ferdinand de Saussure and Charles Peirce, who were long considered to be the "co-founders" of the discipline even though they never actually coordinated their research – with critical theory. In particular, the work of linguistics scholar M.A.K. Halliday, who coined the term "social semiotics", is cited as a major influence. Early advocates of social semiotics – particularly Robert Hodge and Gunther Kress, who co-wrote one of its foundational texts – heavily criticized the work of Saussure and his followers, who they accused of disregarding the socially-constructed aspects of meaning with respect to signs and symbols. Whether or not this criticism was justified, social semiotics scholars have built an impressive foundation of research in which they apply their critical methodologies to a wide variety of cultural artefacts. More recent research has also paid particular attention to the idea of "multimodality" – that is, the use of a variety of representational structures, such as language, images, and sound, within the same text.

Social semiotics, I will argue, may prove valuable to LIS and IS by providing, among other benefits, a platform upon which seeming disparate areas of research could be brought together. For the purposes of this essay, I provide a detailed case study that focuses on a medium
that has become increasingly pervasive (and perhaps invasive) in modern information research and practice – the electronic display screen, or computer "monitor" to employ an older term. These screens, which are now embedded across multiple digital devices, have become ubiquitous in a way that the earliest computer engineers and theorists did not, and perhaps could not, envision. By treating the display screen as a semiotic "resource", to the use the proper terminology, we may situate modern screens within a historical context that will allow us to better understand the socially-constructed affordances and limitations expressed by such technology. As we will see, the ubiquity of the display has served to define and legitimize a model of human-computer interaction in which the user is placed in an extremely limiting role, allowed only to define and modify simple data elements, and then to send this data to programs that largely keep their code hidden from view. While such an approach hews closely to the *data structures* and *algorithms* dichotomy well-known to computer science scholars and programmers, it also prevents the user from fully engaging with the fundamental elements of information-based applications such as online library catalogues and search engines.

Interestingly, social semiotics, as defined by some its strongest advocates, also advocates for the proposal of new ideas and structures that may better address the issues found in the analysis of one of more resources. In my case study, then, I will introduce the concept of *language oriented programming* (LOP), an emerging field in computer science in which simple programming languages are developed to perform complex tasks. The development of LOP-based languages for applications like online library catalogues could allow for levels of user engagement and interaction that go well what exists now, and could even enlist the user as a co-creator in the development of such online applications.
I will begin this essay with a brief discussion of traditional semiotics, and then elucidate the limited ways in which semiotic theory has been applied to current LIS and IS research. Following this, I will describe the emergence of social semiotics from a historical perspective, and then go on to explain its primary tenets, at least according to some of its leading scholars. Finally, I will introduce the case study, discuss the history of the electronic display and analyze this history using an approach grounded in the principles of social semiotics. As indicated, I will finish this case study with a brief discussion of strategies that could be employed to address the issues uncovered during the analysis phase.

**Semiotics Research in LIS and IS**

Social semiotics emerged out of semiotics proper, and while it has done much to distance itself from the work of its antecedent scholars, I feel that it is necessary to survey this older work and to explore the ways in which it has been applied to information research. As we will see, semiotics has gained little traction in LIS and IS, apart from a few articles of note that I will discuss here, and social semiotics appears to be essentially a non-entity. I will begin, then, by describing the essential elements of traditional semiotics, particularly the "sign" models defined by Saussure and Peirce, and then I will discuss the ways in which semiotics has been implemented, albeit tentatively, in the information sciences.

Semiotics, in brief, is concerned with *signs*, and the meanings associated with signs. A sign is, essentially, anything that expresses meaning. This is a rather vague definition, as just about all of human experience could arguably be described as a series of encounters with meaning-making signs, but semiotics is a broad field, and it is generally left up to the researcher to devise a feasible scope for their work. Chandler, in his introductory text to the field, indicates
that "words, images, sounds, gestures and objects" tend to be the most common signs that are observed and analyzed by semiotics researchers (2007, p. 2). It is worth noting, however, that Saussure, the "founder" of one of the two main branches of semiotics, was concerned particularly with spoken language. This is perhaps counterintuitive, given our general understanding of signs as being material entities, but semiotic signs simply have to express meaning in some form or another in order to qualify as such. Saussure, in fact, approached this issue from a position that seems rather phenomenological in retrospect. He claimed that all sensory input creates "impressions" on the mind, so that, for example, an audio sign is "not actually a sound; for a sound is something physical. A sound pattern is the hearer's psychological impression of the sound" (1983, p. 66). Such impressions were similarly made by all the other senses in different circumstances.

