Introduction

In 1963, MIT graduate student Ivan Sutherland submitted his PhD dissertation, titled *Sketchpad: a man-machine graphical communication system*, describing the computer program he wrote—Sketchpad. It is a program with which users can use a light pen to draw geometrical shapes directly on the computer screen with real-time feedback. The user can change the shapes, move them around, and resize them, pretty much like the modern tablet that we use today.

It was revolutionary for a few reasons. Firstly, its interactivity was considered to have pioneered the way for human-computer interaction (HCI).[1] Back in the 50s and early 60s, computers were room-size, clumsy, and expensive. Programmers had to share the computing power by batch processing. They formulated their instructions by punching patterns of holes on reels of paper or decks of cards and then submitted these to the operator of the computer. The operator combined all the programs in one tape and ran it through on the computer, like "a group of people catching a bus."[2] After the batch was run, the programmers came back in the afternoon or the next day to pick up the results, usually printed on piles of paper,[3] later photographed onto ordinary 35-millimeter film.[4] The direct interaction with computer that we have today, however tempting, was beyond imagination. So, there was not much to be done about a friendlier user interface, except for experimental machines at research institutes.

Secondly, Sketchpad also served as one of the earliest demonstrations of graphical user interface (GUI), through which users can interact with computers by clicking graphical icons or directly creating and manipulating graphical elements. GUI is so pervasive and important that as long as you are

[1] For more details, see Jonathan Grudin, ed., *From Tool to Partner: The Evolution of Human-Computer Interaction*, Synthesis Lectures on Human-Centered Interactions, #35 (San Rafael, California: Morgan & Claypool Publishers, 2017).
[2] "A Solution to Computer Bottlenecks," *MIT Science Reporter* (WGBH, 1963), https://www.youtube.com/watch?v=FTcLzZOQTvk&t=3s.
[3] Paul E. Ceruzzi, *A History of Modern Computing* (MIT Press, 2003). 74
[4] Robert Rivlin, *The Algorithmic Image: Graphic Visions of the Computer Age* (Redmond, Washington 980073-9717: Microsoft Press, 1986). 17.

using a laptop, a smartphone, or a tablet today, then you are using a GUI. As mentioned earlier, computer instructions or programs were fed into computers by decks of punched cards in the early 1950s through the 1970s. These punched cards represents what Dourish (2004) calls the “symbolic” phase of interaction development, and are quite advanced compared with the “electric” phase, where computing circuits need to be reconfigured for new tasks.[5] In the late 50s, programming languages that somewhat resembled English, such as Fortran (Formula Translation) and APT (Automatically Programmed Tools), were invented, entering the “textual” phase[6], so that programmers can code with statements such as “IF” “READ TAPE” “END FILE” etc., which made programming a lot easier. However, graphical information still proved to be a problem. Even in APT, specially designed for a numerically controlled milling machine and simplified for graphical information, the programmers still had to calculate the coordinates of points and other geometrical features. Sketchpad eluded all these problems by enabling users to directly draw on the screen, featuring probably the most significant transition into the graphical phase of human-computer interaction.

Thirdly, the direct input and manipulation of graphical information marked Sketchpad as the ancestor of modern computer-aided design (CAD), which itself was a novel and not yet articulated concept in the early 1960s. With CAD, designers did not need to deal with specific coordinates as they would via other means of designing, such as APT. They only needed to sketch out a rough idea. Sketchpad would make lines perpendicular or parallel, once the designer selected the lines in question and specified the constraints. In this way, Sketchpad freed designers from the complexities of geometry and mathematics. Another idea was the “master drawing”, where the user could save a sketch in the library as the “master drawing”, copy it multiple times, and arrange them as he or she wanted, making repetitive drawings, such as electronic engineering, extremely effortless. These ideas exemplified in

[5] Paul Dourish, *Where the Action Is: The Foundations of Embodied Interaction* (MIT Press, 2004). 6-7.
[6] Dourish. 9.

Sketchpad were so far ahead of their time that Rivlin commented in the late 1980s, "except for the addition of color and changes in a few minor details concerning how the graphics processing is accomplished, the 1963 version of Sketchpad has remained virtually unchanged in 95 percent of the graphics programs available today, including those that run on home computers."[7]

Sketchpad's inventor Ivan Sutherland won the 1988 A.M. Turing Award from Association for Computing Machinery, the Nobel Prize for computer scientists, "for this pioneering and visionary contributions to computer graphics, starting with Sketchpad, and continuing after."[8] I was surprised how little history has been written by professional historians because Sketchpad was clearly a milestone and its history sits at the origins of human-computer interaction, graphical user interface, and computer-aided design. The influence of that period is profound. Just as Burton writes in Sutherland's Turing Award Laureate introduction, Sketchpad's "graceful interaction and functionality continue to inspire admiration among computer graphic professionals."[9]

The history of Sketchpad also sits at the intersection of social environment, technological developments, and philosophical thought. It is by no means simple or straightforward, like other apparently comparable software products. Nathan Ensmenger describes how difficult it is to define "software" in history because it is "inextricably intertwined with the larger sociotechnical system of computing that includes machines (computers and their associated peripherals), people (users, designers, and developers), and processes (the corporate payroll system, for example)."[10] In this piece, besides a social history arranged chronologically about who did what where, I also delineate the intellectual background of the time when scholars began to ponder what exactly man-machine

