

Number 574



UNIVERSITY OF  
CAMBRIDGE

Computer Laboratory

## Sketchpad: A man-machine graphical communication system

Ivan Edward Sutherland

September 2003

New preface by Alan Blackwell and  
Kerry Rodden.

15 JJ Thomson Avenue  
Cambridge CB3 0FD  
United Kingdom  
phone +44 1223 763500  
<http://www.cl.cam.ac.uk/>

© 2005 Ivan Edward Sutherland  
This technical report is based on a dissertation submitted January 1963 by the author for the degree of Doctor of Philosophy to the Massachusetts Institute of Technology.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

*<http://www.cl.cam.ac.uk/TechReports/>*

Series editor: Markus Kuhn

ISSN 1476-2986

# Preface to this Electronic Edition

Alan Blackwell and Kerry Rodden  
University of Cambridge Computer Laboratory

Ivan Sutherland's Sketchpad is one of the most influential computer programs ever written by an individual, as recognized in his citation for the Turing award in 1988. The Sketchpad program itself had limited distribution — executable versions were limited to a customized machine at the MIT Lincoln Laboratory — so its influence has been via the ideas that it introduced rather than in its execution. Sutherland's dissertation describing Sketchpad was a critical channel by which those ideas were propagated, along with a movie of the program in use, and a widely-cited conference publication [10]. Copies of the dissertation were distributed relatively widely, but it was never published commercially. It is still available in the form of a technical report from MIT, but we believe it deserves wider readership — hence this electronic archival publication.

After 40 years, ideas introduced in Sketchpad still influence how every computer user thinks about computing. It made fundamental contributions in the area of human–computer interaction, being one of the first graphical user interfaces. It exploited the light-pen, predecessor of the mouse, allowing the user to point at and interact with objects displayed on the screen. This anticipated many of the interaction conventions of direct manipulation, including clicking a button to select a visible object, and dragging to modify it. Smith's Pygmalion [9], heavily influenced by Sketchpad, made a more explicit argument for the cognitive benefits of this kind of direct interaction and feedback, coining the term “icon”, and making it clear that graphical images could represent abstract entities of a programming language. Smith was a member of the team that developed the Xerox Star workstation on these principles; in a retrospective article [4] they acknowledge that “Sketchpad influenced Star's user interface as a whole as well as its graphics applications”, providing a direct link to the commercialization of the Macintosh and Windows interfaces and widely recognized benefits of direct manipulation [8].

Sketchpad encountered a critical challenge that remains central to human-computer interaction. Sutherland's original aim was to make computers accessible to new classes of user (artists and draughtsmen among others), while retaining the powers of abstraction that are critical to programmers. In contrast, direct manipulation interfaces have since succeeded by reducing the levels of abstraction exposed to the user. Ongoing research in end-user program-

ming continues to struggle with the question of how to reduce the cognitive challenges of abstract manipulation [1]. Nevertheless, Sutherland's attempt to remove the division between users and programmers was not the only system that, in failing to do so, provided the imaginative leap to a new programming paradigm. Nygaard and Dahl's Simula [7] was the first conventional programming language incorporating the principles of object orientation, but Sketchpad's implementation of class and instance-based inheritance (though not called objects) predated Simula by several years.

There appears to have been a common influence through the work of Douglas T. Ross, who is mentioned in the acknowledgements of this dissertation and also cited in the MIT Lincoln Laboratory technical report based on it. Ross sat on the Algol 68 committee with C. A. R. Hoare in the mid-1960s, where his previous work on a record-like data structure (called a *plex*) influenced Hoare's own ideas\* on abstract data types [3], later credited by Nygaard and Dahl as the origin of the class definition mechanisms in Simula [7].