Saussure's model of the sign contains two fundamental elements: the signifier, and the signified. The signifier is essentially the impression as discussed above, while the signified represents the meaning or "concept" associated with the sign. This might appear at first to be a rather simple model, but there is much nuance contained in what Saussure is trying to say here. The two elements of the signs – signifier and signified – are not simply separate but related components. Rather, they are both fundamentally co-dependent on each other, so that each and every sign necessarily includes both (Chandler, 2007). Saussure went so far as to say that "the two elements involved in the linguistic sign are both psychological and connected in the brain by an associative link" (1983, p. 66). Modern semiotics scholars need not dwell at this cognitive level, but the schematic relationship between signifier and signified is generally followed.

Peirce's sign model was created independently of Saussure's work, but it contains elements that are quite similar. Peirce's sign, however, contains three elements. The first is the
representamen, which is much like Saussure's signifier. The interpretant, defined by Chandler as "the sense made of the sign" (2007, p. 29), is the second element in the triad, and is reasonably close in conception to Saussure's signified, as long as we remember that Saussure considered signs to be cognitive constructions, as opposed to external objects perceived by the senses. The object, or referent, is the third element in Peirce's sign, and does in fact refer to an external reality to which the sign is associated. Again, there is much subtlety in this concept. Saussure is not implying that a sign can wholly represent an object – rather, the sign is a necessary abstraction of an object that is used to discuss and describe it (Chandler, 2007). There is much to discuss with respect to the differing ideological assumptions that informed Saussure's and Peirce's work, but for our purposes it suffices to stop here. Most of the literature I will review here tends to conflate the two models, save for Peirce's conception of the object. Whether or not this is a fair interpretation of their work is a discussion for another time.

In terms of LIS research, an article by Raber and Budd (2003) arguably comes the closest to a high-level survey of semiotics and its potential benefits for the information sciences. The authors hew largely to Saussure's sign model, though they do acknowledge Peirce's contributions to the field. In any event, they envision a role for semiotics in LIS and IS work by claiming that "all informative objects are necessarily signs, ultimately expressive of a relationship between a signifier and a signified" (p. 515). They go on to discuss the ways in which "text" and "content" may be considered as signifier and signified, respectively, for an information object/sign, and how the processes of information production and retrieval may be considered from such a perspective. For the information user, they indicate that "at the moment of retrieval, representation is exchanged for text. Upon reading, text is exchanged for content, and if access is successful, content is exchanged for knowledge" (p. 516). They then go on to
extend this conception further by incorporating more of Saussure's ideas, the details of which are not relevant here.

Budd and Raber's work is intriguing, but it appears to have had only a minimal impact upon information research. D. Wells refers to their model while looking specifically at the architecture of the OPAC, using the signifier-signified pairing to describe relationships between search terms and documents, as well as search results and documents. These ideas are intriguing, but only form one small component of a larger analysis that incorporates several theoretical models (Wells, 2007). Tredinnick (2006) also refers to the Raber/Budd article, but then dismisses Saussure and Peirce, decrying their supposedly "positivistic claims" before moving on to scholars like Barthes and Foucault who he considers more relevant. Huang (2006), on the other hand, is an enthusiastic supporter of semiotics, though, as with Raber/Budd, his article is largely advocatory. Most other references to the article are largely cursory, and, beyond these works, there is little else in information research that considers semiotics to any noteworthy degree of detail.

Semiotics, then, has still failed gained traction within the LIS and IS research communities. It might be worth analyzing in more detail why this is the case, but it would be difficult and probably inappropriate to try and discern motive here. Rather, I would prefer to move on to a discussion of social semiotics, and the reasons for which it was developed as a reaction against traditional semiotic theory. Such criticism will not focus on semiotics specifically within an information context, but the issues that are raised are certainly relevant to LIS and IS scholars.
Social Semiotics – An Introduction

Like any emerging discipline, social semiotics is built upon a foundational narrative, or perhaps narratives, that discursively objectify and normalize key assumptions with respect to antecedent scholarship. This older work is typically viewed as incomplete in the sense that entrenched methodologies were incapable of addressing important issues. Whether or not this narrative is accurate is almost always a matter for debate, but those who gravitate towards the new discipline will necessarily connect to it in their research. That is not to say that this process is uncritical, but narratives are typically exclusionary, and perhaps are inherently so. What follows, then, is a brief outline of the short history of social semiotics research from the perspectives of those who developed and propagated its origin story.

Social semiotics emerged as a somewhat antagonistic response to traditional semiotics, with Hodge and Kress serving as its chief polemicists. In their work Social Semiotics, they indicate that they do not intend to "break with the past", but they then state the following about Saussure's work:

On the one hand he projected a discipline with the widest possible scope, while on the other he laid down a set of strictures which split his heritage in two, deforming linguistics, and preventing the coming of semiotics for decades (Hodge and Kress, 1988, pp. 13-15).

