Alan Kay's Universal Media Machine

Medium:
8.
a. A specific kind of artistic technique or means of expression as determined by the materials used or the creative methods involved: the medium of lithography.

b. The materials used in a specific artistic technique: oils as a medium.

American Heritage Dictionary, 4th edition (Houghton Mifflin, 2000).

"The best way to predict the future is to invent it."
Alan Kay

Introduction

One way to understand how computerization changed the media we use to represent the world, record our ideas and communicate with others is to examine the work of the people who invented the paradigms and practical techniques of computer media. And one of the key figures in this history is Alan Kay.

Building on the previous work of Ivan Sutherland, Ted Nelson, Douglas Englebart, J.C. Licklider, Seymour Papert, and others, the Learning Research Group at Xerox PARC headed by Kay systematically articulated the paradigm and the technologies of *vernacular media computing,* as it exists today.[1]

[1] Kay has expressed his ideas in a few artciles and a large number of interviews and public lectures. The following have been my main primary sources: Alan Kay and Adele Goldberg, *Personal Dynamic Media*, IEEE Computer. Vol. 10 No. 3 (March), 1977; my quotes are from the reprint of this article in *New Media Reader*, eds. Noah Wardrip-Fruin and Nick Montfort (The MIT Press, 2003); Alan Kay, "The Early History of Smalltalk, " (HOPL-II/4/93/MA, 1993); Alan Kay, "A Personal Computer for Children of All Ages,"

Although selected artists, filmmakers, musicians, and architects were already using computers since the 1950s, often developing their software in collaboration with computer scientists working in research labs such as Bell Labs and IBM Watson Research Center, most of this software was aimed at producing only particular kind of images, animations or music congruent with the ideas of their authors. In addition, each program was designed to run on a particular machine. Therefore, these software programs could not function as general-purpose tools easily usable by others.

It is well known most of the key ingredients of personal computers as they exist today came out from Xerox PARC: Graphical User Interface with overlapping windows, bitmapped display, networking via Ethernet, mouse, laser printer, and WYIWYG ("what you see is what you get") printing. But what is equally important is that Kay and his colleagues also developed a range of applications for media manipulation and creation which all used a graphical interface. They included a word processor, a file system, a drawing and painting program, an animation program, a music editing program, etc. Both the general user interface and the media manipulation programs were written in the same programming language Smalltalk. While some the applications were programmed by members of Kay's group, others were programmed by the users that included seventh-grade high-school students.[2] This was consistent with the essence of Kay's vision: to provide users with a programming environment, examples of programs, and already-written general tools so the users will be able to make their own creative tools.

Proceedings of the ACM National Conference, Boston, August 1972; Alan Kay, *Doing with Images Makes Symbols* (University Video Communications, 1987), videotape (available at www.archive.org); Alan Kay, Alan Kay, "User Interface: A Personal View," p. 193. in The Art of Human-Computer Interface Design, 191-207. Editor Brenda Laurel. Reading, Mass," Addison-Wesley, 1990.; David Canfield Smith at al., "Designing the Star user Interface," *Byte*, issue 4 (1982).

[2] Alan Kay and Adele Goldberg, "Personal Dynamic Media*,"* in *New Media Reader*, eds. Noah Wardrip-Fruin and Nick Montfort (The MIT Press, 2003), 399.

Remediation versus Function

As a result of the adoption of GUI in the 1980s, software has replaced many other tools and technologies for the creative professionals and it has also given hundreds of millions of people the abilities to create, manipulate, sequence and share media – but has this lead to the invention of fundamentally *new* forms of culture? Today computer scientists along with media companies are busy inventing electronic *books* and interactive *television*; the consumers are happily purchasing (or downloading for free) *music albums* and *feature films* distributed in digital form, as well making *photographs* and *video* with their digital cameras and cell phones; office workers are reading PDF *documents which imitate paper*. (And even at the futuristic edge of digital culture - smart objects/ambient intelligence – traditional forms persist: Philips showcases “smart” *household mirror* which can hold electronic notes and videos, while its director of research dreams about a normal looking *vase* which can hold digital photographs.[3])

In short, the revolution in means of production, distribution, and access of media has not been accompanied by a similar revolution in syntax and semantics of media. Should not we blame Kay and his collaborators at PARC for making digital computers imitate older media? By systematically developing easy to use GUI-based software to create and edit familiar media types, Kay and others appear to lock a computer into being a simulation machine for “old media.” Or, to put this in terms of Bolter and Grusin’s influential book *Remediation: Understanding New Media* (2000), we can say that GUI-based software turned digital computers into a “remediation machine:” a machine that expertely represents a range of earlier media. (Other technologies developed at PARC such as bitmapped color display used as the main computer screen, laser printing, and the first Page Description Language which eventually lead to Postscript were similarly conceived to support computer’s new role as a machine for simulation of physical media.)

[3] http://www.research.philips.com/newscenter/pictures/display-mirror.html/.

Bolter and Grusin define remediation as "the representation of one medium in another."[4] According to their argument, new media always remediate the old ones and therefore we should not expect that computers would function any differently. This perspective emphasizes the continuity between computational media and earlier media. Rather than being separated by different logics, all media including computers follow the same logic of remediation. The only difference between computers and other media lies in how and what they remediate. As Bolter and Grusin put this in the first chapter of their book, "What is new about digital media lies in their particular strategies for remediating television, film, photography, and painting." In another place in the same chapter they make an equally strong statement that leaves no ambiguity about their position: "We will argue that remediation is a defining characteristic of the new digital media."

