Tracing the Dynabook: A Historiograph[1]

John W. Maxwell
Canadian Centre for Studies in Publishing
Simon Fraser University
Vancouver, British Columbia
jmax@sfu.ca

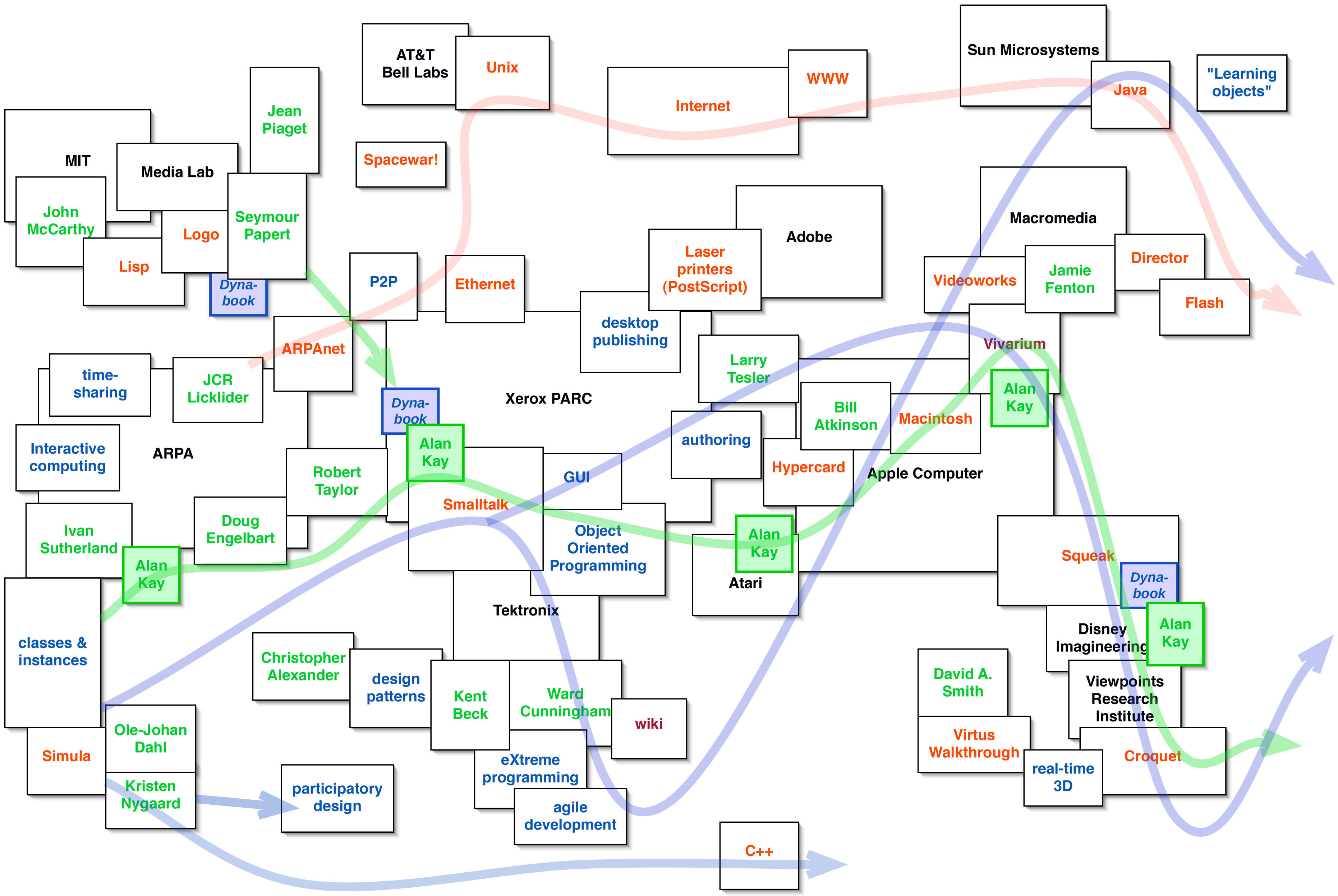
To begin with, this diagram is badly cluttered. The attempt to capture even a handful of the historical threads (genealogies?) through the origin myths of personal computing becomes a tangle of traces: personalities, artifacts, ideas, and interlinkages. To anyone who knows anything of the subject—and of course we all do simply by virtue of our co-construction of this story—it is painfully incomplete. Too add more, however, is to further reduce the legibility of this rendering.