Hodge and Kress criticize Saussure for many reasons, but they dwell in particular on the "sharp dichotomies" described in his Course in General Linguistics (1983).1 Saussure used these dichotomies to distinguish which aspects of language and linguistics could be studied using his conception of semiotics. According to Hodge and Kress, this strategy was unnecessarily

---

1 Course in General Linguistics, it should be noted, is actually a compilation of notes derived from lectures delivered by Saussure, and was compiled after his death.
limiting. Moreover, they accuse Saussure of picking the wrong side, so to speak, within each pair.² Working from this, they list several new areas of study that they believe should be incorporated into an "alternative semiotics", including "other semiotic systems alongside verbal language", "diachrony, time, history, process and change," and "the material nature of signs" (p. 18). By exploring such areas, they claim, semiotics would be able to understand the apparent "chaos" that lies outside of its traditional boundaries (p. 17).

*Social Semiotics* was informed and inspired by work that Hodge and Kress had done several years earlier in the field of linguistics. In *Language and Ideology* (1979), the scholars criticized the influence of Chomsky on the field, just as they would then do with Saussure and semiotics. According to them, "Chomsky's famous dictum that linguistics is a branch of cognitive psychology" had the negative effect of marginalizing approaches to linguistics that explored "social uses of language" (pp. 2-3). In particular, Hodge and Kress wished to emphasize the role that language plays in supporting *ideology*, which they define as "a systematic body of ideas, organized from a particular point of view" (p. 6). From this perspective, according to the authors, we may better understand how language operates "as instrument of control as well as communication," and, more specifically, how language "involves systematic distortion in the service of class interest" (p. 6). These concerns carried over into their later work, but with social semiotics they moved away from the position of treating language as the primary source of cultural meaning making. As they later state, "meaning resides so strongly and pervasively in other systems of meaning, in a multiplicity of visual, aural, behavioural and other codes, that a concentration on words alone is not enough" (Hodge and

² As they put it, "So important are the things that he excluded that it is tempting to see him as that useful phenomenon, the person who is *always wrong*" (Hodge and Kress, 1988, p. 17; emphasis in original).
Kress, 1988, p. vii). A critical approach to cultural artefacts, they believed, could and should apply to any and all relevant "texts", regardless of what systems they use to express meaning.

In the years after Hodge and Kress's work was published, a core group of researchers emerged as champions of social semiotics. Daniel Chandler refers to this group as the "Sydney school" because of the influence of Michael Halliday, who finished his academic career at the University of Sydney (Chandler, 2007, p. 220). Theo van Leeuwen, now one of the leading researchers in social semiotics and related fields, also works out of Sydney, and several other members of the group earned graduate degrees there. In 1991, members of the group established the academic journal *Social Semiotics*. In the first issue, the editors explained the goals of their project: "to put the social back into semiotic, and to see semiotics as a transdisciplinary site - a way of understanding the practices of meaning-making across a range of texts and institutions" (Cranny-Francis et al., 1991, p. 1). Taking texts of all forms as the focus of their work, they were concerned with "the politics of textuality, with exploring the politics of textual semiosis and of the institutional siting of text, and with the politics of the theories and methodologies which undertake that siting" (p. 1).

Social semiotic research has moved in a variety of directions since these early years, though, as noted, there seems to be a tendency to move away from research that focuses purely on language. Kress and van Leeuwen, for example, have collaborated to develop and refine a "visual grammar" to be used in the analysis of images (Kress and van Leeuwen, 2006; see also Kress and van Leeuwen, 1990, and Kress and van Leeuwen, 1996). Drawing on the critical linguistic approach seen in *Language as Ideology*, the scholars make a claim for defining a "social" grammar – that is, a system that "describes a social resource of a particular group, its explicit and implicit knowledge about this resource, and its uses in the practices of that group" –
that will "encompass oil painting as well as magazine layout, the comic strip as well as the scientific diagram" (Kress and van Leeuwen, 2006, p. 3). The term "multimodality" has also begun to emerge as a means to reference and analyze more complex cultural artefacts. According to Kress, multimodality considers "all the means we have for making meanings – the modes of representation – and considers their specific way of configuring the world" (Kress, 2004, p. 110; see also Kress, 2010). It is important to note that Kress is not conflating mode with media – rather, media are used to express specific modes, so that "modes and media exist in culturally and historically shaped 'constellations'" (Kress, 2004, p. 113). These constellations are growing increasingly complex, according to Kress, with the advent of globalization and the "vast web of intertwined social, economic, cultural and technological changes" that come with it (Kress, 2010, p. 5).

Despite all these innovations, or perhaps because of them, van Leeuwen has attempted to synthesize the basic tenets of social semiotics, at least from his perspective, in a recent work titled *Introducing Social Semiotics*. Van Leeuwen's approach focuses on the concept of the *semiotic resource*, which is meant to replace the sign as the key element of analysis. It is worth citing his definition in full here:

> In social semiotics resources are signifiers, observable actions and objects that have been drawn into the domain of social communication and that have a theoretical semiotic potential constituted by all their past uses and all their potential uses and an actual semiotic potential constituted by those past uses that are known to and considered relevant by the users of the resource, and by such

---

3 I have to emphasize that multimodality is not directly related to modality as discussed here with respect to social semiotics. The use of similar terms is unfortunately rather confusing. While multimodality refers to multiple modes of expression, modality refers to the strategies used in any specific mode to convey that an expressed meaning is true or false.
potential uses as might be uncovered by the users on the basis of their specific needs and interests. Such uses take place in a social context, and this context may either have rules or best practices that regulate how specific semiotic resources can be used, or leave the users relatively free in their use of the resource (van Leeuwen, 2006, p. 4; emphasis in original).