It we consider today all the digital media created by both consumers and by professionals– digital photography and video shot with inexpensive cameras and cell phones, the contents of personal blogs and online journals, illustrations created in Photoshop, feature films cut on AVID, etc. – in terms of its appearance digital media indeed often looks exactly the same way as it did before it became digital. Thus, if we limit ourselves at looking at the media *surfaces*, remediation argument accurately describes much of computational media. But rather than accepting this condition as an inevitable consequence of the universal logic of remediation, we should ask why this is the case. In other words, if contemporary computational media imitates other media, how did this become possible? There was definitely nothing in the original theoretical formulations of digital computers by Turing or Von Neumann about computers imitating other media such as books, photography, or film.

[4] Bolter and Grusin, *Remediation: Understanding New Media* (The MIT Press, 2000).

The conceptual and technical gap which separates first room size computers used by military to calculate the shooting tables for anti-aircraft guns and crack German communication codes and contemporary small desktops and laptops used by ordinary people to hold, edit and share media is vast. The contemporary identity of a computer as a media processor took about forty years to emerge – if we count from 1949 when MIT's Lincoln Laboratory started to work on first interactive computers to 1989 when first commercial version of Photoshop was released. It took generations of brilliant and very creative thinkers to invent the multitude of concepts and techniques that today make possible for computers to "remediate" other media so well. What were their reasons for doing this? What was their thinking? In short, why did these people dedicated their careers to inventing the ultimate "remediation machine"?

I cannot consider the thinking of all the key figures in the history of media computing in the space of one article. But we can take a closer look at one place where the identity of a computer as a "remediation machine" was largely put in place – Alan Kay's Learning Research Group at Xerox PARC that was in operation during the first part of the 1970s. We can ask two questions: first, what exactly Kay wanted to do, and second, how he and his colleagues went about achieving it. The brief answer – which will be expanded below - is that Kay wanted to turn computers into a "personal dynamic media" which can be used for learning, discovery, and artistic creation. His group achieved this by systematically simulating most existing media within a computer while simultaneously adding many new properties to these media. Kay and his collaborators also developed a new type of programming language that, at least in theory, would allow the users to quickly invent new types of media using the set of general tools already provided for them. All these tools and simulations of already existing media were given a unified user interface designed to activate multiple mentalities and ways of learning - kinesthetic, iconic, and symbolic.

Kay conceived of “personal dynamic media” as a fundamentally new kind of media with a number of historically unprecedented properties such as the ability to hold all of user’s information, simulate all types media within a single machine, and “involve learner in a two-way conversation.”[5] These properties enable new relationships between the user and the media she may be creating, editing, or viewing on a computer. And this is essential if we want to understand the relatiosnhips between computers and earlier media. Briefly put, while visually computational media may closely mimic other media, these media now function in diffirent ways.

For instance, consider digital photography that often does imitate in appearance traditional photography. For Bolter and Grusin, this is example of how digital media ‘remediates” its predecessors. But rather than only paying attention to their appearnace, let us think about how digital photographs can function. If a digital photograph is turned into a physical object in the world – an illustration in a magazine, a poster on the wall, a print on a t-shirt – it functions in the same ways as its predecessor.[6] But if we leave the same photograph inside its native computer environment – which may be a laptop, a network storage system, or any computer-enabled media device such as a cell phone which allows its user to edit this photograph and move it to other devices and the Internet – it can function in ways which, in my view, make it radically different from its traditional equivalent. To use a different term, we can say that a digital photograph offers its users many affordances that its non-digital predecessor did not. For example, a digital photograph can be quickly modified in numerous ways and equally quickly combined with other images; instantly moved around the world and shared with

---

[5] Since the work of Kay’s group in the 1970s, computer scientists, hackers and designers added many other unique properties – for instance, we can quickly move media around the net and share it with millions of people using Flickr, youtube, and other sites.

[6] However consider the following examples of things to come: “Posters in Japan are being embedded with tag readers that receive signals from the user’s ‘IC’ tag and send relevant information and free products back.” Takashi Hoshimo, “Bloom Time Out East,” *ME: Mobile Entertainment,* November 2005, issue 9, p. 25 <www.mobile-ent.biz>.

other people; and inserted into a text document, or an architectural design. Furthermore, we can automatically (i.e., by running the appropriate algorithms) improve its contrast, make it sharper, and even in some situations remove blur.

Note that only some of these new properties are specific to a particular media – in our example, a digital photograph, i.e. an array of pixels represented as numbers. Other properties are shared by a larger class of media species – for instance, at the current stage of digital culture all types of media files can be attached to an email message. Still others are even more general features of a computer environment within the current GUI paradigm as developed thirty years ago at PARC: for instance, the fast response of computer to user's actions which assures "no discernable pause between cause and effect."[7] Still others are enabled by network protocols such as TCP-IP that allows all kinds of computers and other devices to be connected to the same network. In summary, we can say that only some of the "new DNA" of a digital photograph are due its particular place of birth, i.e., inside a digital camera. Many others are the result of current paradigm of network computing in general.

"Simulation is the central notion of the Dynabook"

While Alan Kay articulated his ideas in a number of articles and talks, his 1977 article co-authored with one of his main PARC collaborators, computer scientist Adele Goldberg, is particularly useful resource if we want to understand contemporary computational media. In this article Kay and Goldberg describes the vision of the Learning Research Group at PARC in the following way: to create "*a personal dynamic medium* the size of a notebook (the Dynabook) which could be owned by everyone and could have the power to handle virtually all of

[7] Kay and Goldberg, *Personal Dynamic Media*, 394.

its owner's information-related needs."[8] Kay and Goldberg ask the readers to imagine that this device "had enough power to outrace your senses of sight and hearing, enough capacity to store for later retrieval thousands of page-equivalents of reference materials, poems, letters, recipes, records, drawings, animations, musical scores, waveforms, dynamic simulations and anything else you would like to remember and change."[9]