The task is to trace the intellectual histories and heritage underlying some of the key agents of the development of "personal computing" as we know it. To identify key people—Douglas Engelbart at SRI; Alan Kay at Xerox PARC; Steve Jobs at Apple Computer—is easy enough. Similarly key ideas and artifacts marking the milestones make themselves readily visible. But naming the icons is a far cry from telling a credible story, because of course the intellectual history of this period is, in its richness, "deeply intertwingled."[2] The writing of this history is a daunting task, as it quickly runs to dissertation length and beyond. But to *diagram* it—poster-style—is a different thing: the result is a kind of data compression which allows a reasonably tight representation of a good number of historical relationships, while replying on the participation of the—*informed(?)*—spectator to reconstitute it as narrative(s).

The method here is simple: the diagram runs roughly chronologically left to right,

[1] This paper was presented as a poster session at the REFRESH conference, First International Conference on the Media Arts, Sciences and Technologies held at the Banff Center sept 29-oct 4 2005 and co sponsored by the Banff New Media Institute, the Database of Virtual Art and Leonardo/ISAST.

[2] After Ted Nelson, a character notably absent from the present discourse.


MIT
John McCarthy
Lisp
Media Lab
Logo
Dyna-book
Seymour Papert
Jean Piaget
AT&T Bell Labs
Unix
Spacewar!
Internet
WWW
Sun Microsystems
Java
"Learning objects"
time-sharing
JCR Licklider
ARPAnet
P2P
Ethernet
Interactive computing
ARPA
Ivan Sutherland
Alan Kay
Doug Engelbart
Robert Taylor
Dyna-book
Alan Kay
Xerox PARC
Smalltalk
GUI
Object Oriented Programming
desktop publishing
Laser printers (PostScript)
Adobe
Larry Tesler
authoring
Hypercard
Bill Atkinson
Macintosh
Apple Computer
Alan Kay
Atari
Macromedia
Videoworks
Jamie Fenton
Director
Flash
Vivarium
Alan Kay
Squeak
Dyna-book
Alan Kay
Disney Imagineering
Viewpoints Research Institute
Croquet
David A. Smith
Virtus Walkthrough
real-time 3D
classes & instances
Simula
Ole-Johan Dahl
Kristen Nygaard
participatory design
Christopher Alexander
design patterns
Tektronix
Kent Beck
Ward Cunningham
wiki
eXtreme programming
agile development
C++

beginning in about 1960 and leading to the early years of this century. Where nodes cluster together and overlap, the implication is one of overt influence, co-habitation, direct communication. More distant links are tentatively established by interconnecting lines. Furthermore, there is a demand made of the spectator to fold the two-dimensional space of the diagram into a multi-dimensional warp, in which distant corners can be imagined touching. Such is the limitation of paper, and our paper-trained minds. Ultimately, it likely requires a *guide*, either in the form of a handwaving agent who attends to the diagram and agrees to add the bits and pieces suggested by various spectators—or a textual one like this, which must resist such contributions.

A Cursory Guide in the Form of Narration

This history, largely focused on the 1970s and 1980s, has of course everything to do with the 60s before it—in particular the vision of interactive computing forwarded by JCR Licklider and his successors at the Pentagon's Advanced Research Projects Agency (ARPA).[3] Herein lies the first nest of omissions; the military-industrial threads leading into this diagram are manifold: Licklider himself was part of the defense department contractor BBN, a psychoacoustician interested in the problem of battlefield command and control systems. That the diagram highlights only academic research in this early period is, however, not an omission; Licklider's largely discretionary research budgets were, in the main, directed at American universities: MIT, Stanford, and the University of Utah are at least implicitly on the map here.

This history equally has everything to do with what comes *afterward*; this is the historian's curse. Key clusterings of nodes reflect the later industrialization of concepts like Desktop Publishing, Object-Oriented Programming, and the commodity PC; these are easily visible with the hindsight that their very establishment provides. The direction of this graph is overtly divergent: from a handful of originary spaces comes a plurality of forms and genres. But it is also, implicitly, convergent as well, and in this is the optimism of this particular historiograph(y).

[3] See M. Mitchell Waldrop. *The Dream Machine: J. C. R. Licklider and the Revolution that Made Computing Personal*. New York: Viking, 2001; and National Research Council. *Funding a Revolution: Government Support for Computing Research*. Washington, D.C.: National Academy Press, 1999.