Alan Kay's seminal Dynabook project, which led both to the Xerox Star and to the explosion of interest in object oriented programming through his language Smalltalk, was directly influenced by Sketchpad. Kay has written of the fact that the genesis of Smalltalk lay in the coincidental appearance on his desk of both a distribution tape of Simula and a copy of Sutherland's Sketchpad thesis [5]. Kay recognized that the two systems were based on the same underlying type concepts (apparently derived via two different routes from Ross's plex), and that these could form the basis of a more widely usable programming system. In comparing these two routes of influence, Simula was a far larger project than Sketchpad, rightly recognized as the first object-oriented programming language, but we hope that the special emphasis of Sketchpad on supporting abstraction in the user interface itself may yet become viable as a result of ongoing research efforts [2,6].

As with many early publications of computer science, this dissertation is also interesting for the way in which it explores important concepts that are now considered familiar, but which at the time demanded continual small discoveries by every researcher. The first-person account of the history of the project in Chapter 2 reads almost like an excerpt from an autobiography, as Sutherland describes how he had to "follow the stumbling trail" towards generality, through the different versions of Sketchpad. His rather charming proposal that dynamic data structures should be described using the terminology "hen and chickens" has been a sad loss when compared to the far more prosaic terminology of linked lists and garbage collection. The struggles of developing custom hardware while also exploring far-reaching abstractions are also far removed from current research experiences.

Chapter 9 provides an immediate illustration of how far computer graphics has moved on in the 40 years since Sketchpad's development. For example, Sutherland says that "if the almost identical but slightly different frames that are required for making a motion picture cartoon could be produced semi-automatically, the entire Sketchpad system could justify itself economically

---

\*Personal communication with C. A. R. Hoare.

in another way". Now, of course, we are used to seeing entire feature films created from computer graphics. Also, in choosing manipulation of facial features as an example, Sutherland has anticipated the sometimes controversial facilities available in modern photograph editing tools.

Sutherland's clear writing makes all of these issues a fresh source of enjoyment to the contemporary reader, and we hope that it will reach a new audience with the assistance of this electronic edition, continuing the great influence that Sketchpad has had on both users and programmers.

## This Edition

Our aim in preparing this edition has been to create an archival copy of the Sketchpad dissertation, suitable for electronic access and scholarly reference. Although created in consultation with current international research efforts in electronic archival, it is clear that there are, as yet, no common conventions for electronic archive formats. Our priorities have been that this edition should be accessible for download using current technology (i.e. in a relatively small file size), that it should be suitable for electronic search and indexing, that it should be easy to read both on paper and on the screen, and that it should be faithful to the original document. These have not been easy criteria to meet, and our chosen solution (L<sup>A</sup>T<sub>E</sub>X to PostScript to PDF) has several disadvantages, but is the best overall solution we could find.

There are some editorial choices that should be explained. We thought it important to indicate original page numbering, so that citations of the original dissertation could be traced, but wished to avoid the decreased readability that would have resulted from simply reproducing the original double-spaced typescript. We therefore chose not to preserve page breaks and line breaks, instead marking the positions of the original page breaks (with the || symbol) throughout the main body of the text, giving the original page number in the margin (next to the ↓ symbol). For figures, the original page number is noted in the caption. We also chose not to correct any errors we found in the original document, in order to provide the textual equivalent of a facsimile edition. These include a few spelling errors (to Ivan's embarrassment), and also the rather idiosyncratic fact that the original dissertation had two pages 106.

An exact facsimile copy of the original dissertation (where the pages have simply been scanned, not transcribed) can be purchased in hardcopy or PDF from the Digital Library of MIT Theses at <http://theses.mit.edu/>. Each individual page can also be viewed as a GIF image, free of charge, which may be a useful reference for readers wishing to check the layout of the original.

We are very grateful to Ivan Sutherland, who has encouraged this project, and who personally proof-read the original scanned text. We are also grateful to Malcolm Sabin, who kindly loaned us his original copy of the dissertation for over a year. This work has been supported by the Engineering and Physical Sciences Research Council, UK.

September 2003

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

## LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

## FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

## E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.