In my view, "all" in the first statement is important: it means that the Dynabook – or computational media environment in general, regardless of the size of a form of device in which it is implemented – should support viewing, creating and editing all possible media which have traditionally were used for human expression and communication. Accordingly, while separate programs to create works in different media were already in existence, Kay's group for the first time implemented them all together within a single machine. In other words, Kay's paradigm was not to simply create a new type of computer-based media which would co-exist with other physical media. Rather, the goal was to establish a computer as an umbrella, a platform for *all* already existing expressive artistic media. (At the end of the article Kay and Goldberg give a name for this platform – "metamedium.") This paradigm changes our understanding of what media is. From Lessing's *Laocoon; or, On the Limits of Painting and Poetry* (1766) to Nelson Goodman's *Languages of Art* (1968), the modern discourse about media depends on the assumption that different mediums have distinct properties and in fact should be understood in opposition to each other. Putting all mediums within a single computer environment does not necessary erases all differences in what various mediums can represent and how they are perceived – but it does bring them closer to each other in a number of ways. Some of these new connections were already apparent to Kay and his colleagues; others became visible only decades later when the new logic of media set in place at PARC unfolded more fully; some maybe still not visible to us today because they have

[8] Ibid., 393. The emphasis in this and all following quotes from this article in mine – L.M.
[9] Ibid., 394.

not been given practical realization. One obvious example such connections is the emergence of multimedia as a standard form of communication: web pages, PowerPoint presentations, multimedia artworks, mobile multimedia messages, media blogs, and other communication forms which combine few mediums. Another is the rise of common interface conventions and tools which we use in working with different types of media regardless of their origin: for instance, a virtual camera, a magnifying lens, and of course the omnipresent copy, cut and paste commands.[10] Yet another is the ability to map one media into another using appropriate software – images into sound, sound into images, quantitative data into a 3D shape or sound, etc. – used widely today in such areas as DJ/VJ/live cinema performances and information visualization. All in all, it is as though different media are actively trying to reach towards each other, exchanging properties and letting each other borrow their unique features. (This situation is the direct opposite of modernist media paradigm of the early twentieth century which was focused on discovering a unique language of each artistic medium.)

Alan Turing theoretically defined a computer as a machine that can simulate a very large class of other machines, and it is this simulation ability that is largely responsible for the proliferation of computers in modern society. But as I already mentioned, neither he nor other theorists and inventors of digital computers explicitly considered that this simulation could also include media. It was only

[10] This elevation of the techniques of particular media to a status of general interface conventions can be undestood as the further unfolding of the principles developed at PARC in the 1970s. Firstly, the PARC team specifically wanted to have a unified interface for all new applications. Secondly, they developed the idea of “universal commands” such as “move,” “copy,”and “delete.” As described by the designers of Xerox Star personal computer released in 1981, “MOVE is the most powerful command in the system. It is used during text editing to rearrange letters in a word, words in a sentence, sentences in a paragraph, and paragraphs in a document. It is used during graphics editing to move picture elements, such as lines and rectangles, around in an illustration. It is used during formula editing to move mathematical structures, such as summations and integrals, around in an equation.” David Canfield Smith et al., “Designing the Star User Interface,” *Byte*, issue 4/1982, pp. 242-282.

Kay and his generation that extended the idea of simulation to media – thus turning Universal Turing Machine into a Universal Media Machine, so to speak.

Accordingly, Kay and Goldberg write: “In a very real sense, simulation is the central notion of the Dynabook.”[11] When we use computers to simulate some process in the real world – the behavior of a weather system, the processing of information in the brain, the deformation of a car in a crash – our concern is to correctly model the necessary features of this process or system. We want to be able to test how our model would behave in different conditions with different data, and the last thing we want to do is for computer to introduce some new properties into the model that we ourselves did not specify. In short, when we use computers as a general-purpose medium for simulation, we want this medium to be completely “transparent.”

But what happens when we simulate different media in a computer? In this case, the appearance of new properties may be welcome as they can extend the expressive and communication potential of these media. Appropriately, when Kay and his colleagues created computer simulations of existing physical media – i.e. the tools for representing, creating, editing, and viewing these media – they “added” many new properties. For instance, in the case of a book, Kay and Goldberg point out “It need not be treated as a simulated paper book since this is *a new medium with new properties.* A dynamic search may be made for a particular context. The non-sequential nature of the file medium and the use of dynamic manipulation allows a story to have many accessible points of view.”[12] Kay and his colleagues also added various other properties to the computer simulation of paper documents. As Kay has referred to this in another article, his idea was not to simply imitate paper but rather to create “magical paper.”[13] For instance, PARC team gave users the ability to modify the fonts in a document

[11] Ibid., 399.

[12] Ibid., 395. Emphasis mine – L.M.

[13] Alan Kay, “User Interface: A Personal View,” p. 199.

and create new fonts. They also implemented another important idea that was already developed by Douglas Englebardt's team in the 1960s: the ability to create different views of the same structure (I will discuss this in more detail below). And both Englebart and Ted Nelson also already "added" something else: the ability to connect different documents or different parts of the same document through hyperlinking – i.e. what we now know as hypertext and hypermedia. Englebart's group also developed the ability for multiple users to collaborate on the same document. This list goes on and on: e-mail in 1965, newsgroups in 1979, World Wide Web in 1991, etc.

Each of these new properties has far-reaching consequences. Take search, for instance. Although the ability to search through a page-long text document does not sound like a very radical innovation, as the document gets longer this ability becomes more and more important. It becomes absolutely crucial if we have a very large collection of documents – such as all the web pages on the Web. Although current search engines are far from being perfect and new technologies will continue to evolve, imagine how different the culture of the Web would be without them.