Based off of this conception of the semiotic resource, van Leeuwen goes on to define what he considers to be the key practices of the social semiotics researcher:

1. Collect, document and systematically catalogue semiotic resources – including their history.
2. Investigate how these resources are used in specific historical, cultural and institutional contexts, and how people talk about them in these contexts – plan them, teach them, justify them, critique them, etc.
3. Contribute to the discovery and development of new semiotic resources and new uses of existing semiotic resources (van Leeuwen, 2006, p. 3).

It probably goes without saying that these ideas and rules are not set in stone. I do find, however, that van Leeuwen's approach is quite elegant in its simplicity and flexibility. Its third "step", moreover, is extremely compelling in that it advocates for the researcher to apply knowledge gained in the analysis of a specific semiotic resource. Such applications may not be suitable in every instance, but I do make an attempt to follow this advice in the case study below, and generally aimed to hew closely to his overall ethos.
Case Study: Electronic Displays

Social semiotics is a powerful tool precisely because it allows researchers to consider a variety of phenomena under the same rubric. For the purposes of LIS and IS, it enables us to expand on existing research by looking into artefacts and texts that were previously considered to be marginally relevant, if they were considered at all. We can then study more familiar information artefacts and practices from new perspectives, gaining insights to how they function as semiotic resources, and how antecedent resources contributed to the ways in which their semiotic potential has been realized. For the purposes of this case study I will focus on a seemingly ubiquitous component of information technology: the electronic display, which may also be called the computer display or "monitor", and is often referred to simply as the "screen". The actual screen is, of course, only one component of a display, but it is by far the most relevant for the majority of computer users, in that it serves as the site for activities ranging from word processing to graphic design to online gaming. With the emergence of the computer as a viable consumer product in the 1980s – opening up a market that now supports derivative products such as the smart phone and tablet – the screen has become an increasingly pervasive medium by which to interact with information. As Lev Manovich puts it, "coupled with the computer, the screen is rapidly becoming the main means of accessing any kind of information, be it still images, moving images, or text" (2001, p. 94). To this list we may add LIS and IS tools such as online public access catalogues (OPACs), article databases, and search engines. Even when we seek physical media, we generally spend at least some time interacting with screens, and we are increasingly trending towards the use of digital media as researchers, as library patrons, and as consumers.
Electronic displays have improved significantly over the years and decades in which they have been used. While the cathode ray tube (CRT) was the primary display technology for much of this time, liquid crystal display (LCD) monitors are much more compact, and offer better image quality, and therefore have supplanted CRT as the technology of choice. Despite all this change, conceptually the display has changed very little since it was first introduced. The user guide for the *IBM 2250 Display Unit*, one of IBM's earliest electronic displays, illustrates this point quite well. The guide describes the display screen as a "virtual square grid", then notes the following: "This grid covers the 12-inch by 12-inch display area on the face of the CRT [screen], and comprises 1,024 equally spaced X positions and 1,024 equally spaced Y positions," adding "the X and Y coordinates of each display element…are specified by data in the display program" (IBM, 1971). The points along this virtual grid are what we now call *pixels*, and the dimensions of this grid are now usually referred to as a display's *resolution*. So the resolution of the IBM 2250 is 1024 x 1024, denoting the X and Y dimensions, respectively. Laptop computers built in recent years tend to use 1366 x 768 displays, but for slightly older machines 1024 x 768 was more common (Melanson, 2012). Pixels may be large or small, depending both on the screen resolution and the size of the display, a measure known as *pixel density*. The 2250 has a pixel density of roughly 85 pixels per inch (PPI). The 2048 x 1536 "Retina" display that is currently available on Apple's *iPad* offers a density of 264 PPI.

When we discuss electronic displays used with computers, then, what we are really talking about are these "pixel grids" that form the basic building blocks of all the text and graphics we see and interact with on a screen. The pixel grid display, then, is a semiotic resource that is employed virtually every time we interact with digital information, and will be the focus

---

4 This was obtained by dividing the width in pixels by the width in inches, or 1024/12 – displays that do not use square pixels require more complex formulas to determine pixel density. Note also that PPI, as opposed to points per cm (ppcm), tends to still be a popular measure due in part to the influence of the United States.
of this case study. Having said this, it may seem odd that the earliest computers, developed in the years following the Second World War, did not employ any kind of electronic display whatsoever. By this point, the cathode ray tube had been in existence for several decades, and CRT television sets were beginning to emerge as a major consumer product. So why was the CRT display left out of the plans for the first computer systems?