---

[7] Rivlin, *The Algorithmic Image*. 19.
[8] Robert Burton, "Ivan Sutherland - A.M. Turing Award Laureate," A.M. Turing Award, accessed March 14, 2019, https://amturing.acm.org/award_winners/sutherland_3467412.cfm.
[9] Burton.
[10] Nathan L. Ensmenger, *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise* (MIT Press, 2012). 7.

communication is, and identify Sutherland's professors at MIT, Douglas T. Ross and Steven Coons, as the primary influences on his philosophy of designing.

As a historian, one has to be extremely careful when speaking of "the first X" especially when X was not yet clearly defined, because such an expression risks anachronism. Concepts are constantly shifting and changing in the longer course of history. As Mahoney puts it nicely, "To the historian, the old way is crucial: it holds the roots of the new way. It does not illuminate history to say, 'We were really doing X,' where X stands for the current state of the art. Talking that way masks the very changes in conceptual structure that explain the development of X and that history aims at elucidating."[11] My work will help to elucidate the characteristics of some now-familiar terms in the 1960s, such as GUI, CAD, and HCI, how they were different from what they are right now, as well as how they were envisioned by the pioneering computer scientists. "Technological systems are themselves embedded in a set of social and cultural practices that give them meaning at the same time as being constrained and transformed by them,"[12] writes Dourish. It reminds us, these terms are moving targets and what they used to be has shaped the world we have today.

*** Assuming what you are reading right now is chpt 1 *** Chapter 2 is a narrative of what happened in the late 1950s and early 60s. It is a social history of people and organizations. Ivan Sutherland designed and implemented Sketchpad in MIT's Lincoln Laboratory, a top-tier laboratory of information science. Lincoln Laboratory was founded as a "Laboratory for Air Defense" in the face of the tensions arising during the early years of the Cold War. In completing its assignment of integrating the radars to an operational system in the development of Semi-Automatic Ground Environment (SAGE) system, the Lincoln Laboratory has contributed various inventions for Information Technology, such as

[11] Michael S. Mahoney, "History of Programming Languages—II: What Makes History?," ed. Thomas J. Bergin Jr. and Richard G. Gibson Jr. (New York, NY, USA: ACM, 1996), 831–832, https://doi.org/10.1145/234286.1057848. 832.
[12] Dourish, *Where the Action Is*. 97.

magnetic-core storage, light-pen input, interactive graphic displays, etc.[13] These innovations helped construct the first modern computer system. Ivan Sutherland came there in 1960 in pursuit of a doctorate. The electronic devices available to him, such as the computer TX-2 and the light pen, together with the opportunity to work with top-notch computer scientists and mathematicians such as Douglas T. Ross and Steven Coons, provided him with unique soil where his ideas could grow and be made into practice. Besides the discussion about Lincoln Lab, this chapter also talks about some of the similar efforts such as EDM (Electronic Drafting Machine, developed by Itek Corporation, Lexington, MA). As I will explain, these simultaneous efforts also have their roots in Lincoln Lab. The last part of this chapter will briefly introduce some later projects which drew inspirations from Sutherland's Sketchpad, such as Alan Kay's Dynabook and the Star workstation by Xerox Corporation, and try to explain why Sutherland's Sketchpad was never commercialized.

***Look for the emphasis-on-interactivity-environment in Waldrop 2001's Licklider book, around p 180 ***

In chapter 3, I will break down Sketchpad into easily understandable explanations based on Sutherland's original dissertation. This break-down is important because an in-depth analysis especially of data structure and algorithms will enable me to compare Sutherland's thoughts with Ross's and Coons's philosophies in chapter 4. Although Sketchpad is probably more influential as something "soft" and abstract, like a set of instructions that enables computer to perform some user-friendly tasks, Sketchpad is never only "soft." Thus, a useful dichotomy here is hardware vs. software (or implementation vs. abstraction[14]). I will talk about the cathode-ray tube (CRT) display[15] as the "magic

[13] Alan A. Grometstein, ed., *MIT Lincoln Laboratory: Technology in Support of National Security* (Lexington, Mass.: MIT Lincoln Laboratory, 2011). 33.
[14] Dourish, *Where the Action Is*. 140
[15] Cathode-ray tube (CRT) is a vacuum tube that contains one or more electron guns and a phosphorescent screen, used to display images. Basically, each of these guns streams a steady flow of electrons, left to right, top to bottom

slate", the light pen as the "magic wand", and the super-power transistorized computer TX-2, the computing power of which made everything possible. I will also talk about the data structure—ring structure, featuring rapid processing of topological information with no searching at all and how the programming techniques that facilitate Sketchpad serve as a form of object-oriented programming (OOP), a phrase which was not yet coined in 1963.