Or take the capacity to collaborate on the same document(s) by a number of users connected to the same network. While it was already widely used by companies in the 1980s and 1990s, it was not until early 2000s that the larger public saw the real cultural potential of this "addition" to print media. By harvesting the small amounts of labor and expertise contributed by a large number of volunteers, social software projects – most famously, Wikipedia – created vast and dynamically updatable pools of knowledge which would be impossible to create in traditional ways. (In a less visible way, every time we do a search on the Web and then click on some of the results, we also contribute to a knowledge set used by everybody else. In deciding in which sequence to present the results of a particular search, Google's algorithms take into account which among the results of previous searches for the same words people found most useful.)

Studying the writings and public presentations of the people who invented interactive media computing – Sutherland, Englebart, and Kay – makes it clear that they did not come with new properties of computational media as an after-thought. On the contrary, they knew that were turning physical media into new media. In 1968 Englebart gave his famous demo at the Fall Joint Computer Conference in San Francisco before few thousand people that included computer scientists, IBM engineers, people from other companies involved in computers, and funding officers from various government agencies.[14] Although Englebart had ninety minutes, he had a lot to show. Over the few previous years, his team at The Research Center for Augmenting Human Intellect had essentially developed modern *office* environment as it exists today (not be confused with modern *media design* environment which was developed later at PARC). Their computer system included word processing with outlining features, documents connected through hypertext, online collaboration (two people at remote locations working on the same document in real-time), online user manuals, online project planning system, and other elements of what is now called "computer-supported collaborative work." The team also developed the key elements of modern user interface that were later refined at PARC: a mouse and multiple windows.

Paying attention to the sequence of the demo reveals that while Englebart had to make sure that his audience would be able to relate the new computer system to what they already know and use, his focus was on new features of simulated media never before available previously. Englebart devotes the first segment of the demo to word processing, but as soon as he briefly demonstrated text entry, cut, paste, insert, naming and saving files – in other words, the set of tools which

[14] M. Mitchell Waldrop, *The Dream Machine: J.C.R. Liicklider and the Revolution That Made Computing Personal* (Viking, 2001), p. 287.

make a computer into a more versatile typewriter – he then goes on to show in more length the features of his system which no writing medium had before: "view control."[15] As Englebart points out, the new writing medium could switch at user's wish between *many different views of the same information*. A text file could be sorted in diffirent ways. It could also be organised as a hierarchy with a number of levels, like in outline processors or outlining mode of contemporary word processors such as Microsoft Word. For example, a list of items can be organised by categories and individual categories can be collapsed and expanded.

Englebart next shows another example of view control, which today, forty years after his demo, is still not available in popular document management software. He makes a long "to do" list and organizes it by locations. He then instructs the computer to displays these locations as a visual graph (a set of points connected by lines.) In front of our eyes, representation in one medium changes into another medium – text becomes a graph. But this is not all. The user can control this graph to display different amounts of information – something that no image in physical media can do. As Englebart clicks on different points in a graph corresponding to particular locations, the graph shows the appropriate part of his "to do" list. (This ability to interactively change how much and what information an image shows is particularly important in today's information visualization applications.)

Next Englebart presents "a chain of views" which he prepared beforehand. He switches between these views using "links" which may look like hyperlinks the way they exist on the Web today – but they actually have a different function. Instead of creating a path between many different documents a la Vannevar Bush's Memex (often seen as the precursor to modern hypertext), Englebart is using links as a method for switching between different views of a single

[15] Complete video of Engelbardt's 1968 demo is available at http://sloan.stanford.edu/MouseSite/1968Demo.html.

document organized hierarchically. He brings a line of words displayed in the upper part of the screen; when he clicks on these words, more detailed information is displayed in the lower part of the screen. This information can in its turn contain links to other views that show even more detail.

Rather than using links to drift through the textual universe associatively and "horizontally," we move "vertically" between more general and more detailed information. Appropriately, in Englebart's paradigm, we are not "navigating" – we are "switching views." We can create many different views of the same information and switch between these views in different ways. And this is what Englebart systematically explains in this first part of his demo. He demonstrates that you can change views by issuing commands, by typing numbers that correspond to different parts of a hierarchy, by clicking on parts of a picture, or on links in the text.

Since new media theory and criticism emerged in the early 1990s, endless texts have been written about interactivity, hypertext, virtual space, cyberspace, cyborgs, and so on. But I have never seen anybody discuss "view control." And yet this is one of the most fundamental and radical new techniques for working with information and media available to us today. It is used daily by each of us numerous times. "View control," i.e. the abilities to switch between many different views and kinds of views of the same information is now implemented in multiple ways not only in word processors and email clients, but also in all "media processors" (i.e. media editing software): AutoCAD, Maya, After Effects, Final Cut, Photoshop, inDesign, and so on. For instance, in the case of 3D software, it can usually display the model in at least half a dozen different ways: in wireframe, fully rendered, etc. In the case of animation and visual effects software, since a typical project may contain dozens of separate objects each having dozens of parameters, it is often displayed in a way similar to how outline processors can show text. In other words, the user can switch between more and less information. You can choose to see only those parameters which you are working on right now. You can also zoom in and out of the composition. When you do this, parts of the composition do not simply get smaller or bigger – they show less or more information automatically. For instance, at a certain scale you may only see the names of different parameters; but when you zoom into the display, the program may also display the graphs which indicate how these parameters change over time.

As we can see from the examples we have analyzed, the aim of the inventors of computational media – Englebart, Nelson, Kay and people who worked with them – was not to simply create accurate simulations of physical media. Instead, in every case the goal was to create "a new medium with new properties" which would allow people to communicate, learn, and create in new ways. So while today the content of these new media may often look the same as with its predecessors, we should not be fooled by this similarity. The newness lies not in the content but in software tools used to create, edit, view, distribute and share this content. Therefore, rather than only looking at the "output" of software-based cultural practices, we need to consider software itself – since it allows people to work with media in of a number of historically unprecedented ways. So while on the level of appearance computational media indeed often remediate (i.e. represents) previous media, the software environment in which this media "lives" is very different.