The 'space' inscribed in this diagram is almost entirely American, from the Pentagon's technology funding of the 1960s and the academic computer-science discipline which both results from and shapes that funding source to the large-scale monopoly-capital sites (Xerox, Bell Labs, Apple, Disney). Noteworthy, then, are the Norwegians Dahl and Nygaard, who created the original object-oriented language Simula in time for it to make a key impact on the Americans who dominate the center of the diagram, and in particular, Alan Kay.[4]

Kay was a graduate student at the University of Utah in the mid-1960s under Ivan Sutherland, creator of SKETCHPAD and thus "father of computer graphics." Kay began working out a vision of 'personal' computing in his graduate work at Utah, influenced by the promises of Moore's Law and the ARPA focus on interactive computing. While a graduate student he had a series of experiences with the embryonic idea of Object-Oriented Programming (OOP): in early computing architectures, in Sutherland's SKETCHPAD, in Lisp's recursion, and finally, in the Dahl and Nygaard's Simula. In 1968 he met MIT's Seymour Papert,[5] who had just begun his famous work with children exploring geometric microworlds via the Logo language. Papert's focus on children impressed Kay to such a degree that Kay's research ever since has had children at the centre:

> This encounter finally hit me with what the destiny of personal computing *really* was going to be. Not a personal dynamic *vehicle*, as in Englebart's metaphor opposed to the IBM "railroads," but something much more profound: a personal dynamic *medium*. With a vehicle one could wait until high school and give "drivers ed," but if it was a medium, it had to extend to the world of childhood.[6]

In 1970, ex-ARPA director Robert Taylor was building a team for Xerox' new Palo Alto Research Center, hiring a substantial number of researchers from the ARPA project with the promise of open-ended research. Alan Kay was given the opportunity to head up

[4] Alan C. Kay. "The Early History of Smalltalk." In *History of Programming Languages II*, edited by Thomas J. Bergin & Richard G. Gibson, 511-578. New York: ACM Press, 1996.

[5] Papert, a South African, had studied developmental psychology with Jean Piaget in Geneva joining Marvin Minsky's Artificial Intelligence lab at MIT in the 1960s.

[6] Kay 1996. p. 523.

the Learning Research Group at PARC with the long-term goal of creating the *Dynabook***,** a portable "personal dynamic medium" which would serve as the master medium of a new literacy in the same way that the printed book had been the master medium of modernity. During the 1970s, Kay and his colleagues at Xerox PARC invented an enormous proportion of the technologies which today comprise "personal computing": desktop computers, the graphic user interface, local-area networks, laser printers and desktop publishing. At the same time, Kay's Learning Research Group turned the embryonic object-oriented ideas in Simula into a full-blown software development paradigm with the canonical OOP language *Smalltalk,* conceived as a "communications medium" for the Dynabook. By the end of the 1970s, though, it had been enshrined as the key technology in an emerging vanguard of professional programming.[7]

The translation of Smalltalk from a medium for kids' media literacy into a software engineering platform is mirrored in the industrialization of "personal computing" in the early 1980s; the guiding vision of children and computers fall away and the personal computer was turned into a commodity sold to businesses. Both personal computing and the OOP community have thrived, as significant clusterings in the diagram indicate, but the focus on children was lost.

In the 1980s and early 1990s, Alan Kay went underground, in a sense. Most importantly, he directed a behind-the-scenes research project at Apple Computer called the *Vivarium*, a long-term school-based program which, among other things, aimed to develop a complex systems simulation environment for schoolchildren at the Los Angeles Open School. The Vivarium's fruits are hard to locate today, lurking behind the scenes in a handful of multimedia "authoring" systems, a little bit of the later history of Smalltalk, and a number of graduate theses from the MIT Media Lab.[8]

[7] The trace on the diagram is via electronic instrument maker Tektronix, for a time in the 1980s the nexus for Smalltalk and OOP development. Two leading figures from Tektronix are Kent Beck and Ward Cunningham who are associated with the "Design Patterns" approach to software engineering (drawing from the work of architect Christopher Alexander); eXtreme Programming and "agile" methodologies, a bottom-up software design approach; and the development and use of Wikis. This clustering of ideas arguably embodies a number of the virtues arising from the Smalltalk community of the 80s and 90s.