Chapter 4 focuses on Ivan Sutherland's intellectual Influences including J.C.R. Licklider, Douglas T. Ross and Steven Coons. As a broader background, engineers since the 1940s began to wonder what computers could do for humans. For example, Vannevar Bush's 1945 essay "As We May Think" describes his envisions of how computer might be helpful in information processing such as xxxxx,[16] which inspired xxxx. In the same vein, J. C. R. Licklider published a very influential article titled "Man-Computer Symbiosis" in 1960, which provides a general research agenda of computers moving beyond functioning as simple calculating devices to functioning in filing, organizing, retrieving information, and aiding human decision-making. To achieve this goal, he identifies five areas of computer technology, including input/output equipment, emphasizing the ability to draw graphs on the monitor as an input of functions.[17] These men's visions of what computers should be able to do shaped the information technology for the following decades, leading to the research in rapid information processing, exchange, and retrieval, such as time-sharing computers[18] and ARPANET[19]. In Licklider's 1965 book *Libraries of the Future*, he also envisions man-computer communication by writing and drawing "on a surface with a stylus" with immediate response as effective "future systems for interaction with the body of

---

of the monitor or screen. When the electrons hit the phosphors on the CRT, the phosphors will glow for a short period of time. These guns repeat this process multiple times per second, displaying images stable to human eyes

[16] Vannevar Bush, "As We May Think," The Atlantic, July 1, 1945, https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/.

[17] J. C. R. Licklider, "Man-Computer Symbiosis," *IRE Transactions on Human Factors in Electronics* HFE-1, no. 1 (March 1960): 4–11, https://doi.org/10.1109/THFE2.1960.4503259.

[18] Multiple users access the same computer via different consoles or electric type-writers

[19] Advanced Research Projects Agency Network was the ancestor of Internet.

knowledge".[20] Thus, Sutherland's framing Sketchpad as a man-machine communication system can be seen as a response to Licklider's envisioning of man-computer symbiosis.

More specifically Sutherland worked closely with faculty from the Computer-Aided Design (CAD) Group at MIT (later a part of MAC project[21]), co-led by Douglas T. Ross and Steven Coons. Their vision of CAD—a novel term then—was to simplify designing and manufacturing and to remove all the tedious and monotonous part of the draftsman job while maintaining flexibility to allow room for creativity.[22] However, Ross and Coons had different foci in implementing CAD.

Douglas Ross invented the Automatic Programmed Tool (APT) for numerically controlled machines[23] in the late 1950s and his philosophy of programing involved inventing a universal language that can represent everything in the world, including physical objects, relationships, etc. He wanted every word of human language to have a corresponding "objet", the basic element/unit/atom for representation, because he saw designing as problem-solving and believed that the verbal articulation or the statement of the problem is the first step in any kind of problem solving. Only by articulating our design intentions in an invented common language between us and computers, will computers be able

[20] J. C. R. Licklider, *Libraries of the Future* (The MIT Press, 1965).
[21] Explain.
[22] Steven Anson Coons, "An Outline of the Requirements for a Computer-Aided Design System," in *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference*, AFIPS '63 (Spring) (Spring Joint Computer Conference, 1963, New York, NY, USA: ACM, 1963), 299–304, https://doi.org/10.1145/1461551.1461588; Douglas Taylor Ross, "Computer-Aided Design: A Statement of Objectives," Technical, Technical Memorandum (Cambridge 39, Massachusetts: MIT Electronic Systems Laboratory, September 1960).
[23] In the early 1950s, MIT's Servomechanisms Laboratory developed the first automatically controlled milling machine. Simply put, the controlling information to instruct the milling machine was fed into the computer in the form of punched cards. That is to say, the manufacturing instruction on the design sketch was translated into numerical form represented by appropriate patterns of the holes on the punch cards. In the late 50s, MIT's Computer Applications Group of the Electronic Systems Laboratory developed an automatic system for preparing these punched cards from detail drawings. That is, Automatically Programmed Tool (APT). My understanding is that it is a high-level language that resembles natural language, designed specifically for numerically controlled milling machine, so that programmers do not need to prepare the machining instructions in detailed numerical form.

to “understand” our instructions.[24] His philosophy of “objet” has influenced Sutherland’s data structure of “ring structure”.

Steven Coons, who worked closely with architects and designers, understood that sketching served as the primary communication between designers, engineers, and manufacturers, not language. He saw graphical manipulation as the first step of design, and “the sketch forms the natural bridge between these vague stirrings of the imagination and the subsequent precise statement of the refined details of the concept.”[25]

The last chapter is the conclusion: What is X? *(I have no idea yet, but I imagine this chapter to be a philosophical argument, following the social history and intellectual in previous chapters, completing my integrated HPS wish.)*

a. Graphical interface
b. Man-machine communication
c. Human-computer interaction
d. Computer-aided design?

[24] Douglas T. Ross, “Gestalt Programming: A New Concept in Automatic Programming,” in *Papers Presented at the February 7-9, 1956, Joint ACM-AIEE-IRE Western Computer Conference*, AIEE-IRE ’56 (Western) (New York, NY, USA: ACM, 1956), 5–10, https://doi.org/10.1145/1455410.1455414.
[25] Coons, “An Outline of the Requirements for a Computer-Aided Design System.”