Let me add to the examples above one more – Ivan Sutherland's *Sketchpad* (1962). Created by Sutherland as a part of his PhD thesis at MIT, Sketchpad deeply influenced all subsequent work in computational media (including that of Kay) not only because it was the first interactive media authoring program but also because it made it clear that computer simulations of physical media can add many exiting new properties to media being simulated. Sketchpad was the first software that allowed its users to interactively create and modify line drawings. As Noah Wardrip-Fruin points out, it "moved beyond paper by allowing the user to work at any of 2000 levels of magnification – enabling the creation of projects that, in physical media, would either be unwieldy large or require detail work at an impractically small size."[16] Sketchpad similarly redefined graphical elements of a design as objects which "can be manipulated, constrained, instantiated, represented ironically, copied, and recursively operated upon, even recursively merged.'[17] For instance, if the designer defined new graphical elements as instances of a master element and later made a change to the master, all these instances would also change automatically.

Another new property which perhaps demonstrated most dramatically how computer-aided drafting and drawing were diffirent from their physical counterparts was Sketchpad's use of constraints. In Sutherland's own words, "The major feature which distinguishes a Sketchpad drawing from a paper and pencil drawing is the user's ability to specify to Sketchpad mathematical conditions on already drawn parts of his drawing which will be automatically satisfied by the computer to make the drawing take the exact shape desired."[18] For instance, if a user drew a few lines, and then gave the appopriate command, Sketchpad automatically moved these lines until they were parallel to each other. If a user gave a diffirent command and selected a particular line, Sketchpad moved the lines in such a way so they would parallel to each other and perpendicular to the selected line.

Although we have not exhausted the list of new properties that Sutherland built into Sketchpad, it should be clear that this first interactive graphical editor was not only simulating existing media. Appropriately, Sutherland's 1963 paper on Sketchpad repeatedly emphasizes the new graphical capacities of his system, marveling how it opens new fields of "graphical manipulation that has never been

[16] Noah Wardrip-Fruin, introduction to "Sketchpad. A Man-Machine Graphical Communication System," in *New Media Reader*, 109.

[17] Ibid.

[18] Ivan Sutherland, "Sketchpad. A Man-Machine Graphical Communication System" (1963), in *New Media Reader*, eds. Noah Wardrip-Fruin and Nick Montfort.

available before."[19] The very title given by Sutherland to his PhD thesis foregrounds the novelty of his work: *Sketchpad: A man-machine graphical communication system*. Rather than conceiving of Sketchpad as simply another media, Sutherland presents it as something else - a communication system between two entities: a human and an intelligent machine. Kay and Goldberg will later also foreground this communication dimension referring to it as "a two-way conversation" and calling the new "metamedium" "active."[20] (We can also think of Sketchpad as a practical demonstration of the idea of "man-machine symbiosis" by J.C. Licklider applied to image making and design.[21]

The Permanent Extendibility

As we saw, Sutherland, Nelson, Englebart, Kay and other pioneers of computational media have added many previously non-existent properties to media they have simulated in a computer. The subsequent generations of computer scientists, hackers, and designers added many more properties – but this process is far from finished. And there is no logical or material reason why it will ever be finished. It is the "nature" of computational media that it is open-ended and new techniques are continuously being invented.

To add new properties to physical media requires modifying its physical substance. But since computational media exists as software, we can add new properties or even invent new types of media by simply changing existing or writing new software. Or by adding plug-ins and extensions, as programmers have been doing it with Photoshop and Firefox, respectively. Or by putting existing software together. (For instance, at the moment of this writing – 2006 -

[19] Ibid., 123.

[20] Kay and Goldberg, "Personal Dynamic Media," 394.

[21] J.C. Licklider, "Man-Machine Symbiosis" (1960), in *New Media Reader*, eds. Noah Wardrip-Fruin and Nick Montfort.

people are daily extending capacities of mapping media by creating software mashups which combining the services and data provided by Goggle Maps, Flickr, Amazon, other sites, and media uploaded by users.)

In short, *“new media” is “new” because new properties (i.e., new software techniques) can always be easily added to it.* Put differently, in industrial, i.e. mass-produced media technologies, “hardware” and “software” were one and the same thing. For example, the book pages were bound in a particular way that fixed the order of pages. The reader could not change nether this order nor the level of detail being displayed a la Englebart’s “view control.” Similarly, the film projector combined hardware and what we know call a “media player” software into a single machine. In the same way, the controls built into a twentiteh-century mass-produced camera could not be modified at user’s will. And although today the user of a digital camera similarly cannot easily modify the hardware of her camera, as soon as transfers the pictures into a computer she has access to endless number of controls and options for modifying her pictures via software.

In the nineteenth and twentieth century there were two types of situations when a normally fixed industrial media was more fluid. The first type of situation is when a new media was being first developed: for instance, the invention of photography in the 1820s-1840s. The second type of situation is when artists would systematically experiment with and “open up” already industrialized media – such as the experiments with film and video during the 1960s, which came to be called “Expanded Cinema.”

What used to be separate moments of experimentations with media during the industrial era became the norm in a software society. In other words, computer legitimizes experimentation with media. Why this is so? What differentiates a modern digital computer from any other machine – including industrial media machines for capturing and playing media – is separation of hardware and software. It is because endless number of different programs performing different tasks can be written to run on the same typemachine, this machine – i.e. a digital computer - is used so widely today. Consequently, the constant invention of new and modification of existing media software is simply one example of this general principle. In other words, experimentation is a default feature of computational media. In its very structure it is "avant-garde" since it is constantly being extended and thus redefined.