[8] Publications drawing on the Vivarium's 7 years are few and far between. Notable are Alan C. Kay. "Computers, Networks and Education." *Scientific American 265*(Sept. 1991):138-148.1991; Mike Travers. "Agar: An Animal Construction Kit". M.S. Thesis, MIT, 1988; Druin, Allison. "NOOBIE: The Animal Design Playstation." *SIGCHI Bulletin 20*(1)(July 1988):45-53.1988; Jay Fenton, & Kent Beck.. "Playground: An Object Oriented Simulation System with Agent Rules for Children of All Ages." *OOPSLA 89*

By the mid-1990s, Kay's vision of personal computing had in a sense been largely realized: hundreds of millions of personal (even portable) computers in use around the world, connected to the global Internet that the ARPA directors had prophesied, the object-oriented paradigm in ascendance with the new Java language unleashed on the Web. But the core of Kay's vision was missing: the focus on children, a new literacy, and the ideal of users actively building their own "personal dynamic media" rather than buying shrinkwrapped application packages. Furthermore, Apple Computer, where Kay had been for a decade, seemed on the verge of its demise. In 1995, Kay reassembled a sizable chunk of his original team from Xerox and prepared *Squeak*—an escape pod from Apple? A response to the supersized architecture of Java? A volley at the corporate shrinkwrapped software industry? Squeak, whatever else it might be, was an open-source cross-platform reimplementation of Smalltalk, branded as a personal multimedia toolkit, and aimed at kids once again.[9]

What followed is no doubt less dramatic than the story may have appeared in the late months of 1995. Apple Computer survived, of course, though Kay's team left to join Disney Imagineering; Java continued to grow in popularity (and complexity); shrink-wrapped software continued to make billions of dollars for Microsoft and Adobe and their kin. And, while the Internet nurtured an energetic open-source community which embodies the spirit of much of Kay's (and the ARPA community's) vision of computing, the focus on children was still absent. Squeak, despite its many virtues, has not succeeded in capturing the imagination of more than a few thousand people around the world; and so the Dynabook vision, while more technically possible today than ever before, remains a lamentably marginal notion. Much of Squeak's current momentum comes from an offshoot project called *Croquet*, which replaces the traditional two-dimensional desktop GUI with a networked, immersive 3D environment.[10]

Proceedings. 1989. There is also a brief mention in Stewart Brand's popular book *The Media Lab: Inventing the Future at MIT.* (New York: Viking. 1987)

[9] See Dan Ingalls et. al. "Back to the Future: The Story of Squeak, A Practical Smalltalk Written in Itself." *OOPSLA '97 Proceedings*. 1997.

[10] David A. Smith et. al. "Croquet: A Menagerie of New User Interfaces." *Proceedings of the Second international Conference on Creating, Connecting and Collaborating Through Computing. C5*. IEEE Computer Society. 2004.

The ‘optimistic’ convergence hinted at by the direction of the arrows on the righthand side of the diagram, then, is due to my tentative observation that the key virtues of the ARPA project—and thus the Internet itself—and in the Xerox PARC vision of personal computing are lurking within the main thrusts of personal computing today. Phenomena like the rise of a “Web 2.0” paradigm implicitly advance many of these values,[11] even while remaining overtly ignorant (or at least shortsighted) of their intellectual history. In a similar vein, the free/open-source software movement[12] of recent years has been enormously successful in promoting a model of software development and distribution that is an alternative to the market. The open-source movement, exemplified by the Linux operating system, but also reaching far into the more mainstream world of Web development, is nothing if not a kind of “curriculum” akin to what Kay foresaw in the 1970s, though the open-source community may be woefully—even perversely—blind to this aspect of its own nature. The seeds, however, are there, and thriving online cultures nurture kindred ideals.

Finally, the kids are alright—now steeped in multiple generations of video-game culture[13] which itself threatens to enter their terrain of “personal dynamic media” as the gaming market and culture grows to ever larger and more complex scale. The ongoing history of personal computing, I believe, reveals a fluid, even pre-paradigmatic, space despite Microsoft’s apparent black-boxing and hyper-successful commodification of personal computing. Larger currents are active; I have no doubt that a re-imagining of this historiograph(y) in five years’ time will produce an even more tangled and complex weave. For now, let the current attempt at least be a starting point for reflection and research.

[11] The “Web 2.0” movement, for instance, embraces the “Model-View-Controller” design pattern, one of the original architectures emerging from the Smalltalk community in the early 1980s.

[12] The free/open-source software movement owes more to Unix culture than to the ARPA project per se, but this heritage cross-crosses the diagram heavily. See Eric S. Raymond. *The Art of Unix Programming.* New York: Addison-Wesley. 2003.

[13] With the minor exception of Atari, where Kay served as research scientist in the early 1980s, video games and game culture is completely absent from the diagram. When I think where it should go, the best answer seems to be a gigantic box underlying all the others.