While there are probably many factors in play here, a simple explanation might be that there was simply no recognizable need for CRT display screens. The first computers were designed largely to process data a piece at a time (or perhaps a few pieces at a time), with punched card and magnetic tape serving as the primary media by which such data was input and output. As such, the functionality of these machines was rather limited. The ENIAC, the first general-purpose computer to be built in the United States, was designed "to assist the U.S. Army in calculating firing tables required for the various new weapons and ammunition then being developed for the conduct of World War II" (Polachek, 1997, p. 25).\(^5\) Remington Rand's UNIVAC I, the first American computer made specifically for the commercial market, was described by the company as "an electronic computer designed to perform repetitive clerical and mathematical computations at a high rate of speed and accuracy" (Remington Rand, 1954, p. 1).\(^6\) Data was fed into the UNIVAC via magnetic tape, and, in a process somewhat akin to a digital assembly line, the data would be processed, printed onto another reel of magnetic tape, and then the process would start over. Output data could then be printed out into a more readable format (Remington Rand, 1954).

It was only through tentative experimentation that the CRT display was introduced into computer system architectures. Such experiments were conducted by researchers working

\(^5\) ENIAC was not actually fully-built until 1946, however.
\(^6\) A UNIVAC system was famously used by CBS during its coverage of the 1952 presidential election and famously predicted the outcome hours in advance (Bohn, 1980).
largely out of university laboratories, equipped with enough resources to allow them some
leeway to try out new ideas. It is generally accepted, then, that the first applications of CRT
displays to computing systems occurred within the context of the "Whirlwind" project at MIT.

Whirlwind was one of the most unusual of the early university-based computer-building projects
– the original Whirlwind was commissioned in 1945 by the United States Navy, in partnership
with the Servomechanics Laboratory as MIT, to be an all-purpose, mechanical flight simulator.
The project evolved quickly to become both a mechanical simulator and a digital computer –
eventually, the simulator aspect was dropped, so Whirlwind was, for a short time, an
experimental digital computer with no specific purpose (Redmond and Smith, 1980).

It was within this chaotic context that the CRT display was introduced. Initially, in 1948,
the only display device in use was a 5-inch oscilloscope, and its purpose was simply to depict the
values of certain memory registers onscreen. Norm Taylor, who worked on the Whirlwind
project, noted the following in a presentation he made at the SIGGRPAH conference in 1989:

Keep in mind we were not trying to build a display here; we were building a
computer. All we used the display for was testing the various parts of the system
so displays were ancillary completely to the main event (Hurst et al., 1989, p. 22).

Taylor went on to discuss a program he called "Waves of One", which displayed the binary
values of a select group of memory registers as a series of dots – the value 1 would be
represented as a dot, while a 0 would be represented as a blank space. The result was a 16 x 16
grid of dots and spaces. This same program, moreover, allowed for the manipulation of these
values via a "light gun", which was essentially a light-sensitive wand that could be used to
interact with the CRT screen. Touching a dot or space on the screen would toggle the value
embedded within the relevant location in internal memory, allowing the user to create and erase dots at will (Taylor, 1989).

As they continued to acquire better displays (i.e. displays with higher resolutions), Whirlwind's engineers and programmers quickly became more creative, and recognized that they could use the display to plot graphs and otherwise graphically represent data that had been processed by the computer. By 1949, the display was considered to be an important element of the overall machine, albeit one with a limited, specific role to play, as this internal report indicates:

The display equipment now in use with WWI is intended primarily for demonstration purposes. It gives a qualitative picture of solutions to problems set up in test storage, and it illustrates a type of output device that can be used when data are desired in graphical rather than numerical form (Servomechanisms Laboratory, 1949, p. 29).

Graphical form typically meant a line graph (or graphs) representing the solutions to one or more equations. While the raw data could be output via tape or printer, the display could be used to make this data visually compelling.

It was this type of work that gave rise to a program known as "Bouncing Ball", arguably the first application that truly showed off the capabilities of the CRT-equipped computer. Bouncing Ball was simply a program that could solve a specific set of differential equations, but, working with the display, it used these solutions to present an animation of sorts. A "ball" – really just a single pixel on the display – would appear near the top-left corner of the screen. As it moved from left to right, it would "fall" – that is, move vertically down the screen – until it hit a horizontal line, at which point it would "bounce" and move vertically up the screen. The end
result was "a series of parabolas of decreasing height" that traced the movement of the ball/pixel (Pias, 2006, p. 169). Without knowing the inner workings of the machine – which was the case for many visitors who witnessed the program in action – there were no clues on the display that suggested that Whirlwind was processing complex iterative mathematical functions in order to produce this animation. This "black boxing" of critical operations would become a mainstay of human-computer interaction.