If in modern culture "experimental" and "avant-garde" were opposed to normalized and stable, this opposition largely disappears in software culture. And the role of the media avant-garde is performed no longer by individual artists in their studios but by a variety of players, from very big to very small - from companies such as Microsoft, Adobe, and Apple to independent programmers, hackers, and designers.

But this process of continual invention of new algorithms does not just move in any direction. If we look at contemporary media software – CAD, computer drawing and painting, image editing, word processors – we will see that most of their fundamental principles were already developed by the generation of Sutherland and Kay. In fact the very first interactive graphical editor – Sketchpad – already contains most of the genes, so to speak, of contemporary graphics applications. As new techniques continue to be invented they are layered over the foundations that were gradually put in place by Sutherland, Englebart, Kay and others in the 1960s and 1970s.

Of course we not dealing here only with the history of ideas. Various social and economic factors – such as the dominance of the media software market by a handful of companies or the wide adoption of particular file formats –– also constrain possible directions of software evolution. Put differently, today software development is an industry and as such it is constantly balances between stability and innovation, standardization and exploration of new possibilities. But it is not just any industry. New programs can be written and existing programs can be extended and modified (if the source code is available) by anybody who has programming skills and access to a computer, a programming language and a compiler. In other words, today software is fundamentally “fabbable” in a way that physical industrially produced objects usually are not.

Although Turing and Von Neumann already formulated this fundamental extendibility of software in theory, its contemporary practice – hundreds of thousands of people daily involved in extending the capabilities of computational media - is a result of a long historical development. This development took us from the few early room-size computers which were not easy to reprogram to a wide availability of cheap computers and programming tools decades later. This democratization of software development was at the core of Kay’s vision. Kay was particularly concerned with how to structure programming tools in such a way that would make development of media software possible for ordinary users. For instance, at the end of the 1977 article I have been already extensively quoting, he and Goldberg write: “We must also provide enough already-written general tools so that a user need not start from scratch for most things she or he may wish to do.”

Comparing the process of continuous media innovation via new software to history of earlier, pre-computational media reveals a new logic at work. According to a commonplace idea, when a new medium is invented, it first closely imitates already existing media before discovering its own language and aesthetics. Indeed, first printed bibles by Guttenberg closely imitated the look of the handwritten manuscripts; early films produced in the 1890s and 1900s mimicked the presentational format of theatre by positioning the actors on the invisible shallow stage and having them face the audience. Slowly printed books developed a different way of presenting information; similarly cinema also developed its own original concept of narrative space. Through repetitive shifts in points of view presented in subsequent shots, the viewers were placed inside this space – thus literally finding themselves inside the story.

Can this logic apply to the history of computer media? As theorized by Turing and Von Neuman, computer is a general-purpose simulation machine. This is its uniqueness and its difference from all other machines and previous media. This means that the idea that a new medium gradually finds its own language cannot apply to computer media. If this was true it would go against the very definition of a modern digital computer. This theoretical argument is supported by practice. The history of computer media so far has been not about arriving at some standardized language – the way this, for instance, happened with cinema – but rather about the gradual expansion of uses, techniques, and possibilities. Rather than arriving at a particular language, we are grdually discovering that the computer can speak more and more languages.

If we are to look more closely at the early history of computer media – for instance, the way we have been looking at Kay's ideas and work in this text – we will discover another reason why the idea of a new medium gradually discovering its own language does not apply to computer media. The systematic practical work on making a computer simulate and extend existing media (Sutherland's Sketchpad, first interactive word processor developed by Englebart's group, etc.) came after computers were already put to multiple uses – performing diffirent types of calculations, solving mathematical problems, controlling other machines in real time, running mathematical simulations, simulating some aspects of human intelligence, and so on. (We should also mention the work on SAGE by MIT Lincoln Laboratory which by the middle of the 1950s already established the idea of interactive communication between a human and a computer via a screen with a graphical display and a pointing device. In fact, Sutherland developed Sketchpad on TX-2 that was the new version of a larger computer MIT constructed for SAGE.) Therefore, when the generation of Sutherland, Nelson and Kay started to create "new media," they built it on top, so to speak, of what computers were already known to be capable off. Consequently they added new properties into physical media they were simulating right away. This can be very clearly seen in the case of Sketchpad. Understanding that one of the roles a computer can play is that of a problem solver, Sutherland built in a powerful new feature that never before existed in a graphical medium – satisfaction of constraints. To rephrase this example in more general terms, we can say that rather than moving from an imitation of older media to finding its own language, computational media was from the very beginning speaking a new language.

In other words, the pioneers of computational media did not have the goal of making the computer into a ‘remediation machine” which would simply represent older media in new ways. Instead, knowing well new capabilities provided by digital computers, they set out to create fundamentally new kinds of media for expression and communication. These new media would use as their raw “content” the older media which already served humans well for hundreds and thousands of years – written language, sound, line drawings and design plans, and continuous tone images, i.e. paintings and photographs. But this does not compomise the newness of new media. For computational media uses these traditional human media simply as building blocks to create previously unimaginable representational structures, creative and thinking tools, and communication options.

Although Sutherland, Engelbart, Nelson, Kay, and others developed computational media on top of already existing developments in computational theory, programming languages, and computer engineering, it will be incorrect to conceive the history of such influences as only going in one direction – from already existing and more general computing principles to particular techniques of computational media. The inventors of computational media had to question many, if not most, already established ideas about computing. They have defined many new fundamental concepts and techniques of how both software and hardware thus making important contributions to hardware and software engineering. A good example is Kay’s development of Smalltalk which for the first time systematically establsihed a paradigm of object-oriented programming.

Kay's rationale to develop this programming langauge was to give a unified apperance to all applications and the interface of PARC system and, even more importantly, to enable its users to quickly program their own media tools. (According to Kay, an object-oriented illustration program written in Smalltalk by a particularly talented twelth-year old girl was only a page long.[22]) Subsequently object-oriented programming paradigm became very popular and object-oriented features have been added to most popular languages such as C++.

Looking at the history of computer media and examining the thinking of its inventors makes it clear that we are dealing with the opposite of technological determinism. When Sutherland designed Sketchpad, Nelson conceived hypertext, Kay programmed a paint program, and so on, each new property of computer media had to be imagined, implemented, tested, and refined. In other words, these characteristics did not simply come as an inevitable result of a meeting between digital computers and modern media. Computational media had to be invented, step-by-step. And it was invented by people who were looking for inspiration in modern art, literature, cognitive and education psychology, and theory of media as much as technology. For example, Kay recalls that reading McLuhan's *Understanding Media* led him to a realization that computer can be a medium rather than only a tool.[23]

So far I have talked about the history of computational media as series of consecutive "additions." However this history is not only a a process of accumulation of more and more options. Although in general we have more techniques at our disposal today when twenty of thirty years ago, it is also important to remember that many fundamentally new techniques which were conceived were never given commercial implementation. Or they were poorly implemented and did not become popular. Or they were not marketed properly.

---

[22] Alan Kay, *Doing with Images Makes Symbols* (University Video Communications, 1987), videotaped lecture (available at www.archive.org).

[23] Alan Kay, "User Interface: A Personal View," 192-193.

Sometimes the company making the software would go out of business. At other times the company that created the software was purchased by another company that "shelved" the software so it would not compete with its own products. And so on. In short, the reasons why many of new techniques did not become common place are multiple, and are not reducible to a single principle such as "the most easy to use techniques become most popular."

For instance, one of the ideas developed at PARC was "project views." Each view "holds all the tools and materials for a particular project and is automatically suspended when you leave it."[24] Although currently (2006) there are some applications that implement this idea, it is not a part of most popular operating systems: Windows, MAC OSX, and Lunix. The same holds true for the contemporary World Wide Web implementation of hyperlinks. The links on the Web are static and one-directional. Ted Nelson who is credited with inventing hypertext around 1964 conceived it from the beginning to have a variety of other link types. In fact, when Tim Berners-Lee submitted his paper about the Web to ACM Hypertext 1991 conference, his paper was only accepted for a poster session rather than the main conference program. The reviewers saw his system as being inferior to many other hypertext systems that were already developed in academic world over previous two decades.[25]

Computer as Metamedium

As we have established, the development of computational media runs contrary to previous media history. But in a certain sense, the idea of a new media gradually discovering its own language actually does apply to the history of computational media after all. And just as it was the case with printed books and

---

[24] Aan Kay, "User Interface: A Personal View," p. 200.

[25] Noah Wardrip-Fruin and Nick Monford, Introduction to Tim Berners-Lee et al., "The World-Wide Web" (1994), reprinted in *New Media Reader.*

cinema, this process took a few decades. When first computers were built in the middle of the 1940s, they could not be used as media for cultural representation, expression and communication. Slowly, through the work of Sutherland, Englebart, Nelson, Papert and others in the 1960s, the ideas and techniques were developed which made computers into a "cultural machine." One could create and edit text, made drawings, move around a virtual object, etc. And finally, when Kay and his colleagues at PARC systematized and refined these techniques and put them under the umbrella of GUI which made computers accessible to multitudes, a digital computer finally was given its own language - in cultural terms. In short, only when a computer became a cultural medium – rather than only a versatile machine.

Or rather, it became something which no other media has been before. For what has emerged was not yet another media, but, as Kay and Goldberg insist in their article, something qualitatively different and historically unprecedented. To mark this difference, they introduce a new term – "metamedium."

This metamedium is unique in a number of different ways. One of them we already discussed in detail – it could represent most other media while augmenting them with many new properties. Kay and Goldberg also name other properties which are equally crucial. The new metamedium is "active – it can respond to queries and experiments – so that the messages may involve the learner in a two way conversation." For Kay who was strongly interested in children and learning, this property was particularly important since, as he puts it, it "has never been available before except through the medium of an individual teacher." [26] Further, the new metamedium can handle "virtually all of its owner's information-related needs." (I have already discussed the consequence of this property above.) It can also "serve as "a programming and problem solving tool,"

[26] Ibid., 394.

and “an interactive memory for the storage and manipulation of data.”[27] But the property which is the most important from the point of view of media history is that *computer metamedium is simultaneously a set of different media and a system for generating new media tools and new types of media.* In other words, a computer can be used to create *new tools for working in the media it already provides as well as to develop new not-yet-invented media*.

Using the analogy with print literacy, Kay’s motivates this property in this way: “The ability to ‘read’ a medium means you can *access* materials and tools generated by others. The ability to write in a medium means you can *generate* materials and tools for others. You must have both to be literate.”[28] Accordingly, Kay’s key effort at PARC was the development of Smalltalk programming language. All media editing applications and GUI itself were written in Smalltalk. This made all the interfaces of all applications consistent facilitating quick learning of new programs. Even more importantly, according to Kay’s vision, Smalltalk language would allow even the beginning users write their own tools and define their own media. In other words, all media editing applications, which would be provided with a computer, were to serve also as examples inspiring users to modify them and to write their own applications.

Accordingly, the large part of Kay and Goldberg’s paper is devoted to description of software developed by the users of their system: “an animation system programmed by animators”; “a drawing and painting system programmed by a child,” “a hospital simulation programmed by a decision-theorist,” “an audio animation system programmed by musicians”; “a musical score capture system programmed by a musician”; “electronic circuit design by a high school student.” As can be seen from this list that corresponds to the sequence of examples in

---

[27] Ibid., 393.