The experimentation with Bouncing Ball did not end there, however. As Taylor explains it, the program became much more intriguing when an interactive component was incorporated:

A little later [Bouncing Ball developers] Adams and Gilmore decided to make the first computer game, and this was also in '49. This is a more interesting display. You see that the bouncing ball finds a hole in the floor and the trick was to set the frequency such that you hit the hole in the floor. This kept a lot of people interested for quite a while and it was clear that man-machine interaction was here to stay. Anyone could turn the frequency-knobs (Norman, 1989, p. 21; see also figure 1).7

While his claim that this was the "first computer game" is debatable, the description he provides here hints at a framework that would be replicated by many, if not most, computer games and other digital games. The "player" is presented with a limited amount of information onscreen – a line with a "hole" (really just a gap in a line) that acts as a target – and demands a level of interactivity that implicates the player in a limited relationship with the computer program. By adjusting an unstated "frequency", the player attempts to get the ball to fall in the hole. The player – and this is crucial – may only perform this one specific action, and must adjust his or

---

7 Taylor’s claim that this was the "first computer game" is debatable, though a good case could be made. Arcade game distributor BMI Gaming has an impressively complete discussion of this issue on its website, though it is lacking in citations (BMI Gaming, n. d.)
her behaviour based purely on the output produced on the digital display. This is now a familiar approach to digital gaming. As software studies scholar Noah Wardrip-Fruin puts it, "successful [digital game] play requires understanding how initial expectation differs from system operation, and incrementally building a model of the system's internal processes based on experimentation" (2006, p. 302). As we will see, however, it is not just games that require the user to simply "play" with limited control over the system.

Moving a little further ahead in time to 1963, MIT was the site of one of the most innovative applications of the CRT monitor – Sketchpad, an interactive drawing and design program developed by PhD student Ivan Sutherland. To use the opening line of his dissertation introduction, Sutherland claims that Sketchpad "makes it possible for a man and a computer to converse rapidly through the medium of line drawings" (Sutherland, 1963, p. 1). To put this more technically, Sketchpad enables its users to interact directly with a CRT display via a "light pen" – which was similar to the light gun used with Whirlwind – to produce drawings and diagrams. Despite its name, the program did not let users draw freehand – rather, users would use the light pen to pinpoint the vertices of a particular shape, and the program would then construct the shape using line segments. Functionality to allow circles and curves was also included, and the program also allowed for easy replication of common elements (Sutherland, 1963).

The impact of Sketchpad on the history of computer software is difficult to overstate. It is generally recognized as the forerunner of virtually all computer-assisted drawing (CAD) software in existence, and is often credited as the inspiration for the graphical user interface (GUI), as well as providing "fundamental work on iconic representations, object-oriented

---

8 Wardrip-Fruin was referring to a specific type of digital game here – strategy/simulation games like SimCity – but I think this model may be applied more widely to describe virtually all digital games, at least in part.
techniques, constraints, interaction techniques, and approaches to animation" (Grudin, 2006, p. 45). For our purposes, it is also worthwhile to know that Sketchpad was developed at MIT's Lincoln Laboratories, a computer technology lab that was built out of, and as a direct result of, the Whirlwind project. It should come as no surprise, then, that it was here that the electronic display was put to such useful purposes. While Sketchpad helped to open new horizons for computer applications, however, it also served to legitimize and normalize limited, and limiting, paradigms by which computers presented information and allowed users to manipulate such information. A social semiotics approach to this issue will demonstrate why and how this happened.

Treating the electronic display screen as a semiotic resource, we can see how its semiotic potential was defined by Sketchpad and later programs, and how this potential was in part a carryover from earlier applications of CRT displays such as Bouncing Ball. Sutherland presented his program as a means to "converse" with a computer, a term suggesting direct, and seemingly equitable, communication between users and computing systems. This notion persists in studies of Sketchpad and its influence: Manovich, for example, states that "with Sketchpad, a human operator could create graphics directly on a computer screen by touching the screen with a light pen" (2001, p. 102). Myers writes that "the now ubiquitous direct manipulation interface, where visible objects on the screen are directly manipulated with a pointing device, was first demonstrated by Ivan Sutherland in Sketchpad" (1998, p. 45). Direct and equitable human-computer interaction, it would seem, is a critical aspect of the semiotic potential of the CRT display screen, and it was programs like Sketchpad that made such interactions possible. This would not be an accurate assessment, however, and in fact all this talk of directness serves to distort a much more complex reality.
To understand what is really going on here, we have to examine the computer display within its full context. Manovich actually hints at the true heart of the matter when he states the following:

Sketchpad exemplified a new paradigm of interacting with computers: By changing something on the screen, the operator changed something in the computer's memory. The real-time screen became interactive (2001, p. 102).