[28] Alan Kay, “User Interface: A Personal View,” p. 193. in The Art of Human-Computer Interface Design, 191-207. Editor Brenda Laurel. Reading, Mass,” Addison-Wesley, 1990. The emphasis is in the original.

the article, Kay and Goldberg deliberately juxtapose different types of users - professionals, high school students, and children – in order to show that everybody can develop new tools using Smalltalk programming environment.

The sequence of examples also strategically juxtaposes media simulations with other kinds of simulations in order to emphasize that simulation of media is only a particular case of computer's general ability to simulate all kinds of processes and systems. This juxtaposition of examples gives us an interesting way to think about computational media. Just as a scientist may use simulation to test different conditions and play different what/if scenarios, a designer, writer, a musician, a filmmaker, or an architect working with computer media can quickly "test" different creative directions in which the project can be developed as well as see how modifications of various "parameters" affect the project. The later is particularly easy today since the interfaces of most media editing software not only explicitly present these parameters but also simultaneously give the user the controls for their modification. For instance, when the Formatting Palette in Microsoft Word shows the font used by the currently selected text, it is displayed in column next to all other fonts available. Trying different font is as easy as scrolling down and selecting the name of a new font.

To give users the ability to write their own programs was a crucial part of Kay's vision for the new "metamedium" he was inventing at PARC. According to Noah Wardrip-Fruin, Englebart research program was focused on a similar goal: "Englebart envisioned users creating tools, sharing tools, and altering the tools of others."[29] Unfortunately, when in 1984 Apple shipped Macintosh, which was to become the first commercially successful personal computer modeled after PARC system, it did not have easy-to-use programming environment. HyperCard written for Macintosh in 1987 by Bill Atkinson (who was one of PARC alumni) gave users the ability to quickly create certain kinds of applications – but it did

[29] Noah Wardrip-Fruin, introduction to Douglas Engelbart and William English, "A Research Center for Augmenting Human Intellect" (168), *New Media Reader,* 232.

not have the versatility and breadth envisioned by Kay. Only recently, as the general computer literacy has widen and many scripting languages became available – Perl, PHP, Python, ActionScript, Vbscript, JavaScript, etc. – more people started to create their own tools by writing software. A good example of a contemporary programming environment, which is currently very popular among artists and designers and which, in my view, is close to Kay's vision is Processing.[30] Build on top of Java programming language, Processing features a simplified programming style and an extensive library of graphical and media functions. It can be used to develop complex media programs and also to quickly test ideas. Appropriately, the official name for Processing projects is sketches.[31] In the words of Processing initiators and main developers Ben Fry and Casey Reas, the language's focus "on the 'process' of creation rather than end results."[32] Another popular programming environment that similarly enables quick development of media software is MAX/MSP and its successor PD developed by Miller Puckette.

Conclusion

The story I just told could also be told diffirently. It is possible to put Sutherland' work on Sketchpad in the center of computational media history; or Englebart and his Research Center for Augmenting Human Intellect which throughout the 1960s developed hypertext (independently of Nelson), the mouse, the window, the word processor, mixed text/graphics displays, and a number of other "firsts." Or we can shift focus to the work of Architecture Machine Group at The MIT which since 1967 was headed by Nicholas Negroponte (In 1985 this group became The Media Lab). We also need to recall that by the time Kay's Learning Research Group at PARC flashed out the details of GUI and programmed

[30] www.processing.org, accessed January 20, 2005.

[31] http://www.processing.org/reference/environment/.

[32] http://processing.org/faq/,

various media editors in Smalltalk (a paint program, an illustration program, an animation program, etc.), artists, filmmakers and architects were already using computers for more than a decade and a number of large-scale exhibitions of computer art were put in major museums around the world such as the Institute of Contemporary Art, London, The Jewish Museum, New York, and Los Angeles County Museum of Art. And certainly, in terms of advancing computer techniques for visual representation enabled by computers, other groups of computer scientists were already ahead. For instance, at University of Utah, which became the main place for computer graphics research during the first part of the 1970s, scientists were producing 3D computer graphics much superior to the simple images that could be created on computers being build at PARC. Next to University of Utah a company called Evans and Sutherland (headed by the same Ivan Sutherland who was also teaching at University of Utah) was already using 3D graphics for flight simulators – essentially pioneering the type of new media that can be called "navigable 3D virtual space."[33]

The reason I decided to focus on Kay is his theoretical formulations that place computers in relation to other media and media history. While Vannevar Bush, J.C. Lindlicker and Douglas Englebart were primary concerned with augmentation of intellectual and in particular scientific work, Kay was equally interested in computers as "a medium of expression through drawing, painting, animating pictures, and composing and generating music."[34] Therefore if we really want to understand how and why computers were redefined as a cultural media, and how the new computational media is different from earlier physical media, I think that Kay provides us with the best perspective. At the end of the 1977 article that served as the basis for our discussion, he and Goldberg summarize their arguments in the phrase, which in my view is a best formulation we have so far of what computational media is artistically and culturally. They call

[33] For more on 3D virtual navigable space as a new media, or a "new cultural form," see chapter "Navigable Space" in *The Language of New Media*.

[34] Ibid., 393.

computer *“a metamedium” whose content is “a wide range of already-existing and not-yet-invented media.”* In another article published in 1984 Kay unfolds this definition. As a way of conlusion, I would like to quote this longer definition which is as accurate and inspiring today as it was when Kay wrote it:

It [a computer] is a medium that can dynamically simulate the details of any other medium, including media that cannot exist physically. It is not a tool, though it can act like many tools. It is the first *metamedium*, and as such it has degrees of freedom for representation and expression never before encountered and as yet barely investigated.[35]

[35] Alan Kay, “Computer Software,” *Scientific American* (September 1984), 52. Quoted in Jean-Louis Gassee with Howard Rheingold, “The Evolution of Thinking Tools,” in T*he Art of Human-Computer Interface Design*, p. 225.