While he continues to employ the concept of direct interaction (here expressed as "real-time" interaction), Manovich also makes the key observation that these interactions actually involve the manipulation of a computer's "memory". This statement echoes Taylor's description of Waves of One, which allowed for a more rudimentary control over values stores in computer memory. Note also that Sutherland, in the abstract for his dissertation, states the following:

The Sketchpad system uses drawing as a novel means of communicating with a computer. The system contains input, output, and computation programs that enable it to interpret information drawn directly on a computer display (1963, p. iii).

Here we get a more complete understanding of what is really going on, then. While users interact with Sketchpad via the light pen and screen, and the computer "responds" to these interactions via the same screen, there is also a great deal of work going on within the computer system – that is, programs to handle input, output, and other critical processes – that is entirely invisible to the user. These programs, created by Sutherland, governed the functionality of Sketchpad in very specific ways. The user could not simply "draw" with the light pen – he or she had to obey the rules laid out by Sketchpad's program(s) in order to produce digital images. With a series of specific gestures, for example, the vertices of a polygon could be laid out, and its
boundary defined with a series of line segments. If these gestures were not followed exactly, the result would not be what the user had intended.

This all may seem somewhat obvious for those of us living in an age in which all manner of information is produced in controlled digital environments (via word processor, graphics editor, and database, among other tools), but that is only because we implicitly accept the limited semiotic potential afforded to the technologies we use. Our own interactions with computers are in fact governed by the same rules that defined these early experiments with electronic displays. To bring in the necessary technical language, what we experience is an interplay between what computer scientists define as data structures and algorithms. According to a classic textbook on the subject, a data structure is a means by which to "store and organize data in order to facilitate access and modifications," while an algorithm is "any well-defined computational procedure that takes some value, or set of values [i.e. structured data], as input and produces some value, or set of values, as output" (Cormen et al., 2009, pp. 1, 9; emphasis in original). Manovich claims that, in computer programming, "the world is reduced to two kinds of software objects that are complementary to each other – data structures and algorithms" (2001, p. 223). This is a rather bold statement – in truth, the data structures and algorithms model is but one of many possible theoretical configurations. But Manovich's statement makes clear that it is the model that has come to predominate in modern computing.

According to the data structures and algorithms paradigm, human-computer interactions proceed along a specific path: data is input into the computer, processed by algorithms, and then new data is made and presented (i.e. output) to the user, at which point the cycle may repeat. A modern GUI allows for many of these cycles to occur simultaneously, but each cycle unfolds essentially in this manner. This process, then, creates critical limitations with respect to the role
of the user. To take the example of Sketchpad, the user may manipulate information on the CRT display – and therefore manipulate specific data elements contained within the computer's memory – but he or she cannot modify the algorithms by which the program interprets and processes these data manipulations. In terms of semiotic potential, then, the electronic display enables only those actions that impact data stored in memory, and not the actual program(s) that are at the heart of Sketchpad. This paradigm, moreover, has been carried over into virtually all modern computer software.9

This is not the end of the story, however. While the electronic display allows only for limited interaction with the computer, it is generally thought about and talked about as a resource that allows for "direct" communication with computer systems. There is where the socially-constructed aspect of the semiotic resource model is crucial. We have already seen several examples of texts in which screen interactions are defined in these terms. Such statements operate by emphasizing the supposed importance of data as the primary resource used by computer hardware and software, while simultaneously minimizing the importance of algorithms, often simply by ignoring their existence. The operating manual for the IBM 2250 Display Unit, for example, claims that the display gives the user "direct and rapid access to stored data which can be scanned visually, selected, processed, modified, and redisplayed in alphanemic [sic] and graphic representation" (IBM Systems Reference Library, 1971, p. 5). In an online advertisement for the high-resolution "Retina display" available on its newest-generation

---

9 There is a "grey area" here that needs to be addressed. When software developers create programs, they do so now largely using the same input and output devices – mouse, keyboard, and display – that are employed by users. Programmers use software – ranging from simple text editors to full-fledged integrated development environments (IDEs) – that allows them to write and compile programs for specific operating systems. As will be discussed in more detail below, such software allows programmers to interact with computer systems at a more fundamental level – a level which is denied to most users of such software. Even then, such access is still limited. Most programmers, for example, use third-party compilers – the inner workings of which are largely inaccessible – to turn their programs into machine-readable code.
iPad, Apple states the following: "pick up the iPad with Retina display and suddenly, it's clear. You're actually touching your photos, reading a book, playing the piano. Nothing comes between you and what you love." These statements define data and users, and the interactions between them, as the foundational element of computing, and leave out the fact that there are rules that define explicitly how such interactions will unfold, and that these rules, moreover, are defined in advance by developers, and cannot be changed. The user is relegated to the role of input "producer", feeding data into algorithms so it may be processed and output. Yet the nature of this role is obscured by the texts that serve to define, legitimize, and normalize the electronic display as a semiotic resource.

By examining the display using a social semiotics approach, then, we are able to expose the historical elements that have gone into shaping how it is recognized and used as a semiotic resource, and how its semiotic potential has been shaped – and distorted – over time. From here, we can sketch out ways in which these insights may be leveraged to potentially reshape the display as a resource that is more beneficial to users. If we look at modern OPACs, article databases, Web search engines, and similar tools, we can see that the same limitations have been imposed on users in terms of information manipulation. These applications allow users to enter search terms, send search "requests" to computer systems, and read the "results" of a search from an electronic display (and, when relevant, to connect to websites and/or electronic documents). While the output data is visible, the actual algorithms used in these searches remain hidden. The display once again serves to legitimize and normalize the user's role as a simple data creator.

How may this system be improved upon? An obvious approach would be to allow the user increased access to the algorithmic code by which search functions are executed. This may lead us to consider an open-source software (OSS) approach when developing search software.

10 http://www.apple.com/ipad/features/
If the user could manipulate onscreen the code that directed how a search was conducted, then they could, possibly, customize this code to better serve their needs. Such an approach would necessitate the recasting of the digital display as a means by which users may manipulate code, as well as data. Programmers have been using displays for such purposes for many years, of course, but the user is generally defined as someone who does not get involved in such messy matters. Systems such as the iPad, for which Apple requires developers to purchase a special licence in order to develop applications, make such interactions even more unfeasible. By promoting more open and accessible development environments, then, the user could be empowered to use the display as more than just a means to exchange data with invisible programs.

The major problem with this vision is that the source code that is used to build a typical search application is often extremely complex, involving algorithms that connect search terms to information sources using mathematical and statistical functions designed to search through vast amounts of data as quickly as possible. To understand how these algorithms operated would require an expertise in programming equal to that of the developers who created the search application itself. While such expertise is undoubtedly useful, it would be a long process to bring all or most computer users up to speed, and it is difficult to predict how willing users would be to engage in such a process.

An alternative, then, might be something that allows for at least some of the customization offered in the OSS approach, but using a development system that may be learned and used more quickly. Interestingly, an emerging field within computer science knows as language oriented programming (LOP) addresses similar concerns. LOP is an approach to the development of programming languages that focuses on using short, simple constructions with
which to complete complex tasks, and has arisen as existing programming languages and software have grown increasingly complex. As explained by Dmitriev, one of LOP's major advocates, "programmers today have very restricted freedom. Sure, I can do anything on a computer, but some things take me years of effort when it should take much less time. Something is wrong here" (Dmitriev, 2004, p. 1). LOP attempts to address these issues by focusing on very specific problems, and then developing simple, or "high-level" languages that can be used to execute complex actions specific to such problems. As Ward, another key proponent of LOP, explains it, "with a problem-specific very high level language, a few lines of code are sufficient to implement highly complex functions" (Ward, 1994, p. 6).

It is not overly difficult to imagine how such an approach might benefit users interacting with search applications. Searching is a very specific problem, particularly when you consider one specific application, such as a particular online catalogue or search engine. With the right LOP-influenced language, an application interface could be altered to allow users to build their own search algorithms, and then feed those into the search system. Such an approach, in fact, could alter the very conception of the passive "user", giving individuals the means to participate in and actively modify the very mechanisms that govern the operation of software applications. Such active users could come to see the electronic display as a much more valuable resource in terms of enabling interactions with computer systems, and they could therefore engage more meaningfully with the programs that they use on a regular basis.

Conclusions

This case study was a cursory attempt to cover the major aspects of a social semiotic analysis as defined by van Leeuwen, and could be expanded upon on a number of fronts. One
major issue that was hinted at many times, but not discussed in detail, is the means by which users interact with both computer and display. The light pen, which arrives on the scene at a surprisingly early date, is one such tool, but mouse (or touchpad) and keyboard are more familiar devices to the contemporary computer user. Direct interaction with the screen, however, has made a comeback with the advent of smart phones and tablets, though it should be noted that these technologies function much differently than a light pen. In any event, such devices could easily be considered as semiotic resources for making meaning, and might possibly exert a significant influence on how users of different devices perceive and consider the information displayed on their screens. I would argue that they do not fundamentally alter the relationships defined in my case study, however, but rather would provide nuance and texture to my analysis.

In any event, social semiotics is still a relatively new field, with a small core of key adherents and advocates. The ideas and problems that it concerns itself with, however, are in line with those of other critical approaches that are more familiar. A key advantage of social semiotic research is the ease by which it integrates a multitude of expressive channels – text, images, sound, etc. – and considers them as parts of a whole, multimodal, meaning-making system. Kress and van Leeuwen, in particular, have been moving in this direction, but hopefully many more scholars will follow suit, adding their own insights into the larger project. Much work remains to be done with respect to the theoretical side of social semiotics, which is perhaps one of the main reasons why it is advantageous for LIS and IS scholars to get involved now. Social semiotics as a field, in turn, would greatly benefit from the insights provided by such scholars, who have been examining many of the same issues it is concerned with since the days in which semiotics was still largely grounded in the work of Saussure and Peirce.
Works Cited


